FORMULATION AND EVALUATION OF PARALLEL ALGORITHMS FOR THE ORBIT DETERMINATION PROBLEM

by

Jeffrey Scott Shaver

B.S., United States Air Force Academy (1970)

M.S., Purdue University (1971)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN INSTRUMENTATION

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

March 1980

© Charles Stark Draper Laboratory 1980

Signature of Author Signature redacted
Department of Aeronautics and Astronautics
March 1980

Approved by Signature redacted
Walter M. Hollister, Chairman of Thesis Committee

Approved by Signature redacted
Richard H. Battin, Thesis Advisor

Approved by Signature redacted
Paul J. Cefola, Thesis Advisor

Approved by Signature redacted
John P. Vinti, Thesis Advisor

Accepted by Signature redacted
Harold Y. Wachman
Chairman, Departmental Graduate Committee

ARCHIVES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
APR 14 1980
LIBRARIES

FORMULATION AND EVALUATION OF PARALLEL ALGORITHMS FOR THE ORBIT DETERMINATION PROBLEM

by

Jeffrey Scott Shaver

Submitted to the Department of Aeronautics and Astronautics on March 1980 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

ABSTRACT

Recent advances in parallel processor computer hardware architectures hold significant promise as a means of bringing large amounts of processing capability to bear on computationally intensive problems, such as the orbit propagation and orbit estimation problems. However, the utility of these new hardware architectures is heavily dependent on the structure of the computational problem. To realize the full advantages of the new parallel processors, the algorithmic structure of the application software must be complementary to the hardware architecture. This thesis presents a parallel orbit propagation algorithm and a parallel orbit estimation algorithm, both of which are compatible with a single instruction stream/multiple data stream (SIMD) parallel processor architecture.

The orbit propagation algorithm computes, in parallel, Chebyshev series approximations to the right-hand members of the equations of motion over orbital arcs up to two revolutions. Analytical formulae are used to directly obtain Chebyshev series representing the integrals of the equations of motion. The algorithm uses a Picard iteration technique to obtain the converged solution. This algorithm has been applied to the Cowell Class II equations, the high-precision Variation of Parameters equations and the averaged Variation of Parameters equations. Numerical comparisons with high-precision Cowell integrations are presented for near-circular and elliptical test cases, including varied fitting parameters, arc lengths and force models. The effects of numerical error accumulation are demonstrated by comparison between a parallel integration of the two body problem and the analytic solution using the Lagrange coefficients.

A Parallel Variable Metric function minimization algorithm (gradient dependent) provides a compatible orbit estimation capability. The cost function minimized is the weighted squares of the observation residuals and the solve-for parameters are the epoch state components. Analytical expressions have been developed in terms of the equinoctial elements for the state partial derivatives (needed for the gradient computation), which include the secular rates in the argument of perigee and the longitude of the ascending node due to the J_2 zonal harmonic. The software implementation of this algorithm is described. The performance of the Parallel Variable Metric algorithm is compared to the standard Differential Corrections algorithm in terms of accuracy and region of convergence. Speed-up ratios are calculated for both the orbit propagation and orbit estimation algorithms, indicating the potential performance improvement to be achieved if the algorithms were executed on a SIMD hardware architecture.

Thesis Supervisor: Walter M. Hollister
Title: Associate Professor of Aeronautics and Astronautics

Thesis Supervisor: John P. Vinti
Title: Lecturer of Aeronautics and Astronautics

Thesis Supervisor: Richard H. Battin
Title: Adjunct Professor of Aeronautics and Astronautics
Associate Department Head, C. S. Draper Laboratory

Thesis Supervisor: Paul J. Cefola
Title: Lecturer of Aeronautics and Astronautics
Section Manager, C. S. Draper Laboratory

ACKNOWLEDGEMENTS

The author is indebted to many people for their support and assistance in this effort. To those who are not listed here, my gratitude and thanks are no less sincere. The author wishes to thank the United States Air Force for the opportunity to complete this research. Additionally, a special thank you goes to Robert Duffy (CSDL), William Manlove (CSDL) and John Sinkiewicz (formerly CSDL) for their part in establishing and nurturing the Commander's Option program.

The most important experience in this effort has been the opportunity to work closely with my doctoral committee, consisting of Prof. Walter Hollister, Prof. Richard Battin, Dr. Paul Cefola and Dr. John Vinti. Their advice, counsel and encouragement sparked all aspects of this research. The author is profoundly grateful to them for their help, but especially to Dr. Cefola, my primary advisor and mentor. A large measure of thanks also goes to Wayne D. McClain (CSDL) for his advice and for his sympathetic ear.

Special thanks go to Sean K. Collins and Captain (Dr.) Andrew J. Green (USA/Armor), gentlemen and scholars both. The author is honored to be numbered among their friends.

A heartfelt thank you goes to Ms. Karen Smith (CSDL), whose unmatched professional skills transformed piles of scrawled notes into a polished product. The author is very grateful for her tireless efforts in completing this document.

Finally, the author would like to state that without the encouragement and support of his wife, Claire, and his children, Jennie and Jaime, this thesis would not have been completed. Their understanding and patience in the darkest of times lit the way. Hopefully, this debt can now be repaid.

Partial support for this research was provided by the USAF Space Division (formerly SAMSO) under Contract Number F04701-79-C-0062 and by the Advanced Orbit Determination Task in the Charles Stark Draper Laboratory Internal Research and Development program.

TABLE OF CONTENTS

Chapter			Page
1	INTRODUCTION		
	1.1	Background and Motivation	29
		1.1.1 Expanding Missions and Data Processing Requirements	31
		1.1.2 Current Operational Orbit Determination System Methods	37
	1.2	Research Objective	
	1.3	Prior Work	
		1.3.1 Advanced Hardware Architectures	53
		1.3.2 Advanced Software Algorithms	57
		1.3.3 Target Hardware Architecture	61
2	A PARALLEL ORBIT PROPAGATION ALGORITHM		
	2.1	Algorithm Development	66
		2.1.1 Picard's Iterative Method	69
		2.1.1.1 Example of Picard's Method	70
		2.1.1.2 Extension to Systems of Equations of First Order	78
		2.1.1.3 Practical Limitations	80
		2.1.2 Chebyshev Series Approximation of Right Hand Members	81
		2.1.2.1 Relevant Properties of Chebyshev Polynomials	82
		2.1.2.2 Function Approximation by Discrete Chebyshev Series	98
		2.1.3 Direct Integration of the Chebyshev Series	109
	2.2	Application to the Orbit Propagation Problem	114
		2.2.1 Cowell Formulation	115
		2.2.2 High Precision Variation of Parameters Formulation	121

Chapter			Page
	2.2.3	Averaged Variation of Parameters Formulation	128
2.3	Algorithmic Architecture		
	2.3.1	Software Description	132
	2.3.2	Algorithm Parallelism	150
2.4	Numerical Results		153
	2.4.1	General Test Case Descriptions	155
	2.4.2	Analytical Comparison Test Results	160
	2.4.3	Low Altitude Circular Test Results	197
	2.4.4	Low Altitude Elliptical Test Results	217
3	A PARALLEL ORBIT ESTIMATION ALGORITHM		232
3.1	Algorithm Flow		234
	3.1.1	Parallel Variable Metric (Straeter's) Method	236
	3.1.1.1	Steps in the PVM Method	238
	3.1.2	Example of the PVM Method	250
3.2	Application to Orbit Estimation Problem		265
3.3	Cost Functional and Gradient Vector Evaluation		278
	3.3.1	Assumptions in Orbit Estimation Formulation	280
	3.3.2	Analytical State Partial Derivatives	283
	3.3.2.1	Partial Derivatives of Position with Respect to Elements	284
	3.3.2.2	Partial Derivatives of Velocity with Respect to Elements	287
	3.3.2.3	Partial Derivatives of Elements with Respect to Velocity	294

Chapter			Page
		3.3.2.4 Partial Derivatives of Elements with Respect to Position	300
		3.3.2.5 Partial Derivatives of Current Elements with Respect to Epoch Elements	303
		3.3.3 Direct Integration Method	325
		3.3.4 Linearization Method	330
	3.4	Estimation Error Covariance Computation	337
	3.5	Algorithmic Architecture	
		3.5.1 Software Description	343
		3.5.1.1 Univariate Search Technique	357
		3.5.2 Algorithm Parallelism	360
	3.6	Numerical Results	
		3.6.1 Basic Test Case Descriptions	368
		3.6.1.1 Battin Test Case	376
		3.6.1.2 Adimurthy and Joy Test Case	380
		3.6.2 Observation Simulation	382
		3.6.3 Battin Test Case Results	387
		3.6.4 Adimurthy and Joy Test Case Results	433
4	ALGORITHM PERFORMANCE EVALUATION		459
	4.1	Performance Measures	460
	4.2	Orbit Propagator Performance Results	464
	4.3	Orbit Estimator Performance Results	469
5	CONCLUSIONS AND FUTURE WORK		
	5.1	Summary	480
	5.2	Conclusions	491
	5.3	Future Work	494

LIST OF APPENDICES

Appendix		Page
A	PERTINENT ORBITAL ELEMENT SETS	
	A.1 Keplerian Elements	502
	A.2 Equinoctial Elements	512
	A.3 Lagrange Coefficient Two Body Analytical Solution	520
B	SIMPLIFIED FORCE MODELS FOR THE STAND-ALONE PARALLEL PROCESSOR ORBIT DETERMINATION SYSTEM	
	B.1 Zonal Gravitational Potential Model	524
	B.2 Drag and Atmospheric Models	526
C	GODDARD TRAJECTORY DETERMINATION SYSTEM (GTDS-RD) SOFTWARE MODIFICATIONS	530
	C.1 Modified GTDS-RD Subroutines	531
	C.2 New GTDS-RD Subroutines	536
	C.3 Modified GTDS-RD Keyword Cards	537
	C.4 Program Listings	538
D	STAND-ALONE PARALLEL PROCESSOR ORBIT DETERMINATION SOFTWARE DESCRIPTION	
	D.1 Overall System Structure	645
	D.2 Subroutine Descriptions	648
	D.3 Common Block Descriptions	648
	D.4 Input Card Image Formats	661
	D.5 Program Listings	682
E	SIMULATION OF OBSERVATIONAL DATA	844
REFERENCES		858

LIST OF FIGURES

Figure		Page
1.1	Single Instruction Stream/Multiple Data Stream (SIMD) Array Processor	63
2.1	Set of Chebyshev Orthogonality Points	92
2.2	Structure of the [TX] Array	140
2.3	Structure of [STK] Array for Storage of k^{th} Iterate of Integration State	143
2.4	Structure of [FORC] Array for Storage of k^{th} Iterate of Force Functions	144
2.5	Structure of [A] Array Containing Integrand Series Coefficients	147
2.6	Structure of [BC] Array for Storage of Integrated Series Coefficients	148
2.7	Picard-Chebyshev Parallel Orbit Propagation Algorithm	151
2.8	Difference in "x" Velocity Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)	161
2.9	Difference in "y" Velocity Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)	162

Figure		Page
2.10	Difference in "z" Velocity Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)	163
2.11	Difference in "x" Position Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)	165
2.12	Difference in "y" Position Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)	166
2.13	Difference in "z" Position Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)	167
2.14	Difference in "x" Acceleration Component: Chebyshev Series Fit vs. Analytical Solution	168
2.15	Difference in "y" Acceleration Component: Chebyshev Series Fit vs. Analytical Solution	169
2.16	Difference in "z" Acceleration Component: Chebyshev Series Fit vs. Analytical Solution	170

Figure		Page
2.17	Velocity Component Differences During First Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution	172
2.18	Position Component Differences During First Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution	173
2.19	Acceleration Component Differences During Final Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution	174
2.20	Velocity Component Differences During Final Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution	175
2.21	Position Component Differences During Final Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution	176
2.22	Difference in "x" Velocity Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)	178
2.23	Difference in "y" Velocity Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)	179

Figure		Page
2.24	Difference in "z" Velocity Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)	180
2.25	Difference in "x" Position Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)	181
2.26	Difference in "y" Position Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)	182
2.27	Difference in "z" Position Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)	183
2.28	Difference in "x" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)	185
2.29	Difference in "y" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)	186

Figure		Page
2.30	Difference in "z" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)	187
2.31	Difference in "x" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)	188
2.32	Difference in "y" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)	189
2.33	Difference in "z" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)	190
2.34	Difference in "x" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)	191
2.35	Difference in "y" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)	192
2.36	Difference in "z" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)	193

Figure		Page
2.37	Difference in "x" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)	194
2.38	Difference in "y" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)	195
2.39	Difference in "z" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)	196
2.40	Radial Position Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)	203
2.41	Cross Track Position Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)	204
2.42	Along Track Position Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)	205

Figure		Page
2.43	Radial Velocity Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)	206
2.44	Cross Track Velocity Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)	207
2.45	Along Track Velocity Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)	208
2.46	Semimajor Axis Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)	209
2.47	Equinoctial Element "h" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)	210
2.48	Equinoctial Element "k" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)	211
2.49	Equinoctial Element "p" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)	212

Figure		Page
2.50	Equinoctial Element "q" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)	213
2.51	Mean Longitude Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)	214
2.52	Semimajor Axis History for Two Revolutions of the Low Altitude Elliptical Test Case	220
2.53	Eccentricity History for Two Revolutions of the Low Altitude Elliptical Test Case	221
2.54	Inclination History for Two Revolutions of the Low Altitude Elliptical Test Case	222
2.55	Along Track Difference over 4.5 Hour Compare Span: Picard-Chebyshev (1 Rev Stepsize) versus Time Regularized Cowell Integration	225
2.56	Along Track Difference over 4.5 Hour Compare Span: Picard-Chebyshev (1/4 Rev Stepsize) versus Time Regularized Cowell Integration	227
2.57	Along Track Difference over 4.5 Hour Compare Span: Picard-Chebyshev (1 Rev Stepsize; Epoch Mean Anomaly = 0°) versus Time Regularized Cowell Integration	229

Figure		Page
3.1	Example Problem Cost Functional Projected on (x_1, x_2) Plane	251
3.2	Example Problem Cost Functional Projected on $(x_1, J(\vec{x}))$ Plane	252
3.3	PVM Algorithm Results for Example Problem	255
3.4	Definition of Cost Functional and Gradient Vector in PVM Orbit Estimation Algorithm	276
3.5	Pictorial Representation of the Direct Integration Method for Cost and Gradient Evaluations	328
3.6	Pictorial Representation of the Linearized Method for Cost and Gradient Evaluations	336
3.7	Structure of the [BASIS] Array for Storing the Parameter Space Basis Vectors	350
3.8	Structure of [XMAT] Array for Storing n Parallel Values of the Solve-for Vector	352
3.9	Structure of [XJATXJ] and [XGATXJ] Arrays for Storage of n Evaluations of the Cost Functional and Gradient Vector	355
3.10	Structure of [YMAT] Array for Storage of the n Gradient Vector Differences	356
3.11	Direct Integration Method Implemented on MIMD Hardware Architecture	362

Figure		Page
3.12	Direct Integration Method with Parallel Observation Processing	365
3.13	Linearization Method Implemented on SIMD Hardware Architecture	369
3.14	Coggins Univariate Search with Linearization and Parallel Observation Processing	372
3.15	Component Differences Between Estimated and True Epoch Position Vectors for Battin Test Case	392
3.16	Component Differences Between Estimated and True Velocity Vectors for Battin Test Case	393
3.17	Magnitude of Vector Difference Between Estimated and True Epoch Position Vectors for Battin Test Case	395
3.18	Magnitude of Vector Difference Between Estimated and True Epoch Velocity Vectors for Battin Test Case	396
3.19	Weighted Least Squares Cost Functional Value vs. Iteration Number for Battin Test Case	398
3.20	Cost Functional vs. Iteration Number for Battin Test Case: Iterations Two Through Five	399

Figure		Page
3.21	Magnitude of Gradient Vector vs. Iteration Number for Battin Test Case	401
3.22	Magnitude of Gradient Vector vs. Iteration Number for Battin Test Case: Iterations Two Through Five	402
3.23	Difference in "x" Position Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span	415
3.24	Difference in "y" Position Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span	416
3.25	Difference in "z" Position Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation-- Span	417
3.26	Difference in "x" Velocity Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span	418
3.27	Difference in "y" Velocity Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span	419

Figure		Page
3.28	Difference in "z" Velocity Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span	420
3.29	Radial Difference over 24 Hour Compare Span (21 Hour Predict) for Battin Test Case: PVM Estimated Ephemeris versus Truth Ephemeris	422
3.30	Cross Track Difference over 24 Hour Compare Span (21 Hour Predict) for Battin Test Case: PVM Estimated Ephemeris versus Truth Ephemeris	423
3.31	Along Track Difference over 24 Hour Compare Span (21 Hour Predict) for Battin Test Case: PVM Estimated Ephemeris versus Truth Ephemeris	424
3.32	Semimajor Axis Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case	427
3.33	Equinoctial Element "h" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case	428
3.34	Equinoctial Element "k" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case	429

Figure		Page
3.35	Equinoctial Element "p" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case	430
3.36	Equinoctial Element "q" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case	431
3.37	Mean Longitude Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case	432
3.38	Component Differences Between Estimated and True Epoch Position Vectors for Adimurthy and Joy Test Case	438
3.39	Component Differences Between Estimated and True Epoch Velocity Vectors for Adimurthy and Joy Test Case	439
3.40	Magnitude of Vector Difference Between Estimated and True Epoch Position Vectors for Adimurthy and Joy Test Case	440
3.41	Magnitude of Vector Difference Between Estimated and True Epoch Velocity Vectors for Adimurthy and Joy Test Case	441
3.42	Weighted Least Squares Cost Functional vs. Iteration Number for Adimurthy and Joy Test Case	442
3.43	Magnitude of Gradient Vector vs. Iteration Number for Adimurthy and Joy Test Case	443

Figure		Page
3.44	Cost Functional vs. Iteration Number for Adimurthy and Joy Test Case: Iterations Nine Through Eighteen	445
3.45	Magnitude of Gradient Vector vs. Iteration Number for Adimurthy and Joy Test Case: Iterations Nine Through Eighteen	446
A.1	Keplerian Elliptic Orbit in the Plane ($0 \leq e < 1$)	504
A.2	Spatial Orientation of the Keplerian Orbit	508
A.3	Equinoctial Reference Frame for the Direct Equinoctial Set in the Plane	514
A.4	Spatial Orientation of the Equinoctial Reference Frame for the Direct Equinoctial Set	517
C.1	Modified ORBTYPE Keyword Card for Picard-Chebyshev/Cowell Orbit Generator	539
C.2	Modified ORBTYPE Keyword Card for Picard-Chebyshev/Variation of Parameters Orbit Generator	540
D.1	Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System	649
D.2	Example Input Deck for Ephemeris Generation Run	663
D.3	Example Input Deck for Orbit Estimation Run	664

Figure		Page
D.4	Input Card Formats	665
E.1	Data Simulation Software Listing	847
E.2	Named Common Block OBSDAT for Storage of Simulated Observational Data	851

LIST OF TABLES

Table		Page
1.1	Characteristics and Methods of Example Operational Orbit Determination Systems	48
2.1	Comparison of Picard Iteration Method Approximations to Exact Solution of Equation 2.4	77
2.2	Software Module Descriptions for the Picard-Chebyshev Orbit Propagation Algorithm	134
2.3	Low Altitude Circular Test Case Epoch Conditions	156
2.4	Low Altitude Elliptical Test Case Epoch Conditions	156
2.5	Low Altitude Circular Test Cases for Picard-Chebyshev Integrator	158
2.6	Low Altitude Elliptical Test Cases for Picard-Chebyshev Integrator	159
2.7	Order of the Maximum Sampled Position Differences for the Low Altitude Circular Test Case	199
2.8	Order of the Maximum Sampled Position Differences for the Low Altitude Elliptical Test Case	218

Table		Page
3.1	Partial Derivatives of Position with Respect to Equinoctial Elements	288
3.2	Partial Derivatives of Velocity with Respect to Equinoctial Elements	290
3.3	Non-Zero Lagrange Brackets of the Equinoctial Orbital Elements	296
3.4	Non-Zero Poisson Brackets of the Equinoctial Orbital Elements	297
3.5	Partial Derivatives of Equinoctial Elements with Respect to Velocity Components	299
3.6	Partial Derivatives of Equinoctial Elements with Respect to Position Components	301
3.7	Summary of Secular Equations of Motion Due to J_2 Zonal Harmonic	314
3.8	Useful Partial Derivatives in Computation of J_2-Secular Perturbed Partial Derivatives	316
3.9	Partial Derivatives of "a" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation	318
3.10	Partial Derivatives of "h" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation	319
3.11	Partial Derivatives of "k" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation	320

Table		Page
3.12	Partial Derivatives of "p" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation	321
3.13	Partial Derivatives of "q" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation	322
3.14	Partial Derivatives of "λ" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation	323
3.15	Software Module Descriptions for the Stand-Alone Parallel Variable Metric Orbit Estimation Algorithm	345
3.16	Truth Ephemeris Epoch Conditions for the Battin Test Case	377
3.17	*A Priori* Solve-for Vectors for the Battin Test Case	379
3.18	Truth Ephemeris Epoch Conditions for the Adimurthy and Joy Test Case	381
3.19	*A Priori* Solve-for Vectors for the Adimurthy and Joy Test Case	383
3.20	Summary of Observational Data for Estimation Test Cases	386
3.21	Direct Integration Method for Battin Test Case with Cartesian Solve-for Vector (Conv. Criterion = 0.001)	388

Table		Page
3.22	Linearization Method for Battin Test Case with Cartesian Solve-for Vector (Conv. Criterion = 0.001)	403
3.23	Linearization Method for Battin Test Case with Keplerian Solve-for Vector (Conv. Criterion = 0.001)	406
3.24	Linearization Method for Battin Test Case with Equinoctial Solve-for Vector (Conv. Criterion = 0.001)	408
3.25	Comparison of Estimated Trajectory vs. Truth Ephemeris During 3 Hour Observation Span for Battin Test Cases	411
3.26	Comparison of Estimated Trajectory vs. Truth Ephemeris During 21 Hour Predict Span for Battin Test Cases	412
3.27	Linearization Method for Adimurthy and Joy Test Case with Cartesian Solve-for Vector	436
3.28	Direct Integration vs. Linearization Method for Adimurthy and Joy Test Case with Equinoctial Solve-for Vector	448
3.29	Orbit Determination Accuracies: PVM Results vs. Adimurthy and Joy Results	450
3.30	Convergence Results for the Parallel Variable Metric Orbit Estimation Algorithm	454

Table		Page
3.31	Adimurthy and Joy Test Case Estimation Results with Drag Effects Included in the Force Model	456
4.1	Speed-Up Ratio Computation for Low Altitude Circular Test Runs	465
4.2	Speed-Up Ratio Computation for Low Altitude Elliptical Test Runs	470
4.3	Progress per Iteration of the PVM Algorithm for the Battin Test Case (Linearized Method/ Position-Only Observations)	472
4.4	Progress per Iteration of the PVM Algorithm for the Adimurthy and Joy Test Case (Linearized Method/No Drag/Position-Only Observations)	475
4.5	Effect of Sequential Univariate Search on the Parallel Variable Metric Orbit Estimation Algorithm	478
D.1	Parallel Processor Orbit Determination System Subroutine Description: "Who Calls Whom"	655
D.2	Parallel Processor Orbit Determination System Subroutine Description: "Who Is Called By Whom"	658
D.3	Named Common Block References	662

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Since the beginning of the "Space Age" in 1957, space flights have been essentially large-scale scientific experiments. In these experiments, expensive and virtually hand-made and hand-tested spacecraft were expressly designed to determine the specific information desired and were launched by costly expendable booster systems. In the civil sector, these experiments searched for information about the space environment and man's ability to adapt to it, about nearby celestial bodies and about the Earth itself. The military sector was more concerned with the close observation of Earth from space, as related to determination of the intentions and capabilities of potential enemies, and the form and structure of military space experiments reflected these interests.

While experimentation must be continued to explore the more remote reaches and complex phenomena of space, a new phase in the Space Age is about to begin. This new

phase will concern the use of space, and its attendant advantages, for productive commercial ventures. Additionally, in this phase of activity, spaceborne intelligence gathering systems will become key elements in the structure of the national defense and policy-making mechanisms of this country.

The increased use of space results in increased ground support requirements, especially in the area of ground based orbit determination (OD) systems. These OD systems perform two basic functions:

1. Orbit propagation, in which orbital initial conditions are propagated from some initial time to some final time under the influence of appropriate perturbation models, and

2. Orbit estimation, in which noisy observational data are used to improve the estimate of certain quantities chosen as representative of the satellite orbit.

Most current and future satellite systems require this type of ground support and, as these requirements expand, ground facilities are beginning to search for faster and more efficient methods of providing the requisite orbital

data. This thesis investigates a hardware/software approach to satisfying these needs which shows the potential of a significant performance improvement over current methods while maintaining high accuracies.

1.1.1 Expanding Missions and Data Processing Requirements

The new era of space use, as opposed to space exploration, has already begun in the civil sector. The economic health and growth of the Comsat and Intelsat organizations are well known, as is the success of recent domestic communication satellite ventures. These companies have found that the use of space systems for long haul communications is a profitable venture. It is anticipated that the advent of the Space Transportation System (STS) will assist these ventures and make possible new commercial operations using the unique environment of space. Already modifications and extensions of basic STS capabilities are being proposed as new and more ambitious space efforts are being planned (130). A partial description of these potential applications follows.

As stated above, satellites acting as long haul communication forwarding nodes were probably the first successful commercial use of space and these technologies are

being expanded and extended. New approaches such as the Tracking Data Relay Satellite System (TDRSS) promise new opportunities for growth (8, 18). These dual-mission spacecraft will provide near-continuous tracking and communication with on-orbit space vehicles, including STS, and will also provide expanded domestic communication capabilities. Since they will provide tracking and command and control channels to other satellites, the TDRSS is expected to allow closure of many National Aeronautics and Space Administration (NASA) overseas ground stations. This will allow such significant cost savings that the system is expected to pay for itself.

Another area of near-term commercial value is the use of space systems for the analysis of earth resources. Beginning with the original Landsat spacecraft (69), the scientific community has been extremely enthusiastic over the results and potentials of these experiments. A highly active user community has developed, prompting expansions in the capabilities and benefits of these systems (9, 14, 15). The profit making potential of commercial use of these technologies has been demonstrated (12), and at least one corporation has been formed to capitalize on these capabilities (13).

While not considered a commercial use of space, the civil sector plans to expand its use of space for astronomical observations and exploration. NASA future plans include a comet rendezvous and landing mission, a Martian soil sample mission, a solar research mission and expanded high-energy astronomy capabilities (57, 58). These future projects will expand our understanding of the origins of the universe and other cosmic phenomena.

Even as these near-term projects are being planned and executed, longer-term large-scale uses of space are being considered. It has been proposed that immense orbiting solar power stations could be constructed in space to convert sunlight to electricity and beam it to Earth in the form of microwave energy. To be economically realistic, these stations would have to deliver between 5×10^9 and 10×10^9 watts at the output of the earth terminal. It has been estimated that the operating costs of this form of energy production could be comparable to terrestrial nuclear power generation facilities (72).

The construction of "space factories" has also been considered (64, 65). These factories would use the vacuum and "zero-g" environment of space to aid in the production and fabrication of items which would be much more difficult

to produce on earth. Estimates indicate that an annual market of up to $1,500 million may exist for vaccines and drugs produced in space and that production rates of silicon ribbons for microelectronic circuits may increase by a factor of five in space production facilities. These estimates indicate that space industrialization will be a strong future trend.

Expansion of the space factory concept into larger scale habitable space structures designed for long term human occupation may eventually lead to the creation of "space colonies" (137). These colonies will not be intended as solutions to the Earth's expanding population, but may be used as processing facilities for ore and other resources mined from the moon or the asteroids. As terrestrial resources dwindle and production costs rise, space-based systems may become quite competitive with their earth-based counterparts.

In the military sector, space represents the "high ground" from which the activities of potential enemies can be observed, and their intentions and capabilities deduced. If necessary, weapons can then be directed upon them. For these reasons, the strategic significance of space capabilities to our national defense posture is

immense (136). These military and space capabilities can be conveniently divided into four categories: reconnaissance, command control and communications (C^3), navigation and interception.

Space seems to provide a unique medium for the discrete observation of denied terrestrial areas. Satellites have a nearly vibration-free environment for photography, they can scan large surface areas in a relatively short time and, until recently, they were considered impervious to attack from the Earth. The genesis of this military space activity is comprehensively discussed in Klass' book, Secret Sentries in Space (91). Although most of these military space efforts are highly classified, it is now generally believed that this country, the Soviet Union and perhaps China (90, 91, 123, 134) have extensive space-borne reconnaissance capabilities. Photographic data, electromagnetic spectrum information, nuclear detonation data and missile launch detection capabilities would represent vital elements of intelligence data in the national decision-making processes.

The command and control of current sophisticated weapon systems depend heavily on extensive satellite communication networks. These wide-ranging and varying arrays

of satellites will provide global communications on all levels of military operations. Satellite navigation systems and their requisite ground support systems will act as vital force multipliers for both tactical and strategic weapons systems.

Interception represents a recent military space effort initiated by the Soviet Union and only recently acknowledged by this country (10, 16). The ability of an enemy to intercept, inspect and destroy satellites in space represents a serious threat to both military and civil space efforts. Several diplomatic efforts have been undertaken to limit or ban such weapons (17, 19, 20, 21, 22), but their level of success is unclear. In response, this country has initiated a program for research and development in this area (7, 11). Requirements to detect maneuvers by potentially hostile spacecraft may significantly increase the tracking and data processing requirements of those ground installations charged with monitoring earth orbiting objects.

The effect on ground-based support facilities of the burgeoning space activities encapsulated in the previous paragraphs is twofold. First, the number of vehicles which require support is rapidly growing. For example,

as of mid-1977 the North American Air Defense Command had catalogued more than 10,000 objects of which more than 4,300 were still in space and actively tracked. Current trends point to an increasing population in space (117), requiring expanded ground-based support. Second, the increasingly sophisticated missions and payloads of these earth satellites require improved accuracy in the orbit determination process. Highly accurate ephemeris information is required to meet mission requirements for such programs as Global Positioning System (GPS), Tracking Data Relay Satellite System (TDRSS) and LANDSAT. In many cases, the need for this data to be readily available to the user in a near real-time environment presents further significant ground-based or space-borne processing requirements.

1.1.2 Current Operational Orbit Determination System Methods

Numerous orbit determination (OD) systems have been created to meet previous support requirements for satellite ephemeris information. Each system has evolved to include certain capabilities necessary to satisfy its given operational needs. Ekelund (68) presents a detailed breakdown of many of these systems and their capabilities.

Rather than describe each OD system here, it is felt that a generic description of such a system augmented by some examples of operational OD systems displaying these generic characteristics would be more helpful. Each OD system must have some method of propagating the equations of motion either forward or backward in time. The satellite theories used to solve these equations of motion fall into two basic categories. In General Perturbation (GP) satellite theories, the solutions to the equations of motion are found in the form of analytical expressions for the position and velocity components, the orbital elements or the perturbation quantities. These analytical expressions can then be evaluated for any given time by direct substitution of that time into the analytical expressions. Special Perturbation (SP) satellite theories rely on numerical integration of the dynamical equations of motion in terms of the satellite position and velocity components, the orbital elements or the perturbations away from a reference solution.

Several GP theories are implemented in current OD systems. A primary theory in use is that due to Brouwer (39), in which the Hamilton equations of motion for the perturbed system, written in the canonical Delaunay variables, are solved by the method of von Zeipel. The secular,

long periodic and short periodic components of the motion are found for the second zonal harmonic in the gravitational potential while the long periodic and additional secular components of the motion are found for the third, fourth and fifth harmonics. Lyddane (107) improved the method by eliminating the zero eccentricity and zero inclination singularities in the original formulation. The work of Brouwer and Hori (40) extended this treatment to include drag effects, but the resulting analytical expressions are extremely complex. Reference 157 points out quite effectively the difficulty in including non-gravitational perturbations in the Brouwer-type formulations in that the transformed differential equations no longer have the simple solutions of Reference 39, such that very complex expressions result or numerical integrations are required.

The GP theory of Lane (101, 102) is built upon the basic concepts of the Brouwer-Hori theory. However, where Brouwer-Hori used a non-rotating spherical exponential model for the atmosphere, Lane employed a non-rotating spherical power function atmosphere model. The form of the atmosphere model allowed the doubly-primed (second transformation in the von Zeipel method) equations to be directly integrable without resort to series expansion for the

density function, as was required in the Brouwer-Hori development. The resulting theory is of major operational importance (108).

Another important approach to a General Perturbation satellite theory is that of Vinti (149, 150, 151, 152). In this method the axially symmetric gravitational potential is expressed in oblate spheroidal coordinates, such that the resulting Hamilton-Jacobi equation for the satellite orbit is separable. By making an assumption that the magnitude of the fourth zonal harmonic is equal to the negative of the square of the second harmonic, the necessary equations can be solved by quadratures. Later work allowed the exact inclusion of the third zonal harmonic. The work of Sherill (135) used Vinti's intermediary orbit as a point of departure to develop an analytical satellite drag theory similar to the Brouwer-Hori development. Additionally, Watson, Mistretta, and Bonavito (153) developed an alternate method to include drag effects in the Vinti theory. In this method, both a drag-free oblateness solution and an oblateness-free drag perturbed Variation of Parameters solution are carried out over a given time interval. These two solutions are combined iteratively to yield a single solution over that interval that includes both perturbations and the drag-oblateness coupling effect.

Numerous other General Perturbation methods exist, although they are not currently in widespread operational use. For example, Reference 6 describes several General Perturbation methods in addition to those discussed above. The primary advantage of any General Perturbation satellite theory lies in its computational efficiency. Because of the explicit expressions developed in these methods, the evaluations of the required state parameters can be done directly and only when output is required. This is in contrast to a numerical integration scheme, where the integration state parameters must be evaluated at each integration step whether they are used there or not. These explicit expressions can be found, however, only for a restricted set of perturbation models. In many cases of operational interest, these restricted models provide sufficient accuracy (93). However, as accuracy requirements become more stringent, satellite theories must accommodate more extensive perturbation models. This is not usually possible in General Perturbation theories without extensive reformulation and reprogramming. The restricted force models are considered to be the primary disadvantage of General Perturbation satellite theories.

When very high accuracy is required in the solution of the satellite equations of motion, Special Perturbation (SP) satellite theories are generally used. These techniques rely on numerical integration of the state equations. The integrands of these state equations may contain extremely complex force models. However, these integrands must be evaluated and the state components determined at every integration grid point. Thus, while the Special Perturbation satellite theories allow inclusion of extensive perturbation models, they have the corresponding disadvantage of being quite computationally expensive.

The most prominent formulation of the satellite equations of motion used in a Special Perturbation satellite theory is one for which the differential equations for the inertial position and velocity components are integrated directly. This formulation is usually termed the Cowell formulation. The Cowell formulation does not take advantage of the fact that the unperturbed satellite motion describes a conic section. The method of Encke, on the other hand, numerically integrates the equations for the differences between the true ephemeris and the ephemeris described by the osculating Keplerian orbit, for which the analytical solution is known. Reference 41 provides an

excellent description of the Cowell and Encke formulations. Another Special Perturbation formulation of interest is that of Variation of Parameters. In this formulation, the perturbed element rates are formed, generally in the Gaussian form of the VOP equations, and these are numerically integrated. Appendix A of Reference 82 and Reference 105 present detailed derivations of the VOP-type equations. Again, the Special Perturbation satellite theories described above all are computationally expensive and become more so as accuracy requirements increase.

Recently a new body of satellite theories has emerged with the goal of overcoming the primary disadvantages of both the General Perturbation and Special Perturbation satellite theories. The goal is to produce satellite ephemeris information with the accuracy of current high-precision Special Perturbation methods, but with computational efficiency approaching that of General Perturbation methods. These new approaches to the artificial satellite problem are classed as semianalytical satellite theories.

In the semianalytical satellite theory, the Generalized Method of Averaging (GMA) is applied to the Variation of Parameters equations of motion. In principle, the

application of the GMA removes the short periodic or high frequency components of the motion. The resulting averaged equations contain only secular and long periodic effects. These equations can then be numerically integrated using very large integration stepsizes, which allows significant computational efficiency. At the required output times, the short periodic components of the motion due to the perturbation sources included in the force model are then computed. These short periodics are then added to the numerically integrated (or interpolated) solution to the averaged equations of motion to produce the osculating or high precision orbital elements.

As previously stated, the increased interest in improving the throughput capabilities of operational orbit determination systems while maintaining high accuracies has sparked significant interest in semianalytical satellite theories. The works of Uphoff (148) and Lutzky and Uphoff (106) developed a fully numerical semianalytical satellite theory which relied on numerical averaging of all perturbations. The short periodic components of the motion were approximated by Fourier series whose coefficients were found by numerical quadratures. The large number of numerical averaging and quadrature operations required for a

relatively complex force model caused this approach to be computationally costly.

A more efficient approach to the semianalytical satellite theory is that due to Cefola, Long and Holloway (49). In this approach, analytical averaging techniques are applied to conservative perturbations while numerical averaging is reserved for non-conservative perturbations. The recent work of Green (82) provides an excellent discussion of the background of this semianalytical satellite theory. His work develops a formulation for the first order short periodic components of the motion for either conservative or non-conservative perturbations. He also extends the averaged equations of motion to second order in single or multiple perturbations, allowing rigorous inclusion of the oblateness-drag coupling terms in the averaged equations. These efforts, in conjunction with several other works described by Green, form the basis of a semianalytical satellite theory currently under development and test at the Charles Stark Draper Laboratory for possible use in operational orbit determination systems.

Orbit determination systems must also include some capability to incorporate observational data into an improved estimate of the orbital parameters. The primary

method for accomplishing this task in most operational orbit determination systems is that of Gaussian weighted least squares (41). In this method, the observational data are processed in "batches" and the resulting estimate minimizes the aggregate of the weighted squares of the observation residuals. Recent developments in sequential estimation techniques have resulted in several new approaches to the orbit estimation problem. The work of Myers (119) developed and evaluated several sequential filtering methods for use in orbit estimation for near-earth satellites.

Another approach which shows significant promise for the orbit estimation problem relies on nonlinear function minimization techniques. Recent work by Adimurthy and Joy (4, 5) applied several nonlinear function minimization algorithms to the orbit estimation problem formulated as a nonlinear parameter optimization problem. These results indicated that certain of their test algorithms converged for *a priori* solve-for vectors far removed from the true orbital parameters. This type of algorithm "robustness" could be very useful in such operational orbit estimation tasks as early orbit determination.

Numerous operational orbit determination systems currently exist. Examples of such systems are the Goddard Trajectory Determination System (GTDS) of NASA's Goddard Space Flight Center (46), the Advanced Orbit/Ephemeris Subsystem (AOES) of the Air Force Satellite Control Facility (125), the Simplified General Perturbation (SGP, SPG4) theories (86) of the Air Force Space Computation Center (1), the TRACE66 program of the Aerospace Corporation (127), and the CELEST system of the Naval Surface Weapons Center (122). Each such system contains those particular algorithms and capabilities required to accomplish its particular mission. Table 1.1 presents a brief examination of the major components of each operational system, as described in their respective program documentation. This table indicates the wide variety of approaches which may be used to accomplish the generic orbit determination system requirements described above. However, a single characteristic is common among all these systems. That single unifying characteristic is that every system is designed with serial algorithms for use on large-scale, sequentially architected computer hardware. This type of hardware/software architecture limits the system throughput in many orbit determination applications.

Table 1.1. Characteristics and Methods of Example Operational Orbit Determination Systems (Page 1 of 4)

SYSTEM	GENERAL PERTURBATION FORMULATIONS	SPECIAL PERTURBATION FORMULATIONS	ESTIMATION METHODS
GTDS Refs. 46, 155	• Brouwer • Brouwer-Lyddane • Brouwer-BMW* • Brouwer-Lyddane-BMW* • Vinti • Vinti-BMW*	• Cowell • Cowell (time-regularized) • Variation of Parameters Element Sets: -- Keplerian -- Equinoctial (mean long.) -- Equinoctial (ecc. long.) -- Dallas -- Ideal (true long.) -- Ideal (ecc. long.) -- Kustaanheimo-Stiefel -- Delaunay-Similar -- Averaged Keplerian -- Averaged Equinoctial	• Differential Corrections (Batch) • Kalman Filter (Sequential) • Extended Kalman Filter (Sequential)

* BMW = Bonavito, Mistretta and Watson drag method

Table 1.1. Characteristics and Methods of Example Operational Orbit Determination Systems (Page 2 of 4)

SYSTEM	GENERAL PERTURBATION FORMULATIONS	SPECIAL PERTURBATION FORMULATIONS	ESTIMATION METHODS
AOES Refs. 92, 125	• Extended f, g and h Series • Variation of Parameters Secular effects only for: J_2 (Kozai) Drag (e^0) Thrust (e^1) Outgassing (e^1) Long period effects for Solar radiation pressure	• Cowell • Variation of Parameters Element sets: -- Keplerian -- FG† -- Averaged Keplerian	• Differential Corrections (Batch and Sequential Batch)
SCC Refs. 1, 2, 3, 86, 101, 102, 108	• SGP4 (Lane) Brouwer w/power law atmosphere model	• Cowell	• Differential Corrections (Batch and Sequential Batch)

† FG elements are closely related to the equinoctial elements (see Reference 125).

Table 1.1. Characteristics and Methods of Example Operational Orbit Determination Systems (Page 3 of 4)

SYSTEM	GENERAL PERTURBATION FORMULATIONS	SPECIAL PERTURBATION FORMULATIONS	ESTIMATION METHODS
SCC (cont.)	• SGP (Aeronutronic Ford) Kozai Drag Expansions • SEGP Special case of SGP for vanishing inclinations • DP4 Special case of SGP4 to include first order luni-solar effects and tesseral/sectoral resonances for 12 h. and 24 h. orbits • IGP4 Special case of SGP4 for satellites with perigee heights >10,000 km or eccentricity >0.5		

Table 1.1. Characteristics and Methods of Example Operational Orbit Determination Systems (Page 4 of 4)

SYSTEM	GENERAL PERTURBATION FORMULATIONS	SPECIAL PERTURBATION FORMULATIONS	ESTIMATION METHODS
TRACE66 Ref. 127	• Variation of SGP/ SGP4 theories (see SCC above)	• Cowell • Cowell (time regularized)	• Differential Correction (Batch) • Differential Correction (Sequential) • Potter Square Root Filter (Sequential)
CELEST Ref. 122	• None	• Cowell	• Differential Correction (Sequential Batch)

1.2 Research Objective

Significant progress has been made in new computer hardware architectures which apply the concept of hardware parallelism to achieve extremely high effective computation rates. These multiprocessors, or parallel processors, are most effective in dealing with such computationally intensive tasks as linear system solutions, digital signal processing, digital image correlation and global weather prediction. However, current sequential numerical algorithms make poor use of the potential processing power available in these new hardware architectures. Use of such parallel processors in new applications requires considerable effort in the formulation of numerical algorithms appropriate to the target parallel processor hardware architecture.

It is the objective of this research to formulate parallel numerical algorithms to perform the basic orbit propagation and orbit estimation tasks required in an operational orbit determination system. These algorithms will be formulated to take best advantage of the architecture of the target computer hardware specified. The algorithms will be evaluated in terms of the accuracy of their orbit determination results in comparison to those

results produced by current high precision sequential methods. Additionally, the potential execution time performance improvement will be examined and certain estimated speed-up measures computed.

1.3 Prior Work

In many technical areas, computational requirements are increasing rapidly. To meet these increased processing requirements, new computer hardware and numerical algorithms have been developed. Prominent among the new methods are multiprocessor or parallel processor hardware architectures and their associated numerical algorithms. This section describes the recent advances in parallel processor architectures and algorithms.

1.3.1 Advanced Hardware Architectures

In the last twenty years, significant work has been done on multiprocessor or parallel processor hardware architectures. Thurber (142) points out that there are many approaches to classifying these computer architectures. A particularly convenient approach for this study is due to Flynn (78, 79), in which hardware architectures are divided into categories based upon their instruction and data streams. In this case, the term "stream" refers to

the sequence of data or instructions as seen by the machine during the execution of a program.

Flynn's architectural classifications are as follows:

1. Single Instruction Stream/Single Data Stream (SISD) -- a single control unit issuing instructions to be executed on a single stream of data.

2. Single Instruction Stream/Multiple Data Stream (SIMD) -- single control unit issuing instructions to be executed on several streams of data.

3. Multiple Instruction Stream/Single Data Stream (MISD) -- multiple control units issuing instructions to be executed on a single stream of data (not generally considered to be physically realizable).

4. Multiple Instruction Stream/Multiple Data Stream (MIMD) -- multiple control units issuing instructions to be executed independently on multiple data streams.

Thurber examines the general characteristics of these hardware architectures for both general purpose systems (142) and special purpose and scientific attached computing systems (143). Baer (24) also surveys the state of the art in SIMD and MIMD architectures. His work considers the increased complexity of the software design requirements for both the system software and the application software. Enslow (74) surveys multiprocessor organization schemes characterized by the various methods for interconnecting processors, memories and input/output channels within the machine. Such interconnection schemes may include buses, redundant buses and crossbar switching networks. Of particular interest in this survey is a detailed chronology and a table of characteristics of actual multiprocessor and parallel processor systems. The works of Baer (23), Miller (114), Noe and Nutt (120) and others examine several different approaches to the theoretical modelling of parallel computations. These theoretical methods include Petri nets and their extensions, graph theory models, parallel flowcharts and flow graph schemata.

In Flynn's evaluation of the effectiveness of computer organizations under his classification scheme (78), he further subdivided the SIMD architectures into array processors, pipelined processors and associative processors.

The array processor could be considered the classical synchronous parallel processor. In this architecture, one control unit is directly connected to an array of independent processing elements. Each processing element (PE) has its own registers and local storage, but only executes the single stream of instructions from the control unit on the particular stream of data in its local memory. The pipelined processor is a time-multiplexed version of the array processor in which the processing units are tailored to perform specific functions. These units are arranged in a "chain" or a "pipeline" and at every machine cycle, the results proceed through the pipeline. The associative processor is a derivative of the array processor in which the processing elements are not addressed directly but are activated by associative matching of certain data values. A survey of associative and parallel processors is presented by Thurber and Wald (144), in which they discuss the major classifications of computer architectures, the design tradeoffs associated with each, some potential applications in which use of parallel and associative architectures might be cost effective, and descriptions of several actual parallel processing machines. Yau and Fung (154) survey the area of associative processor architectures as to their characteristics, categorization and implementation. Ramamoorthy and Li (128) provide a comprehensive

definition of pipelining and describe its attendant architectural features.

The above references reflect attempts to classify and describe the general characteristics of these new hardware architectures. Several actual machines embodying the architectural principles described above are currently in use in specialized applications. Recently several descriptions and performance evaluations of these actual machines have been published. Russell (129) and Baskett and Keller (27) describe the CRAY-1 computer system, Stokes (138) discusses the Burroughs Scientific Processor, Martin (109) details PEPE and Flanders, et al. (77) describe the Distributed Array Processor. Additionally, new architectures such as Fung's massively parallel processing computer (81) are being postulated to handle other special-purpose processing needs.

1.3.2 Advanced Software Algorithms

It is clear that parallel processing architectures hold much promise as a means of bringing large amounts of processing capability to bear on a computational problem. However, the utility of these new machines is heavily dependent on the structure of the problem at hand. In fact,

Thurber and Wald (144) caution their readers that "these [associative and parallel processors] are special-purpose machines and any attempt to apply them to an incorrectly sized, or designed, problem is an exercise in futility." The interrelationships between the software processes; algorithms and data structures and the hardware architectures play a major role in the total effectiveness of the parallel processing system. To realize the full potential presented by parallel processors, the algorithmic structure of the software must be complementary to the hardware architecture, lest the advantages of parallel computation be lost.

Both Baer (23) and Kuck (95) have written excellent surveys of the programming implications of multiprocessing and the organizations of parallel machines. These articles include methods of detecting parallelisms, analyzing programs and modelling parallel computations. Principato (126) surveys techniques for specifying parallel processes in the context of real-time systems to express in the abstract the requirements for process synchronization and communication. DeWolf (62) indicates a method of detecting parallelisms within a process by the functional decomposition of that process using his data flow primitives.

The area of system software support for the parallel processor hardware architectures described above appears to be somewhat of a weak point in such systems. Operating system software and associated support software, such as compilers, do not appear to have kept pace with the rapid growth in the hardware technology. However, new strategies for operating system software for parallel processor computer systems have been studied by Nutt (121), with emphasis on input/output operations, resource allocation and interprocess communication. Additionally, Baer and Ellis (25) discuss a methodology for compiler design in which sequential structures are transformed to a pipeline of processes suitable for parallel processing.

Considerable attention has been focused on the form and structure of application numerical algorithms to be used on parallel processor machines. Perhaps the best reference on the relationship between hardware architectures and structure and performance of numerical algorithms is by Kuck, Lawrie and Sameh (96). Kung (98) describes the characteristics of synchronized and asynchronous parallel algorithms and provides several examples of each type of algorithm. Baudet, Brent and Kung (31) treat the parallel execution of a sequence of tasks on a multiprocessor system

and show how throughput may be improved by taking advantage of the fluctuations in execution times of the individual tasks.

Examples of specific parallel processor system applications are beginning to appear more often in the literature. Keller and Jameson (89) describe a vector processor solution to transonic flow calculations. Calahan (44) discusses the applications of vector processors in simulation-oriented research. Taylor and Widhopf (141) describe the potential application of parallel processors and pipelined computers to the computationally intensive tasks associated with fluid mechanical simulations. Travassos and Kaufman (87, 146, 147) have proposed two parallel algorithms for the solution of the nonlinear two point boundary value problems arising in their design of nonlinear flight controllers. Recent work by Larson and Tse (103) and Bucy and Senne (42) have formulated parallel algorithms for nonlinear filtering and estimation applications. Further examples of specific parallel processor architectures applied to radar signal processing, weather computations, matrix algebra and air traffic control may be found in Feng (76).

Significant progress has been made in applying parallel computer architectures to certain fundamental areas of computational numerical analysis. Miranker (115) presents a survey of such parallel numerical algorithms as of 1970. Included in that survey is a class of algorithms devised by Miranker for the parallel integration of ordinary differential equations (116). It is in the area of numerical solutions to linear systems of equations that parallel processors have performed best and algorithms for these tasks have received considerable attention. A linear algebraic algorithm for solution of triangular systems of equations is formulated by Sameh and Brent (131) while Sameh and Kuck (132) discuss other algorithms for solving dense systems of linear equations. Other areas of numerical software which are beginning to take advantage of new hardware architectures are such areas as manipulation of power series (33), evaluation of algebraic functions and expansions (99) and processing of recursive functions (43, 51).

1.3.3 Target Hardware Architecture

The above discussion indicates that parallel computer hardware and its associated software have the potential to increase significantly the performance/cost ratio for

certain kinds of problems (32). The types of applications best suited to these parallel architectures have been delineated above. One application which seems well suited to such system architectures and which has not been previously investigated is the orbit determination problem. As has been stated in the previous section, the objective of this research is to formulate parallel algorithms for the orbit determination application.

Before these algorithms can be formulated, however, it is necessary to have a target hardware architecture. This is because the algorithms must attempt to take full advantage of the available hardware parallelisms. In this study, the target hardware architecture will be that of a Single Instruction Stream/Multiple Data Stream (SIMD) array processor. The basic architectural framework of this target machine is depicted in Figure 1.1. It can be seen that the architecture consists of a single control unit (CU) which decodes instructions and issues them along a bus to numerous independent processing elements. Simultaneously in parallel, each processing element (PE) executes the instruction independently on the particular operands (data stream) in its local memory (LM). When necessary, results are transferred from local memory to

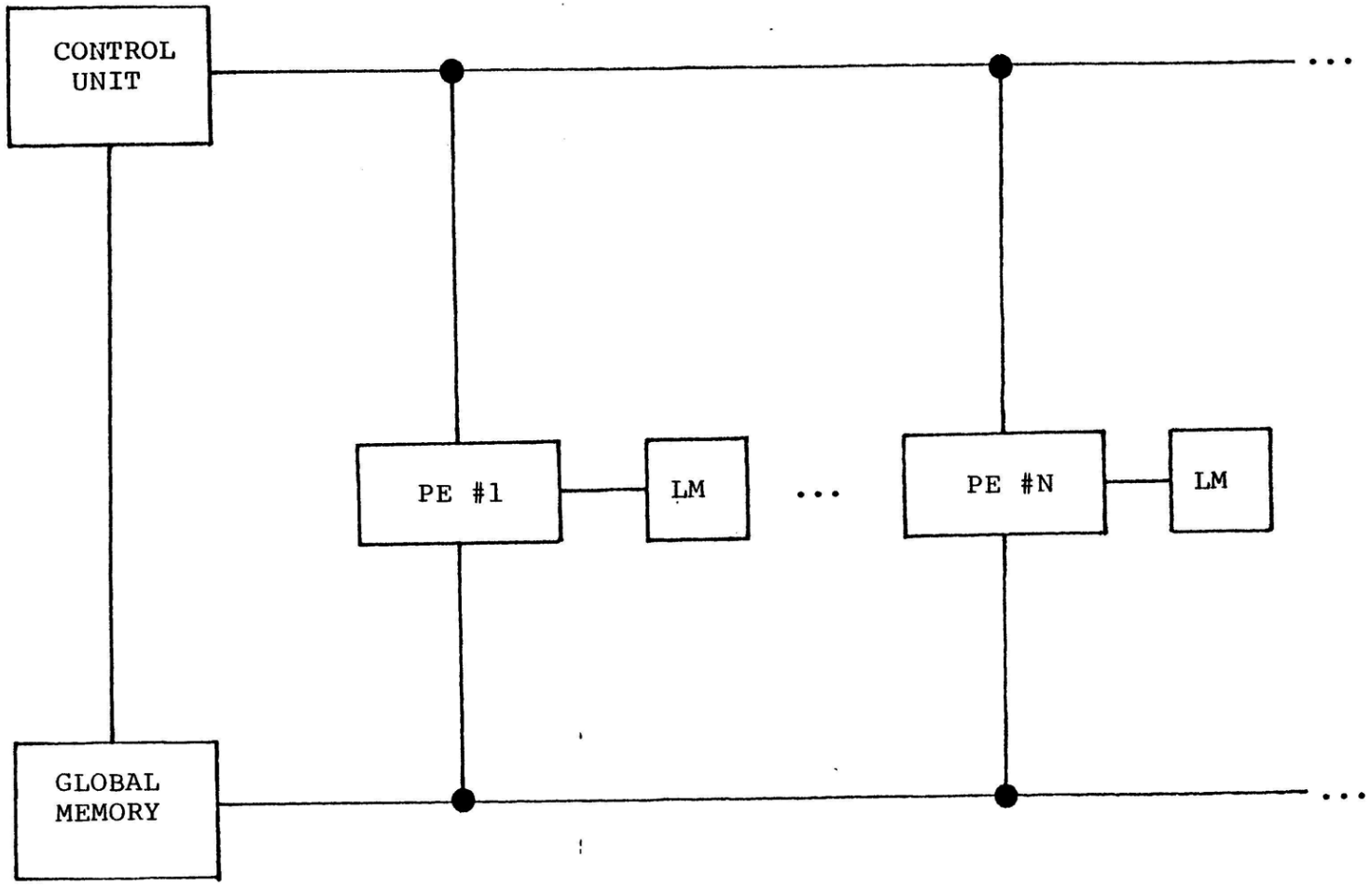


Figure 1.1. Single Instruction Stream/Multiple Data Stream (SIMD) Array Processor Target Hardware Architecture

global memory (GM) for processing by the main control unit. Clearly this is a highly synchronized process, with the control unit coordinating the efforts of each processing element with precise timing.

Projecting the specific architecture of the next generation of computer hardware is a very difficult task. There is certainly no guarantee that the architecture postulated here will be the architecture of the future. However, this architecture was selected because it is conceptually quite simple, especially with respect to the necessary control structure required (78). Additionally, recent literature has indicated a trend towards development of processors with very large numbers of processing elements connected in this type of architecture (71, 73, 100). Even if processors of this architecture are not constructed on a large scale, the state-of-the-art (145) appears to be such that a special purpose processor of this type could be constructed especially for the orbit determination application at a reasonable cost. For these reasons, the SIMD hardware architecture has been chosen as the target architecture for this study.

As stated in Section 1.2, the objective of this research is to formulate parallel algorithms for the orbit determination problem. Chapter 2 discusses a parallel orbit propagation algorithm ideally suited to the target SIMD hardware architecture. Chapter 3 develops a parallel orbit estimation algorithm which displays high accuracy and a robust radius of convergence. Estimates of the potential performance improvements are formulated in Chapter 4. Finally, Chapter 5 summarizes these developments, discusses the conclusions derived from this research and gives suggestions for future research.

CHAPTER 2

A PARALLEL ORBIT PROPAGATION ALGORITHM

The first task required of a parallel processing orbit determination system is to propagate certain initial parameters representative of the orbital trajectory over some time span using dynamical models appropriate to the mission and flight regime of the spacecraft. Because of the desire to exhibit high accuracy for all types of disturbing forces, a Special Perturbations formulation was chosen. This chapter will discuss the development of a parallel orbit propagation algorithm in which a high degree of parallelism is induced in the algorithm.

2.1 Algorithm Development

The algorithm presented here was developed originally by Clenshaw and Norton (52). The algorithm combines the technique of Picard iteration with that of Chebyshev series approximation. It has been applied in two cases dealing with the problem of artificial satellite orbit propagation. Feagin (75) applied this method to the integration of the Cowell equations of motion and implemented this capability

into the Goddard Trajectory Determination System research and development version. Chan (50) applied the same method to the orbit propagation problem formulated in Hamilton's equations of motion. In the first case, a set of three coupled, second order, ordinary differential equations are integrated, while in the second case six coupled, first order, ordinary differential equations are integrated. In either case the accuracy achieved was comparable to high precision Cowell ephemeris data, but the execution times were considerably longer.

Other recent efforts have applied this method to achieve high precision and high order planetary theories for long time spans. Carpenter (45) demonstrated the value of using Chebyshev series in the solution of the planetary equations of motion. He used Chebyshev approximations to the integrands of Musen's (118) perturbation formulation and determined subsequent series representations for the mutual gravitational attractions of the five outer planets. Broucke (37) applied the method to several planetary perturbation formulations to form series representations for heliocentric, rectangular displacements from a fixed Keplerian reference orbit. Similarly, Doggett (63) derived series representations for the right hand sides of Danby's formulation for the variations of the elliptic elements

for the planet Mars. Again in these cases it was possible to achieve high accuracies, but at the expense of extensive computer run time.

This Picard-Chebyshev method was chosen because it is ideally suited to the target parallel processor hardware architecture. The target SIMD architecture has numerous identical processing elements under the direction of a single control unit. As the control unit issues instructions, each processing element executes those instructions, but on separate streams of data. It will be shown that the approximation over long arcs of the right-hand members of the equations of motion by Chebyshev series uses fully this parallel structure. Additionally, this method uses the available hardware parallelism in the most advantageous way, that is in the parallel computation of the requisite force function evaluations over the long approximation arc. The force function evaluations are generally considered to be the most expensive computation involved in numerical integration, and the efficiency of many numerical integration methods is often measured by the number of force evaluations required for an integration arc. By this criterion, the Picard-Chebyshev integration algorithm will be shown to be extremely efficient when coupled with a SIMD hardware architecture.

2.1.1 Picard's Iterative Method

The method of Picard is a well known method (85) of solving the initial value problem using a successive substitution approach. The differential equation to be solved is of the general form (for the scalar case)

$$\frac{d\,y(t)}{dt} = \dot{y}(t) = F[t,y(t)] \qquad (2.1)$$
$$\text{with } y(t_0) = y_0$$

The first step in the method is to convert the differential equation into an integral equation. If t is replaced by the dummy variable σ in Equation (2.1) and both sides are integrated with respect to σ over the interval (t_0,t) the following results

$$\int_{t_o}^{t} \frac{d\,y(\sigma)}{d\sigma}\,d\sigma = \int_{t_o}^{t} F[\sigma,y(\sigma)]d\sigma$$

$$y(\sigma)\Big|_{t_o}^{t} = \int_{t_o}^{t} F[\sigma,y(\sigma)]d\sigma$$

$$y(t) - y(t_o) = \int_{t_o}^{t} F[\sigma,y(\sigma)]d\sigma$$

Applying the associated initial condition yields

$$y(t) = y_0 + \int_{t_0}^{t} F[\sigma, y(\sigma)]d\sigma \qquad (2.2)$$

A sequence of functions approximating $y(t)$ near $t = t_0$ can then be found by the following iterative procedure

$$y^{[k+1]}(t) = y_0 + \int_{t_0}^{t} F[\sigma, y^{[k]}(\sigma)]d\sigma \qquad (2.3)$$

The zeroth iterate $y^{[0]}(t)$ can be taken to be the initial condition $y(t_o) = y_0 = y^{[0]}(t)$. If the function $y(t)$ is continuous over the region $t_0 \leq t \leq t_f$ and satisfies the Lipschitz condition, then it has been shown (53, 84) that a sequence of approximating functions exist and converge uniformly over the interval to the unique solution of the previously stated initial value problem (Picard-Lindelöf theorem).

2.1.1.1 Example of Picard's Method

Consider the scalar linear first-order, ordinary differential equation

$$\frac{dy}{dt} = t^2 - y \; ; \qquad y(o) = 1 \qquad (2.4)$$

The homogeneous part of the exact solution is found to be

$$y_H(t) = c_1 e^{-t}$$

By applying the Method of Undetermined Coefficients, a particular solution of the form

$$y_P(t) = t^2 - 2t + 2$$

is found. The complete solution then is the sum of the homogeneous and the particular solutions, such that

$$y(t) = y_H(t) + y_P(t) = 2 - 2t + t^2 + c_1 e^{-t}$$

By applying the condition $y(0) = 1$, the constant c_1 is determined to be -1. Therefore, the initial value problem specified by Equation (2.4) has the exact solution

$$y(t) = 2 - 2t + t^2 - e^{-t} \qquad (2.5)$$

To find an approximation to this solution by Picard's iterative method, Equation (2.3) is used. For this example

$$F[\sigma, y^{[k]}(\sigma)] = \sigma^2 - y^{[k]}(\sigma)$$

$$y_0 = 1$$

Thus, the successive approximation formula becomes

$$y^{[k+1]}(t) = 1 + \int_0^t [\sigma^2 - y^{[k]}(\sigma)]d\sigma \qquad (2.6)$$

By setting the zeroth iterate of the solution equal to the initial condition, the first iterate of the approximate solution can be found by direct integration of Equation (2.6) as

$$y^{[1]}(t) = 1 + \int_0^t [\sigma^2 - y^{[0]}(\sigma)]d\sigma$$

But $$y^{[0]}(\sigma) = y(t_0) = y_0 = 1$$

$$y^{[1]} = 1 + \int_0^t [\sigma^2 - 1]d\sigma$$

$$= 1 + [\tfrac{1}{3}\sigma^3 - \sigma]\Big|_0^t$$

$$\therefore \quad y^{[1]}(t) = 1 - t + \tfrac{1}{3}t^3 \qquad (2.7a)$$

Similarly, the second iterate is obtained by substituting Equation (2.7a) into Equation (2.6) and directly integrating as follows

$$y^{[2]}(t) = 1 + \int_0^t [\sigma^2 - y^{[1]}(\sigma)]d\sigma$$

$$= 1 + \int_0^t [\sigma^2 - 1 + \sigma - \tfrac{1}{3}\sigma^3]d\sigma$$

$$= 1 + [-\sigma + \tfrac{1}{2}\sigma^2 + \tfrac{1}{3}\sigma^3 - \tfrac{1}{12}\sigma^4]\Big|_0^t$$

$$\therefore \quad y^{[2]}(t) = 1 - t + \tfrac{1}{2}t^2 + \tfrac{1}{3}t^3 - \tfrac{1}{12}t^4 \qquad (2.7b)$$

Further application of this technique yields the following successive approximations to the solution of Equation (2.4)

$$y^{[3]}(t) = 1 - t + \tfrac{1}{2}t^2 + \tfrac{1}{6}t^3 - \tfrac{1}{12}t^4 + \tfrac{1}{60}t^5 \qquad (2.7c)$$

$$y^{[4]}(t) = 1 - t + \tfrac{1}{2}t^2 + \tfrac{1}{6}t^3 - \tfrac{1}{24}t^4 + \tfrac{1}{60}t^5 - \tfrac{1}{360}t^6 \qquad (2.7d)$$

$$y^{[5]}(t) = 1 - t + \tfrac{1}{2}t^2 + \tfrac{1}{6}t^3 - \tfrac{1}{24}t^4 + \tfrac{1}{120}t^5 - \tfrac{1}{360}t^6 + \tfrac{1}{2520}t^7 \qquad (2.7e)$$

$$y^{[6]}(t) = 1 - t + \frac{1}{2}t^2 + \frac{1}{6}t^3 - \frac{1}{24}t^4 + \frac{1}{120}t^5$$

$$- \frac{1}{720}t^6 + \frac{1}{2520}t^7 - \frac{1}{20,160}t^8$$

(2.7f)

.
.
.

To compare this sequence of approximations with the exact solution, recall that the exact solution is

$$y(t) = 2 - 2t + t^2 - e^{-t}$$

By expanding e^{-t} in its series representation, the following form of the solution may be found

Since $$e^z = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!} + \ldots$$

for all real values of z

$$e^{-z} = 1 - z + \frac{z^2}{2!} - \frac{z^3}{3!} + \frac{z^4}{4!} - \ldots$$

for all real values of z

Then $$y(t) = 2 - 2t + t^2 - [1 - t + \frac{1}{2} t^2 - \frac{1}{6} t^3 + \frac{1}{24} t^4 - \frac{1}{120} t^5 + \frac{1}{720} t^6 - \frac{1}{5040} t^7 + \frac{1}{40,320} t^8 + \ldots]$$

$$\therefore \quad y(t) = 1 - t + \frac{1}{2} t^2 + \frac{1}{6} t^3 - \frac{1}{24} t^4 + \frac{1}{120} t^5 - \frac{1}{720} t^6 + \frac{1}{5040} t^7 - \frac{1}{40320} t^8 + \ldots \tag{2.8}$$

By comparing Equation (2.8) with Equation (2.7f), for example, it is seen that the sixth iteration of the Picard method has resulted in a portion of a series solution whose coefficients agree exactly with the coefficients of Equation (2.8) through the sixth power of the independent variable t. Further iterations will result in agreement of the coefficients of higher powers of t. As the iteration number k approaches infinity, the kth approximation to the solution approaches the exact solution, i.e.,

$$|y(t) - y^{[k]}(t)| \rightarrow 0 \text{ as } k \rightarrow \infty$$

as guaranteed by the Picard-Lindelöf Theorem.

Table 2.1 indicates the numerical behavior of the approximate solutions of Equation (2.4) over the interval $0 \leq t \leq 1$. It is seen that the interval of integration can significantly influence the convergence rate of the Picard iterative method. Table 2.1 shows that $y^{[4]}(t)$ agrees with the exact solution to five places (rounded) on the interval $0 \leq t \leq 0.20$. However, to get the same five place agreement over the interval $0 \leq t \leq 0.60$, two more iterations would be required.

Additionally, in the numerical solution of perturbed nonlinear equations it is sometimes possible to use the exact solution of the unperturbed problem as the zeroth iteration in Picard's method. The number of iterations to reach convergence is then dependent on the "goodness" of the zeroth iteration with respect to the true solution of the perturbed problem. For weakly perturbed problems, the unperturbed solution will closely approximate the true solution and few iterations will be required to reach convergence. However, for strongly perturbed problems the true solution of the perturbed problem may differ greatly from the unperturbed solution and a large number of Picard iterations would be required for convergence. Thus, the rate of convergence of the Picard method is controlled by the integra-

Table 2.1

Comparison of Picard Iteration Method Approximations to Exact Solution of Equation 2.4

t	Exact	$y^{[1]}(t)$	$y^{[2]}(t)$	$y^{[3]}(t)$	$y^{[4]}(t)$	$y^{[5]}(t)$	$y^{[6]}(t)$
0.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0.1	0.905163	0.900333	0.905325	0.905159	0.905163	0.905163	0.905163
0.2	0.821269	0.802667	0.822533	0.821205	0.821272	0.821269	0.821269
0.3	0.749182	0.709000	0.753325	0.748866	0.749201	0.749181	0.749182
0.4	0.689680	0.621333	0.699200	0.688704	0.689759	0.689675	0.689680
0.5	0.643469	0.541667	0.661458	0.641146	0.643707	0.643449	0.643471
0.6	0.611188	0.472000	0.641206	0.606496	0.611766	0.611130	0.611193
0.7	0.593415	0.414333	0.639325	0.584960	0.594637	0.593269	0.593430
0.8	0.590617	0.370667	0.656533	0.576661	0.593000	0.590352	0.590708
0.9	0.603430	0.343000	0.693325	0.581667	0.607528	0.602797	0.603514
1.0	0.632121	0.333333	0.750000	0.600000	0.638889	0.630952	0.632292

tion span of interest and by the "goodness" of the zeroth approximation to the solution.

2.1.1.2 Extension to Systems of Equations of First Order

The previous discussion considered only a single (scalar) first order, ordinary differential equation. Henrici (84) and others (53) have shown that the scalar results extend to systems of ordinary differential equations of first order of the form

$$\begin{aligned} \dot{y}^1(t) &= F^1(t,y^1,y^2,\ldots,y^n) \\ \dot{y}^2(t) &= F^2(t,y^1,y^2,\ldots,y^n) \\ &\vdots \qquad\qquad \vdots \\ \dot{y}^n(t) &= F^n(t,y^1,y^2,\ldots,y^n) \end{aligned} \tag{2.9}$$

The inclusion of an appropriate set of initial conditions on Equations (2.9)

$$\begin{aligned} y^1(t_o) &= y^1{}_o \\ y^2(t_o) &= y^2{}_o \\ &\vdots \\ y^n(t_o) &= y^n{}_o \end{aligned} \tag{2.10}$$

completes the specification of the initial value problem in this case. These expressions are more easily handled in vector form. If the vector $\vec{y}(t)$ is defined as

$$\vec{y}(t) = \begin{pmatrix} y^1(t) \\ y^2(t) \\ \vdots \\ y^n(t) \end{pmatrix} \tag{2.11a}$$

and the vector valued function $\vec{F}$ is defined as

$$\vec{F}(t,\vec{y}) = \begin{pmatrix} F^1(t,\vec{y}) \\ F^2(t,\vec{y}) \\ \vdots \\ F^n(t,\vec{y}) \end{pmatrix} \tag{2.11b}$$

and the initial condition vector $\vec{y}(t_o)$ is defined as

$$\vec{y}(t_o) = \begin{pmatrix} y^1(t_o) \\ y^2(t_o) \\ \vdots \\ y^n(t_o) \end{pmatrix} = \vec{y}_o \tag{2.11c}$$

then the initial value problem for the system of Equations (2.9) and (2.10) can be simply stated

$$\dot{\vec{y}}(t) = \vec{F}[t,\vec{y}(t)] \; ; \quad \vec{y}(t_o) = \vec{y}_o \qquad (2.12)$$

Henrici has directly extended the theorems of existence, uniqueness and convergence of solutions for the scalar case of Equation (2.1) to the vector case of Equation (2.12). He has also shown that the solution to Equation (2.12) may be expressed as

$$\vec{y}^{[k+1]}(t) = \vec{y}_o + \int_{t_o}^{t} \vec{F}[\sigma,\vec{y}^{[k]}(\sigma)]d\sigma \qquad (2.13)$$

This vector form of the Picard iteration method will be used in the subsequent discussions.

2.1.1.3 Practical Limitations

Hildebrand (85) points out that, while the method is theoretically very important, it has definite practical limitations. If $F(t,y)$ is expressed as an analytical function, performing the required direct integration of $F(t,y^{[k]})$ may become increasingly difficult with each successive iteration for any function $F(t,y)$ which is at all complex. If $F(t,y)$ is given as a set of tabular values, and not as an analytical function, then the Picard iterative method is not directly applicable.

In both of the above cases, it may be possible to approximate the integrands of Equations (2.2) and (2.13) by some type of a series representation. This approach should result in representations for the right hand sides of Equations (2.1) and (2.12) which are analytically more tractable for integration than the original integrands, or which approximate (or interpolate) the integrands at the given tabular values. In this case, a convenient approximation may be formed by using a series of orthogonal polynomials, specifically Chebyshev polynomials.

2.1.2 Chebyshev Series Approximation of Right Hand Members

The approximation of explicit functions or the interpolation of tabular data values are well known applications of a family of orthogonal polynomials known as Chebyshev polynomials. The properties of Chebyshev polynomials and series of Chebyshev polynomials are widely documented (34, 35, 55, 80, 83, 85). Recent works by several authors have applied Chebyshev series to the representation of satellite ephemeris data. Corio (56) used Chebyshev series to approximate satellite inertial Cartesian position and velocity components over a one day time span. Deprit, Pickard and Poplarchek (61) developed a method of compressing satellite ephemeris data using Chebyshev polynomials of moderate

degree. Their method features the capabilities to automatically find the minimum degree polynomial to achieve a given accuracy level and to find the maximum error in this fit. In both of the above cases, the position and velocity data were fitted directly and no integration of the series representations was required.

The following sections summarize existing derivations of those properties used in this formulation of the Picard-Chebyshev integration method. These derivations are well known and are included here for completeness. Additionally, the formulas used for approximating the integrands in this version of the integration formulation are described. The particular choice of orthogonality (or "special") points is discussed.

2.1.2.1 Relevant Properties of Chebyshev Polynomials

Before discussing orthogonal functions, one must define the concept of a scalar product. Given an interval $a \leq z \leq b$ and a weighting function $w(z)$ defined on this interval and positive there, then the scalar product of two function $g(z)$ and $h(z)$ (both defined on the given interval) with respect to $w(z)$ may be defined in the following two ways:

Continuous Case:

$$\langle wgh \rangle = \int_a^b w(z)\, g(z)\, h(z)\, dz \qquad (2.14a)$$

Discrete Case:

$$\langle wgh \rangle = \sum_{n=1}^{N} w(z_n)\, g(z_n)\, h(z_n) \qquad 2.14b)$$

With the scalar product defined as above, the two functions $g(z)$ and $h(z)$ are said to be orthogonal with respect to $w(z)$ if

$$\langle wgh \rangle = 0 \qquad (2.15)$$

over the domain of definition of the functions $g(z)$, $h(z)$ and $w(z)$.

In this case, it is desired to approximate a function as a linear combination of a set of polynomial functions and that the members of this set be mutually orthogonal. The rth member of a set of orthogonal polynomials may be found be requiring that member $\phi_r(z)$ be orthogonal to any arbitrary polynomial of degree inferior to r, i.e.,

$$\int_a^b w(z)\, \phi_r(z)\, q_{r-1}(z)\, dz = 0$$

where $q_{r-1}(z)$ is an arbitrary polynomial of degree r-1 or less.

In cases where errors near the ends of the approximation interval [a,b] are important, a weighting function of the form

$$w(z) = \frac{1}{\sqrt{(z - a)(b - z)}} \tag{2.16}$$

can be used. Notice that the weighting function gets large as z approaches either end point. A linear change of variables is used to transform the interval of representation $a \leq z \leq b$ to the interval $-1 \leq \chi \leq 1$ by

$$\chi = 2\left(\frac{z - a}{b - a}\right) - 1 \tag{2.17}$$

$$d\chi = \frac{2}{b - a}\, dz \tag{2.18}$$

and $z = a \Rightarrow \chi = -1$

$z = b \Rightarrow \chi = +1$

Using this change of variables, the weighting function in the new variable χ becomes

$$w(z) = \frac{1}{\sqrt{(z - a)(b - z)}} = \left(\frac{2}{b - a}\right) \frac{1}{\sqrt{1 - \chi^2}}$$

$$= w(\chi) \qquad (2.19)$$

The transformed orthogonality condition takes the form

$$\int_a^b w(z)\ \phi_r(z)\ q_{r-1}(z)\ dz$$

$$= \left(\frac{b - a}{2}\right) \int_{-1}^{1} \frac{\phi_r(\chi)\ q_{r-1}(\chi)}{\sqrt{1 - \chi^2}}\ d\chi = 0 \qquad (2.20)$$

If the new variable χ is defined by

$$\chi = \cos \Theta \qquad (2.21a)$$

then $$\Theta = \cos^{-1} \chi \qquad (2.21b)$$

and $$d\chi = -\sin \Theta\ d\Theta \qquad (2.21c)$$

The orthogonality condition becomes

$$\int_{-1}^{1} \frac{\phi_r(\chi)\, q_{r-1}(\chi)}{\sqrt{1-\chi^2}}\, d\chi$$

$$= \int_{\pi}^{0} \frac{\phi_r(\cos\theta)\, q_{r-1}(\cos\theta)}{\sqrt{1-\cos^2\theta}} (-\sin\theta)\, d\theta$$

$$= -\int_{0}^{\pi} \frac{\phi_r(\cos\theta)\, q_{r-1}(\cos\theta)}{\sin\theta} (-\sin\theta)\, d\theta$$

$$\int_{0}^{\pi} \phi_r(\cos\theta)\, q_{r-1}(\cos\theta)\, d\theta = 0 \qquad (2.22)$$

Because cos $k\theta$ can be expressed as a polynomial of degree k in cos θ, $\phi_r(\cos\theta)$ will satisfy Equation (2.22) if and only if

$$\int_{0}^{\pi} \phi_r(\cos\theta) \cos k\theta\, d\theta = 0$$

for $k = 0,1,2,\ldots,r-1$

The function $\phi_r(\cos\theta) = C_r \cos r\theta$ satisfies the above condition where C_r is an arbitrary multiplicative constant. In terms of the independent variable χ, $\phi_r(\chi) = C_r \cos(r \cos^{-1}\chi)$ With $C_r = 1$, these polynomials are called Chebyshev polynomials of the first kind, denoted by $T_r(\chi)$

$$\phi_r(\chi) = T_r(\chi) = \cos(r \cos^{-1} \chi) \qquad (2.23)$$

and they possess the orthogonality property

$$\int_{-1}^{1} \frac{T_m(\chi)\, T_n(\chi)}{\sqrt{1 - \chi^2}}\, d\chi = 0 \,, \quad m \neq n \qquad (2.24a)$$

By making the substitutions indicated by Equations (2.21a) and (2.21c), Equation (2.24a) may be expanded as follows

$$\int_{-1}^{1} \frac{T_m(\chi)T_n(\chi)}{\sqrt{1 - \chi^2}}\, d\chi = \int_{\pi}^{0} \frac{\cos m\theta \cos n\theta}{\sin \theta} (-\sin \theta)\, d\theta$$

$$= \int_{0}^{\pi} \cos m\theta \cos n\theta \, d\theta$$

$$= \begin{cases} \int_0^{\pi} 1 \, d\theta & , \; m = n = 0 \\ \int_0^{\pi} \cos^2 m\theta \, d\theta & , \; m = n \neq 0 \\ \int_0^{\pi} \cos m\theta \cos n\theta \, d\theta & , \; m \neq n \end{cases}$$

$$= \begin{cases} \theta \,|_0^{\pi} & m = n = 0 \\ [\frac{\theta}{2} + \frac{1}{4m} \sin 2m\theta]\,|_0^{\pi} & m = n \neq 0 \\ \frac{\sin(m+n)\theta}{2(m+n)} + \frac{\sin(m-n)\theta}{2(m-n)} \,|_0^{\pi}, & m \neq n \end{cases}$$

$$\therefore \quad \int_{-1}^{1} \frac{T_m(\chi)T_n(\chi)}{\sqrt{1-\chi^2}}\, d\chi = \begin{cases} \pi\,, & m = n = 0 \\ \frac{\pi}{2}\,, & m = n \neq 0 \\ 0\,, & m \neq n \end{cases} \qquad (2.24b)$$

Using the basic definition of Equation (2.23), the first few Chebyshev polynomials can be computed

$$T_0(\chi) = \cos(0) = 1 \qquad (2.25a)$$

$$T_1(\chi) = \cos(\cos^{-1}\chi) = \chi \qquad (2.25b)$$

$$T_2(\chi) = \cos(2\cos^{-1}\chi)$$
$$= 2\cos^2(\cos^{-1}\chi) - 1$$

$$T_2(\chi) = 2\chi^2 - 1 \qquad (2.25c)$$

$$T_3(\chi) = \cos(3 \cos^{-1}\chi)$$

$$= 4 \cos^3(\cos^{-1}\chi) - 3 \cos(\cos^{-1}\chi)$$

$$T_3(\chi) = 4\chi^3 - 3\chi \qquad (2.25d)$$

Three Term Recursion Relation

Higher order Chebyshev polynomials can be calculated using a three term recursion relation. From the definition of Equation (2.23) and the change of variable $\chi = \cos\theta$, the $n + 1^{st}$ Chebyshev polynomial may be written as

$$T_{n+1}(\cos\theta) = \cos[(n + 1)\theta]$$

$$= \cos(n\theta + \theta)$$

$$T_{n+1}(\cos\theta) = \cos n\theta \cos\theta - \sin n\theta \sin\theta \qquad (2.26)$$

Similarly, the $n - 1^{st}$ Chebyshev polynomial is given by

$$T_{n-1}(\cos\theta) = \cos[(n - 1)\theta]$$

$$= \cos(n\theta - \theta)$$

$$T_{n-1}(\cos\theta) = \cos n\theta \cos\theta + \sin n\theta \sin\theta \qquad (2.27)$$

Adding Equations (2.26) and (2.27) yields

$$T_{n+1}(\cos\theta) + T_{n-1}(\cos\theta) = 2\cos n\theta \cos\theta$$

But $$\cos n\theta = T_n(\cos\theta)$$

$$\therefore \quad T_{n+1}(\cos\theta) = 2\cos\theta\, T_n(\cos\theta) - T_{n-1}(\cos\theta)$$

Using Equation (2.21a) to convert to the variable χ yields the required three term recursion

$$T_{n+1}(\chi) = 2\chi\, T_n(\chi) - T_{n-1}(\chi) \;,\; n \geq 1 \qquad (2.28)$$

with the starting values

$$T_0(\chi) = 1$$

$$T_1(\chi) = \chi$$

Discrete Orthogonality Points

It is another property of Chebyshev polynomials that they are orthogonal with respect to summation over the following discrete set of N + 1 points χ_i, equally spaced in θ

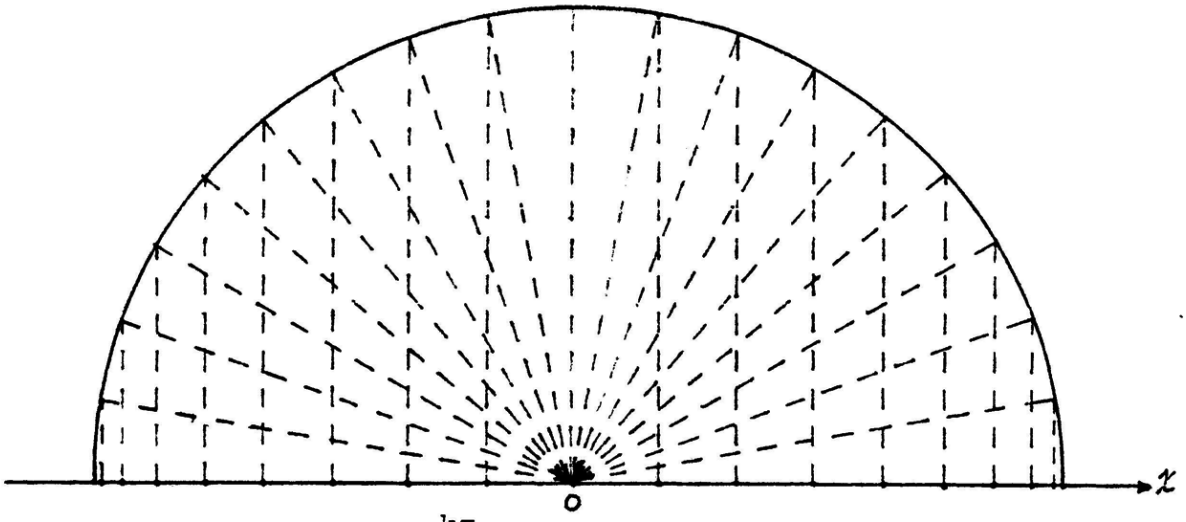
$$\chi_i = \cos \Theta_i \tag{2.29}$$

where $\Theta_i = 0, \frac{\pi}{N}, \ldots, \frac{(N-1)\pi}{N}, \pi$

These points are graphically represented in Figure 2.1 (83) for N = 18. Certain characteristics should be noted about this set of discrete orthogonality points. First, the points, although equally spaced in Θ, are not equally spaced in χ. It is seen that more points are distributed towards the ends of the interval $-1 \leq \chi \leq 1$. This is the discrete analog to the continuous weighting function

$$w(\chi) = \frac{1}{\sqrt{1 - \chi^2}}$$

and is important for counteracting the well known Gibbs phenomena found in classical Fourier Series approximations. Additionally, the location of these points are totally determined once the value of N is selected. That is to say, the location of point χ_ℓ does not directly depend on the location of points $\chi_{\ell-1}$ or $\chi_{\ell+1}$ or any other orthogonality point for a given value of N. Therefore, if an integration is being performed over the interval $-1 \leq \chi \leq 1$ and the right hand members must be computed at each orthogonality



$$\theta_k = \frac{k\pi}{18} \ , \ k = 0,1,\ldots,18$$

$$\chi_k = \cos \theta_k$$

Figure 2.1. Set of Chebyshev Orthogonality Points

point, the independence of these points makes the force function computations independent and hence candidates for parallel processing. This is in contrast to standard single step or multi-step integration techniques in which the current evaluation of the force function depends on one or more prior values of the force function and/or the independent variable. Finally, it may be noted that these values of χ are the points at which the Nth Chebyshev polynomial takes on its maximum and minimum values.

For this choice of discrete orthogonality points the Chebyshev polynomials possess the property of orthogonality

with respect to summation. In this case, the orthogonality condition is given by

$$\frac{1}{2} T_m(-1)T_n(-1) + \sum_{k=1}^{N-1} T_m(\chi_k)T_n(\chi_k) + \frac{1}{2} T_m(1)T_n(1) = \begin{cases} 0 , & m \neq n \\ \frac{N}{2} , & m = n \neq 0 \\ N , & m = n = 0 \end{cases} \tag{2.30}$$

where N = number of orthogonality points minus 1.

An alternate set of N orthogonality points exists of the form

$$\chi'_i = \cos \theta'_i \tag{2.31}$$

where $$\theta'_i = \frac{(2i - 1)\pi}{2N} , \quad i = 1,2,\ldots,N$$

These points are the mid points of the intervals defined by the previous set of orthogonality points and therefore are the zeroes of the N^{th} Chebyshev polynomial. The corresponding orthogonality condition is

$$\sum_{k=1}^{N} T_m(\chi_k) T_n(\chi_k) = \begin{cases} 0 , & m \neq n \\ \frac{N}{2} , & m = n \neq 0 \\ N , & m = n = 0 \end{cases} \qquad (2.32)$$

Either set of orthogonality points and corresponding orthogonality condition may be used in constructing Chebyshev approximating or interpolating series, but the series representations will have slightly different forms, as will be shown later.

Multiplication of Chebyshev Polynomials

The simple relation for the multiplication of two Chebyshev polynomials may be found from the trigonometric relation

$$\cos \alpha \cos \beta = \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta) \qquad (2.33)$$

if $\alpha = m\Theta$ and $\beta = n\Theta$, then Equation (2.33) becomes

$$\cos m\Theta \cos n\Theta = \frac{1}{2} \cos(m + n)\Theta + \frac{1}{2} \cos(m - n)\Theta$$

But $$\cos(-\beta) = \cos(\beta)$$

$$\cos(m - n)\Theta = \cos(n - m)\Theta = \cos(|m - n|\Theta)$$

Returning to χ as the independent variable, using Equations (2.21a) and (2.23), Equation (2.33) becomes

$$T_m(\chi)T_n(\chi) = \frac{1}{2}\left[T_{m+n}(\chi) + T_{|m-n|}(\chi)\right] \qquad (2.34)$$

which allows replacement of the product of two Chebyshev polynomials by the sum of two other Chebyshev polynomials.

Differentiation of Chebyshev Polynomials

The relation for differentiating Chebyshev polynomials is most easily derived from the trigonometric nature of these polynomials. The basic definition of the $n + 1^{st}$ Chebyshev polynomial is

$$T_{n+1}(\chi) = \cos[(n + 1)\cos^{-1}\chi] = \cos[(n + 1)\theta]$$

Taking the derivative of this with respect to χ yields

$$\frac{d}{d\chi} T_{n+1}(\chi) = \frac{d}{d\theta}\cos[(n + 1)\theta]\frac{d\theta}{d\chi} \qquad (2.35)$$

From Equation (2.21c), $\frac{d\theta}{d\chi} = -\frac{1}{\sin\theta}$. Using a prime to represent differentiation with respect to χ, Equation (2.35) becomes

$$\frac{T_{n+1}'(\chi)}{n+1} = \frac{\sin[(n+1)\theta]}{\sin\theta} \tag{2.36}$$

Applying the same process to $T_{n-1}(\chi)$ yields a similar expression

$$\frac{T_{n-1}'(\chi)}{n-1} = \frac{\sin[(n-1)\theta]}{\sin\theta} \tag{2.37}$$

Subtracting Equation (2.37) from Equation (2.36) gives

$$\frac{T_{n+1}'(\chi)}{n+1} - \frac{T_{n-1}'(\chi)}{n-1} = \frac{1}{\sin\theta}\{\sin[(n+1)\theta] - \sin[(n-1)\theta]\} \tag{2.38}$$

But $\sin[(n+1)\theta] = \sin n\theta \cos\theta + \cos n\theta \sin\theta$

and $\sin[(n-1)\theta] = \sin n\theta \cos\theta - \cos n\theta \sin\theta$

$\therefore$ $\sin[(n+1)\theta] - \sin[(n-1)\theta] = 2 \cos n\theta \sin\theta$

Substituting this into Equation (2.38) and using Equation (2.21b), the derivative relation is found to be

$$\frac{T_{n+1}'(\chi)}{n+1} - \frac{T_{n-1}'(\chi)}{n-1} = 2\,T_n(\chi)\,,\ n \geq 2 \tag{2.39}$$

This equation can be used as a recursion to find the value of $T_{n+1}'(\chi)$, given the starting values

$$T_0'(\chi) = 0$$
$$T_1'(\chi) = 1 = T_0$$
$$T_2'(\chi) = 4\chi = 4T_1$$

Integration of Chebyshev Polynomials

The relation for integrating a Chebyshev polynomial can be most easily found by direct integration of the derivative relation just derived as Equation (2.39). If both sides of Equation (2.39) are integrated with respect to the Chebyshev variable χ, then the result is

$$2\int T_n(\chi)d\chi = \frac{1}{n+1}\int T_{n+1}'(\chi)d\chi - \frac{1}{n-1}\int T_{n-1}'(\chi)d\chi \quad \text{for } n \geq 2$$

$$= \frac{T_{n+1}(\chi)}{n+1} - \frac{T_{n-1}(\chi)}{n-1}$$

$$\therefore \quad \int T_n(\chi)d\chi = \frac{1}{2}\left[\frac{T_{n+1}(\chi)}{n+1} - \frac{T_{n-1}(\chi)}{n-1}\right] + c_1 \qquad (2.40)$$

with $\int T_0(\chi)d\chi = T_1(\chi) + c_2$

$$\int T_1(\chi)d\chi = \frac{1}{4}T_2(\chi) + c_3$$

where the c's are arbitrary constants of integration. These simple relations will be applied later to derive the formulae for direct integration of a series of Chebyshev polynomials. The result will give the series coefficients of the integrated series in terms of the coefficients of the Chebyshev series representing the integrand.

2.1.2.2 Function Approximation by Discrete Chebyshev Series

The preceding section displayed some of the basic properties of Chebyshev polynomials. This section will briefly summarize how these polynomials may be used to represent a given function over some region. This summary includes information contained in Chapters 1 and 2 of Fox and Parker (80).

It is desired to approximate a given function $f(\chi)$ by a linear combination of a set of orthogonal polynomials. That is

$$f(\chi) \simeq \sum_{k=0}^{M} A_k \phi_k(\chi) \triangleq P_M(\chi) \qquad (2.41)$$

where $\phi_0(\chi)$, $\phi_1(\chi)$,..., $\phi_M(\chi)$ are M + 1 linearly independent, mutually orthogonal polynomial functions. Additionally, it is desired that the error between the function and

its series representation be minimum in the sense that the aggregate (sum or integral) of the weighted squared error over the interval of representation be minimum. If $e_M(\chi)$ represents the error and $w(\chi)$ the appropriate weighting function over the interval of representation, then the desired approximating polynomial will be that for which

$$\left\langle w(\chi) e_M^2(\chi) \right\rangle = \left\langle w(\chi) \left[f(\chi) - \sum_{k=0}^{M} A_k \phi_k(\chi) \right]^2 \right\rangle \tag{2.42}$$

$$= \text{minimum}$$

Since the approximating polynomial is a function of the series coefficients A_k $(k = 0,1,\ldots,M)$ as well as χ, a necessary condition for the weighted square error to be minimum is that the first partial derivatives of the squared error with respect to the A_k's vanish. Thus, the necessary condition for a minimum becomes

$$\frac{\partial}{\partial A_r} \left\langle w(\chi) e_M^2(\chi) \right\rangle = \frac{\partial}{\partial A_r} \left\langle w(\chi) \left[f(\chi) - \sum_{k=0}^{M} A_k \phi_k(\chi) \right]^2 \right\rangle \equiv 0 \tag{2.43}$$

$$0 = \left\langle 2w(\chi) \left[f(\chi) - \sum_{k=0}^{M} A_k \phi_k(\chi) \right] \left[-\phi_r(\chi) \right] \right\rangle$$

$$= \left\langle w(\chi) \phi_r(\chi) f(\chi) - w(\chi) \phi_r(\chi) \sum_{k=0}^{M} A_k \phi_k(\chi) \right\rangle$$

Hence

$$\left\langle w(\chi)\ \phi_r(\chi)\ \sum_{k=0}^{M} A_k \phi_k(\chi) \right\rangle = \left\langle w(\chi)\phi_r(\chi)f(\chi) \right\rangle$$

Since $w(\chi)\ \phi_r(\chi)$ multiplies every term in the summation, it may be brought inside the summation and since the A_k's are scalars, they may be taken outside the scalar product. The result is

$$\sum_{k=0}^{M} A_k \left\langle w(\chi)\phi_r(\chi)\phi_k(\chi) \right\rangle = \left\langle w(\chi)\phi_r(\chi)f(\chi) \right\rangle \tag{2.44}$$

$$\text{for } r = 0,1,\ldots,M$$

These are M + 1 simultaneous linear equations in the M + 1 unknown parameters $A_0, A_1, \ldots, A_M$ and are known as the <u>normal equations</u>. In expanded form these M + 1 equations become

$$\begin{aligned}
&A_0\langle w\phi_0\phi_0\rangle + A_1\langle w\phi_0\phi_1\rangle + \ldots + A_M\langle w\phi_0\phi_M\rangle = \langle w\phi_o f\rangle \\
&A_0\langle w\phi_1\phi_0\rangle + A_1\langle w\phi_1\phi_1\rangle + \ldots + A_M\langle w\phi_1\phi_M\rangle = \langle w\phi_1 f\rangle \\
&\quad\vdots \qquad\quad \vdots \qquad\quad \cdots \qquad\quad \vdots \qquad\quad \vdots \\
&A_0\langle w\phi_M\phi_0\rangle + A_1\langle w\phi_M\phi_1\rangle + \ldots + A_M\langle w\phi_M\phi_M\rangle = \langle w\phi_M f\rangle
\end{aligned}$$

These equations are more easily expressed in matrix form as

$$\begin{bmatrix} \langle w\phi_0\phi_0\rangle & \langle w\phi_0\phi_1\rangle & \cdots & \langle w\phi_0\phi_M\rangle \\ \langle w\phi_1\phi_0\rangle & \langle w\phi_1\phi_1\rangle & \cdots & \langle w\phi_1\phi_M\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle w\phi_M\phi_0\rangle & \langle w\phi_M\phi_1\rangle & \cdots & \langle w\phi_M\phi_M\rangle \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_M \end{bmatrix} = \begin{bmatrix} \langle w\phi_0 f\rangle \\ \langle w\phi_1 f\rangle \\ \vdots \\ \langle w\phi_M f\rangle \end{bmatrix} \tag{2.45}$$

Given the weighting function $w(\chi)$, the coordinate function $\phi_k(\chi)$ and a function to be approximated $f(\chi)$, the coefficient matrix and right hand side vector can be computed. This matrix equation can then be solved in theory to yield the set of coefficients A_k which provide the series approximation to $f(\chi)$ that minimizes the error in the sense of weighted least squares.

In the case at hand, selecting the coordinate functions to be orthogonal polynomials greatly simplifies Equation (2.45) and overcomes the practical problem of possible ill-conditioning of the coefficient matrix. If the functions are orthogonal, the orthogonality condition of Equation (2.15) will reduce the coefficient matrix to a diagonal matrix, since

$$\langle w(\chi)\ \phi_r(\chi)\ \phi_k(\chi)\rangle = 0 \quad \text{for } r \neq k$$

This effectively decouples the equations and gives the following results for the series coefficients

$$A_r = \frac{\langle w(\chi)\ \phi_r(\chi)\ f(\chi)\rangle}{\langle w(\chi)\ \phi_r^{\,2}(\chi)\rangle} \quad , r = 0,1,\ldots,M \tag{2.46}$$

Assuming the ϕ functions are the Chebyshev polynomials defined by Equation (2.23), Equation (2.46) becomes

$$A_r = \frac{\langle w(\chi)\ T_r(\chi)\ f(\chi)\rangle}{\langle w(\chi)\ T_r^{\,2}(\chi)\rangle} \quad , r = 0,1,\ldots,M \tag{2.47}$$

When $f(\chi)$ is explicitly defined over the continuous range $-1 \leq \chi \leq 1$, then the inner product definition of Equation (2.14a) yields

$$A_r = \frac{\int_{-1}^{1} \frac{f(\chi)T_r(\chi)}{\sqrt{1-\chi^2}}\, d\chi}{\int_{-1}^{1} \frac{T_r^{\,2}(\chi)}{\sqrt{1-\chi^2}}\, d\chi} \quad , r = 0,1,\ldots,M \tag{2.48}$$

The orthogonality condition of Equation (2.24b) indicates that the denominator may take on two values, depending on the value of r

$$\int_{-1}^{1} \frac{T_r^2(\chi)}{\sqrt{1-\chi^2}}\, d\chi = \begin{cases} \pi & \text{for } r = 0 \\ \frac{\pi}{2} & \text{for } r = 1,2,\ldots,M \end{cases}$$

Hence, two formulae result for the coefficients A_r

$$A_r = \frac{1}{\pi} \int_{-1}^{1} (1-\chi^2)^{-1/2}\, T_r(\chi)\, f(\chi)\, d\chi \qquad r = 0$$

$$A_r = \frac{2}{\pi} \int_{-1}^{1} (1-\chi^2)^{-1/2}\, T_r(\chi)\, f(\chi)\, d\chi \qquad r = 1,2,\ldots,M$$

If A_0 is redefined to be twice the given value above, then a single formula will give all the series coefficients. In this case, however, this multiplicative factor must be removed in the series representation by dividing the A_0 term by two. Thus, the finite Chebyshev series least squares representation for a continuous function $f(\chi)$ over $-1 \leq \chi \leq 1$ is given by

$$P_M(\chi) = \frac{A_o}{2} + \sum_{k=1}^{M} A_k T_k(\chi) = \sum_{k=0}^{M}{}' A_k T_k(\chi) \qquad (2.49a)$$

(' indicates first term factored by 2)

where the series coefficients are computed by

$$A_k = \frac{2}{\pi} \int_{-1}^{1} (1 - \chi^2)^{-1/2} T_k(\chi) f(\chi) d\chi \qquad (2.49b)$$

In an analogous manner the series representation and coefficient formulae for the first discrete case can be shown as

$$P_M(\chi) = \sum_{k=0}^{M}{}' A_k T_k(\chi) \qquad (2.50a)$$

where the coefficients are computed by

$$A_k = \frac{2}{N} \sum_{n=0}^{N}{}'' f(\chi_n) T_k(\chi_n) \qquad (2.50b)$$

(" indicates first and last terms factored by 2)

with the N+1 χ_n's given by

$$\chi_n = \cos \frac{n\pi}{N} \quad , n = 0,1,\ldots,N \qquad (2.50c)$$

For the second choice of discrete orthogonality points, the series is given by (2.50a), but the coefficients are given by

$$A_k = \frac{2}{N + 1} \sum_{n=0}^{N} f(\chi_n)\ T_k(\chi_n) \qquad (2.50d)$$

with the $N + 1$ χ_n's given by

$$\chi_n = \cos\left[\frac{2n + 1}{2(N + 1)}\ \pi\right] \qquad (2.50e)$$

which are the zeroes of $T_{N+1}(\chi)$.

The above relations are for least squares approximating polynomials. If it is desired to exactly fit the function $f(\chi)$ at the orthogonality points, an interpolating polynomial is required. In this case the order of the interpolating polynomial (M) will be the same as the value of N, which is the number of orthogonality points minus one. For the first set of discrete points, the interpolating polynomial is given by

$$P_N(\chi) = \sum_{k=0}^{N}{}'' A_k T_k(\chi) \qquad (2.50f)$$

with the coefficients given by Equation (2.50b) and the orthogonality points defined by Equation (2.50c). For the

second set of discrete points, the interpolating polynomial is given by Equation (2.50a) (but with M = N) and with the series coefficients and orthogonality points are given by Equations (2.50d) and (2.50e).

In the current implementation of the Picard-Chebyshev integrator, the discrete least squares approximating series was used instead of the interpolating series. There were several reasons for this selection. First, because the Picard-Chebyshev method is iterative, the computation of the right-hand sides at the orthogonality points in any one iteration will in general not represent the true right-hand sides. Therefore, fitting these points exactly in the sense of an interpolating polynomial will not provide improved convergence properties. Second, estimating the error in the case of interpolating series requires a knowledge of the maximum value attained on the interval of representation by the $(N + 1)^{st}$ derivative of the function being approximated. In practice this bound is often difficult to determine, especially for high order interpolating polynomials. In the case of a least squares approximating polynomial, however, the error can be bounded after the series representation has been computed. The error is given by

$$e_M(\chi) = f(\chi) - P_M(\chi)$$

$$= f(\chi) - \sum_{k=0}^{M}{}' A_k T_k(\chi) \qquad (2.51)$$

This error satisfies the discrete least squares criterion of minimum aggregate error

$$E = \sum_{n=0}^{N}{}'' e_M^2(\chi_n) = \text{minimum}$$

$$\therefore \quad E_{min} = \sum_{n=0}^{N}{}'' \{f^2(\chi_n) - \sum_{k=0}^{M} A_k^2 T_k^2(\chi_n)\} \qquad (2.52a)$$

for the orthogonality points of Equation (2.50c) and analogously

$$E_{min} = \sum_{n=0}^{N} \{f^2(\chi_n) - \sum_{k=0}^{M} A_k^2 T_k^2(\chi_n)\} \qquad (2.52b)$$

for the orthogonality points of Equation (2.50e). Third, and most important, the use of an interpolating polynomial forces the order of the polynomial to be one less than the number of orthogonality points (i.e., M = N). This removes flexibility from the algorithm in that the order of the polynomial and the number of orthogonality points are not independent but are intimately bound together. Fitting the right hand sides of the differential equations by approxi-

mating polynomials allows the order of the polynomial to be less than the number of orthogonality points and, therefore, reduces the amount of computation required. Additionally, having two independent input parameters in the integration algorithm allows flexibility in tailoring the integrator to balance accuracy and speed.

Based on the preceding discussions and the choice of approximating series just described, the right hand sides of the initial value problem specified by Equations (2.1) or (2.12) (on a component basis) will be expressed as least squares approximating polynomial series. These become the integrands in the Picard iterative solution of Equations (2.3) and (2.13), such that

$$F[t,y^{[k]}(t)] \sim \frac{A_o}{2} + \sum_{m=1}^{M} A_m T_m(\chi) \qquad (2.53)$$

The range $t_o \leq t \leq t_f$ is mapped into the Chebyshev variable χ by Equation (2.17)

$$\chi = 2\left(\frac{t - t_o}{t_f - t_o}\right) - 1 \qquad (2.54a)$$

which can be inverted to give t in terms of χ as

$$t = \frac{1}{2}\left[\chi(t_f - t_o) + t_f + t_o\right] \qquad (2.54b)$$

This relation will be useful in transforming the easily determined N + 1 orthogonality points χ_n of Equation (2.50c) into their corresponding values t_n in the true independent variable t. Then, for this choice of orthogonality points, the coefficients of the series are found by Equation (2.50b)

$$A_m = \frac{2}{N} \sum_{n=0}^{N}{}'' F[t_n, y^{[k]}(t_n)]\, T_m(\chi_n) \qquad (2.55)$$

where $\chi_n = \frac{n\pi}{N}$, $n = 0,1,\ldots,N$

Thus, the integrands of the Picard iterative solution have been approximated by appropriate series of Chebyshev polynomials. It now remains to integrate these series to form new series representing the next iterate of the solution.

2.1.3 Direct Integration of the Chebyshev Series

It is now assumed that the integrand in the Picard iterative method has been approximated as described above, such that

$$\int_{t_o}^{t_f} F[\sigma, y^{[k]}(\sigma)]\, d\sigma \simeq \frac{t_f - t_o}{2} \int_{-1}^{1} \sum_{m=0}^{M}{}' A_m T_m(\chi)\, d\chi$$

The factor $\frac{t_f - t_o}{2}$ is a normalization factor arising from the linear change of variables of Equation (2.54a). If this factor is brought inside the summation and integration, it can be applied to each A_m. Defining $A_m^* = \frac{t_f - t_o}{2} A_m$ and assuming an approximating series for the integrated result, the above equation can be written as

$$\int_{t_o}^{t_f} F[\sigma, y^{[k]}(\sigma)]d\sigma \simeq \int_{-1}^{1} \sum_{m=0}^{M}{}' A_m^* T_m(\chi)d\chi$$

$$= \sum_{m=0}^{M+1}{}' B_m T_m(\chi) \qquad (2.56)$$

It is now desired to know the relation between the A_m^* and the B_m coefficients.

In expanded form, Equation (2.56) becomes

$$\int_{-1}^{1} [\tfrac{1}{2} A_0^* + A_1^* T_1(\chi) + \ldots + A_m^* T_M(\chi)]d\chi$$

$$= \frac{A_o^*}{2} \int_{-1}^{1} T_0(\chi)d\chi + A_1^* \int_{-1}^{1} T_1(\chi)d\chi + \ldots$$

$$+ A_M^* \int_{-1}^{1} T_M(\chi)d\chi$$

Equations (2.40) give the necessary formulae to accomplish the term-by-term integration of the above equation. The result is

$$\int_{-1}^{1} {\sum_{m=0}^{M}}' A_m^* T_m(\chi)\,d\chi = \frac{A_o^*}{2} T_1 + \frac{A_1^*}{4} T_2 + \frac{A_2^*}{2}\left[\frac{T_3}{3} - \frac{T_1}{1}\right] + \cdots$$

$$\cdots + \frac{A_\ell^*}{2}\left[\frac{T_{\ell+1}}{\ell+1} - \frac{T_{\ell-1}}{\ell-1}\right] + \cdots + \frac{A_M^*}{2}\left[\frac{T_{M+1}}{M+1} - \frac{T_{M-1}}{M-1}\right]$$

which gives

$$\int_{-1}^{1} {\sum_{m=0}^{M}}' A_m^* T_m(\chi)\,d\chi = \left[\frac{A_o^*}{2} - \frac{A_2^*}{2}\right] T_1 + \left[\frac{A_1^*}{4} - \frac{A_3^*}{4}\right] T_2 + \cdots$$

$$\cdots + \left[\frac{A_{M-2}^*}{2(M-1)} - \frac{A_M^*}{2(M-1)}\right] T_{M-1} + \frac{A_{M-1}^*}{2M} T_M + \frac{A_M^*}{2(M+1)} T_{M+1}$$

(2.57)

Thus, the general coefficient of T_ℓ in the integrated series is made up of two terms, one each from the integrals of $T_{\ell-1}$ and $T_{\ell+1}$ in the integrand series.

$$A_{\ell-1}^* \int T_{\ell-1}\,d\chi = \frac{A_{\ell-1}^*}{2}\left[\frac{T_\ell}{\ell} - \frac{T_{\ell-2}}{\ell-2}\right]$$

$$A_{\ell+1}^* \int T_{\ell+1}\, d\chi = \frac{A_{\ell+1}^*}{2}\left[\frac{T_{\ell+2}}{\ell+2} - \frac{T_\ell}{\ell}\right]$$

Summing these two equations yields

$$A_{\ell-1}^* \int T_{\ell-1}\, d\chi + A_{\ell+1}^* \int T_{\ell+1}\, d\chi = \frac{A_{\ell+1}^*}{2(\ell+2)} T_{\ell+2} +$$

$$+ \frac{1}{2\ell}\,[A_{\ell-1}^* - A_{\ell+1}^*]T_\ell - \frac{A_{\ell-1}^*}{2(\ell-2)} T_{\ell-2}$$

Since the Chebyshev polynomials are linearly independent, it is possible to equate coefficients of $T_\ell(\chi)$ between the integrated series of Equation (2.57) and the assumed series representation, shown as the last of Equation (2.56). Hence

$$T_1(\chi) \;:\; B_1 = \frac{1}{2}\,[A_0^* - A_2^*]$$

$$T_2(\chi) \;:\; B_2 = \frac{1}{4}\,[A_1^* - A_3^*]$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$T_\ell(\chi) \;:\; B_\ell = \frac{1}{2\ell}\,[A_{\ell-1}^* - A_{\ell+1}^*]$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$T_M(\chi) \;:\; B_M = \frac{1}{2M} A_{M-1}^*$$

$$T_{M+1}(\chi) \;:\; B_{M+1} = \frac{1}{2(M+1)} A_M^*$$

This can be simply expressed as

$$B_\ell = \frac{1}{2\ell} [A_{\ell-1}^* - A_{\ell+1}^*] \quad , \quad \ell = 1,2,\ldots,M+1 \tag{2.58}$$

with $A_{M+1}^* = A_{M+2}^* = 0$

Thus, with the exception of B_o, the coefficients of the integrated series can be formed by simple arithmetic operations from the known coefficients of the Chebyshev polynomial series approximating the integrand.

It remains now to determine the free parameter B_o. In this case B_o is selected such that the initial condition is satisfied. That is

$$B_o = 2 \left[y_o - \sum_{m=1}^{M+1} B_m T_m(-1) \right] \tag{2.59}$$

The integrated series is now totally determined for any single iteration of the Picard method, since

$$y^{[k+1]}(t) \simeq \sum_{m=0}^{M+1}{}' B_m T_m(\chi)$$

where B_o is given by Equation (2.59) and the remaining coefficients B_m are given by Equation (2.58).

2.2 Application to the Orbit Propagation Problem

The basic Picard-Chebyshev algorithm derived in the previous section has been applied to three different formulations of the orbit propagation problem for artificial satellites. First, the method was applied to the Cowell equations of motion. This implementation has been previously documented in Reference 75. Second, the method was applied to the high precision Variation of Parameters (VOP) equations of motion. While previous efforts have used the Picard-Chebyshev method to integrate these types of formulations for planetary perturbations, no previous application of this method to artificial satellite theory has been implemented. Third, the method was applied to the averaged Variation of Parameters equations of motion formulated in equinoctial elements with the mean mean longitude as the rapidly varying phase element. Again, this implementation is new and has a distinct advantage. The averaged equations are smooth, since the high frequency (or short periodic) components of the element histories have been removed from the equations of motion by analytical or numerical averaging. This smoothness makes approximating the averaged element rates by Chebyshev series extremely simple, and low order approximating polynomials will give good accuracy results, even using relatively few orthogonality

points to compute series coefficients. The smoothness also allows the Chebyshev fits to be taken over very long integration arcs and still have rapid convergence of the Picard method. The disadvantage of this formulation is, of course, that the short periodic contributions must be added on to the averaged equations any time the true satellite position and velocity are desired. Much work has been done in this area, most recently the excellent work of Green (82).

The following sections will briefly describe the application of the Picard-Chebyshev method to each formulation of the orbit propagation problem. Implementation of the algorithm was accomplished in the Research and Development version of the Goddard Trajectory Determination System (GTDS) software package. Additionally, a stand-alone version of the algorithm was coupled with the parallel orbit estimation algorithm. This implementation will be described in the following chapter.

2.2.1 Cowell Formulation

The equations of motion for an artificial satellite in inertial Cartesian coordinates may be written as (28)

$$\ddot{\vec{r}} + \frac{\mu}{r^3}\vec{r} = \vec{a}_D \qquad (2.60)$$

with $\vec{r} \triangleq$ inertial position vector

$\mu \triangleq$ gravitational constant of the central body

$\vec{a}_D \triangleq$ disturbing acceleration due to perturbation sources

Note: $\vec{a}_D = 0$ for ideal Keplerian motion

This second order, nonlinear, vector ordinary differential equation can be reduced to two first order vector equations as in Reference 46. Thus

$$\frac{d(\dot{\vec{r}})}{dt} = -\frac{\mu}{r^3}\vec{r} + \vec{a}_D(\vec{r},\dot{\vec{r}},t)$$

$$= \vec{F}[t,\vec{r}(t),\dot{\vec{r}}(t)] \qquad (2.61a)$$

$$\frac{d(\vec{r})}{dt} = \dot{\vec{r}} \qquad (2.61b)$$

It is desired to integrate these equations over the interval $t_0 \leq t \leq t_f$, satisfying the initial conditions

$$\dot{\vec{r}}(t_o) = \dot{\vec{r}}_o \qquad (2.62a)$$

$$\vec{r}(t_o) = \vec{r}_o \qquad (2.62b)$$

In this implementation, the vector-valued function $\vec{F}$ represents the output of the mathematical force models, and may include conservative forces such as central body point mass gravitational attraction and central body asphericity and nonconservative forces such as atmospheric drag. In component form Equations (2.61a) and (2.61b) result in six coupled differential equations to be simultaneously integrated.

Following Equation (2.13), the vector form of the Picard iterative technique for the initial value problem posed by Equations (2.61) and (2.62) is given by

$$\dot{\vec{r}}^{[k+1]}(t) = \dot{\vec{r}}_o + \int_{t_o}^{t} \vec{F}[\sigma, \vec{r}^{[k]}(\sigma), \dot{\vec{r}}^{[k]}(\sigma)]d\sigma \qquad (2.63a)$$

$$\vec{r}^{[k+1]}(t) = \vec{r}_o + \int_{t_o}^{t} \dot{\vec{r}}^{[k+1]}(\sigma)d\sigma \qquad (2.63b)$$

In this implementation, the zeroth iterates $\dot{\vec{r}}^{[0]}(t)$ and $\vec{r}^{[0]}(t)$ are taken to be the solutions of the unperturbed Kepler problem. They are continuous vector functions on the interval $t_0 \leq t \leq t_f$ and they satisfy the initial conditions of Equations (2.62a) and (2.62b). The Kepler solution is found by using the Lagrange F, G, F_t and G_t coefficients given in Reference 29 (see Appendix A).

The integrands of Equation (2.63a) must now be approximated by Chebyshev series. Equations (2.54a) and (2.54b) map the time t in the interval $t_0 \leq t \leq t_f$ into the Chebyshev variable χ in the interval $-1 \leq \chi \leq +1$, and the converse. The least squares approximating series for the i^{th} component of the spacecraft total inertial acceleration in Cartesian coordinates is given by Equation (2.53), such that

$$F_i[\sigma, \vec{r}^{[k]}(\sigma), \dot{\vec{r}}^{[k]}(\sigma)] = {\sum_{m=0}^{M}}' A_{i,m} T_m(\chi) \quad (2.64)$$

$$\text{for } i = 1,2,3$$

The coefficients for these series representations are found from Equation (2.55) as

$$A_{i,m} = \frac{2}{N} {\sum_{n=0}^{N}}'' F_i[t_n, \vec{r}^{[k]}(t_n), \dot{\vec{r}}^{[k]}(t_n)] T_m(\chi_n) \quad (2.65)$$

$$\text{for } m = 0,1,\ldots,M$$
$$\text{and } i = 1,2,3$$

where the orthogonality points χ_n are given by Equation (2.50c).

Finally, these series representations must be integrated twice to give the series representations for the inertial Cartesian velocity and position components. The integrals on the right hand side of Equation (2.13) become, in component form

$$\int_{t_o}^{t_f} F[\sigma, \vec{r}^{[k]}(\sigma), \dot{\vec{r}}^{[k]}(\sigma)]d\sigma \simeq \frac{t_f - t_o}{2} \int_{-1}^{1} {\sum_{m=0}^{M}}' A_{i,m} T_m(\chi) d\chi$$

$$\simeq \int_{-1}^{1} {\sum_{m=0}^{M}}' A_{i,m}^* T_m(\chi) d\chi$$

$$\simeq {\sum_{m=0}^{M+1}}' B_{i,m} T_m(\chi) \qquad (2.66)$$

In Equation (2.66), $A_{i,m}^*$ is defined by the relation

$$A_{i,m}^* = \frac{t_f - t_o}{2} A_{i,m}$$

with the $A_{i,m}$ values given by Equation (2.65). Following the development of Section 2.1.3, the coefficients for the series representing the i^{th} component of velocity are given by Equation (2.58), such that

$$B_{i,m} = \frac{1}{2m} [A^*_{i,m-1} - A^*_{i,m+1}] \qquad (2.67)$$

for $m = 1,2,\ldots,M+1$

and $i = 1,2,3$

with $A^{*}_{i,M+1} = A^{*}_{i,M+2} = 0$

The free parameters $B_{i,0}$ $(i = 1,2,3)$ are selected such that the velocity initial conditions are satisfied, as shown by Equation (2.59). Thus

$$\dot{r}_i^{[k+1]}(t) = \sum_{m=0}^{M+1}{}' B_{i,m} T_m(\chi) \qquad (2.68)$$

for $i = 1,2,3$

with the $B_{i,m}$ values given by Equation (2.67). These approximating series now become the representations of the integrands in Equation (2.63b). These must be integrated again to yield the series representations for the position components. By similar computation as above, the i^{th} component of position is found from

$$r_i^{[k+1]}(t) = \sum_{m=0}^{M+2}{}' C_{i,m} T_m(\chi) \qquad (2.69)$$

for $i = 1,2,3$

The $C_{i,m}$ values are given by

$$C_{i,m} = \frac{1}{2m} [B_{i,m-1} - B_{i,m+1}] \qquad (2.70)$$

for $m = 1,2,\ldots,M+2$

and $i = 1,2,3$

with $B_{i,M+2} = B_{i,M+3} = 0$

Again, the free parameters $C_{i,0}$ ($i = 1,2,3$) are selected such that the position initial conditions are satisfied.

In the above manner the $k+1^{st}$ iterate of the position and velocity components are found by Picard's method. In the current implementation, the algorithm is considered to have converged when the relative error in each component at every orthogonality point in the arc does not exceed some convergence criterion for two successive iterations. If this convergence criterion is not met, then the iterative method continues. As previously stated, the results of integrating these Cowell equations are Chebyshev series representing the Cartesian components of the inertial position and velocity vectors over the approximating interval.

2.2.2 High Precision Variation of Parameters Formulation

The Picard-Chebyshev integrator has also been applied to the Gaussian form of the Variation of Parameters (VOP)

equations of motion. These equations have the form given in Reference 46 as

$$\frac{da_i}{dt} = \frac{\partial a_i}{\partial \vec{r}} \cdot \vec{a}_D \qquad (2.71a)$$

$$\text{for } i = 1,2,\ldots,5$$

where $a_1, a_2, \ldots, a_5$ represent the five slowly varying orbital elements and $\vec{a}_D$ is the perturbing (or non-Keplerian) accelerations acting on the spacecraft. The equation for the sixth and rapidly varying phase element is given by

$$\frac{da_6}{dt} = a'_6 + \frac{\partial a_6}{\partial \vec{r}} \cdot \vec{a}_D \qquad (2.71b)$$

where a'_6 represents the time derivative of element a_6 for the case of unperturbed Keplerian motion. Derivations of these equations may be found in References 82 and 105.

At any instant the relations between the inertial Cartesian components of the position and velocity vectors $\vec{r}$ and $\dot{\vec{r}}$ and the vector of orbital elements $\vec{a}$ in the perturbed problem are two-body relations. Thus, it is possible to write a functional relationship connecting position and velocity to orbital elements at any instant as

$$\vec{r}(t) = \vec{r}(t,\vec{a})$$

$$\dot{\vec{r}}(t) = \dot{\vec{r}}(t,\vec{a}) \qquad (2.72)$$

Conversely

$$\vec{a} = \vec{a}[t,\vec{r}(t),\dot{\vec{r}}(t)] \qquad (2.73)$$

relates the orbital elements to the position and velocity. If the solutions of Equations (2.71a) and (2.71b) can be found, then Equation (2.72) can be used to compute the inertial Cartesian components of the spacecraft position and velocity at any desired time.

The VOP equations in the current implementation have been formulated in three different element sets: Keplerian (classical) elements, equinoctial elements with mean longitude (λ) as the rapidly varying phase element, and equinoctial elements with eccentric longitude (F) as the rapidly varying phase element. Appendix A defines these different element sets. In vector form the equations to be integrated are

$$\frac{d\vec{a}}{dt} = a'_6 \hat{e}_6 + \frac{\partial \vec{a}}{\partial \dot{\vec{r}}} \cdot \vec{a}_D \qquad (2.74)$$

where $\vec{a}$ = vector of orbital elements

and $\hat{e}_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

with a'_6 = the unperturbed time derivative of the fast orbital element

For the three sets of elements, the values of a'_6 are given below

Keplerian:

$$a'_6 = n \tag{2.75a}$$

Equinoctial (λ):

$$a'_6 = n \tag{2.75b}$$

Equinoctial (F):

$$a'_6 = \frac{na}{r} \tag{2.75c}$$

where n = mean motion = $\sqrt{\mu/a^3}$

μ = central body gravitational parameter

a = semimajor axis

r = magnitude of the position vector = $\sqrt{\vec{r} \cdot \vec{r}}$

Equation (2.72) shows that the position and velocity vectors are functions of the orbital elements at a given time t. From Equation (2.61a) it is seen that the perturbing acceleration $\vec{a}_D$ at a given time t is a function of the position and velocity and therefore of the corresponding orbital elements at that time. The 6x3 matrix of the partial derivatives of the orbital elements with respect to the components of velocity at time t are derived in Section 3.3.2.3. It is seen that, at a given time t, the elements of this matrix can also be thought of as functions of the orbital elements at that time. Hence, the entire right hand side of the vector differential equation of motion given by Equation (2.74) is a function of the orbital elements. The equation can therefore be written as

$$\frac{d\vec{a}}{dt} = \vec{F}[t,\vec{a}(t)] \qquad (2.76)$$

The above equation in component form represents six coupled, first order, nonlinear, ordinary differential equations which can be solved by the method of Picard iteration. To complete the specification of the initial value problem, however, it remains to establish the appropriate initial conditions for Equation (2.76). In the Cowell formulation the components of the epoch position

and velocity vectors represented the initial conditions for that integration problem. Analogously, the epoch orbital elements, either given or computed from the epoch position and velocity components, serve as the initial conditions for this integration. Hence

$$\vec{a}(t_o) = \vec{a}[t_o,\vec{r}(t_o),\dot{\vec{r}}(t_o)] = \vec{a}_o \qquad (2.77)$$

From Equation (2.13), the vector form of the Picard iterative technique for the initial value problem posed by Equation (2.76) and (2.77) is given by

$$\vec{a}^{[k+1]}(t) = \vec{a}_o + \int_{t_o}^{t_f} \vec{F}[\sigma,\vec{a}^{[k]}(\sigma)]d\sigma \qquad (2.78)$$

In the Cowell implementation, the zeroth iterate for the position and velocity components was taken to be the solution of the unperturbed Kepler problem. By analogy, the zeroth iterate for Equation (2.78) is

$$\begin{aligned} a_i^{[0]}(t) &= a_i(t_o) \ , \ i = 1,2,\ldots,5 \\ a_6^{[0]}(t) &= a'_6(t - t_o) + a_6(t_o) \end{aligned} \qquad (2.79)$$

where a'_6 is given by Equations (2.75a), (2.75b) or (2.75c) depending on the element set being integrated.

The remainder of the development for the VOP formulation follows exactly that previously discussed for the Cowell development, with the exception that only one integration is required for these first order equations. The right hand sides of Equation (2.78) (in component form) are approximated by the Chebyshev series of Equation (2.53), where the coefficients are given by Equation (2.55) and the orthogonality points by Equation (2.50c). Hence

$$\int_{t_o}^{t_f} F_i[\sigma,\vec{a}^{[k]}(\sigma)]d\sigma \simeq \int_{-1}^{1} {\sum_{m=0}^{M}}' A_{i,m}T_m(\chi)d\chi \tag{2.80a}$$

$$\text{for} \quad i = 1,2,\ldots,6$$

$$\text{with} \qquad A^*_{i,m} = \frac{t_f - t_o}{N} {\sum_{n=0}^{N}}'' F_i[t_n,\vec{a}^{[k]}(t_n)]\bar{T}_m(\chi_n) \tag{2.80b}$$

$$\text{and} \qquad \chi_n = \frac{n\pi}{N} \quad , \quad n = 0,1,\ldots,N \tag{2.80c}$$

Again, t_n and χ_n are related by the reciprocal relations of Equations (2.54a) and (2.54b).

The series representing the elements are given by

$$a_i^{[k+1]}(t) = \sum_{m=0}^{M+1}{}' B_{i,m} T_m(\chi) \qquad (2.81a)$$

$$\text{for } i = 1,2,\ldots,6$$

$$\text{with} \qquad B_{i,m} = \frac{1}{2m}\left[A^*_{i,m-1} - A^*_{i,m+1}\right] \qquad (2.81b)$$

$$\text{for } m = 1,2,\ldots,M+1$$

$$\text{assuming } A^*_{i,M+1} = A^*_{i,M+2} = 0$$

Once again the free parameters $B_{i,0}$ $(i = 1,2,\ldots,6)$ are adjusted such that the series of Equation (2.81a) satisfy the initial conditions $\vec{a}(t_o)$. The results of these integrations are Chebyshev approximating series representing the orbital element histories over the interval of representation. From these element histories, the inertial Cartesian position and velocity vectors can be computed at any given time by the transformation equations given functionally by Equation (2.72).

2.2.3 Averaged Variation of Parameters Formulation

The application of the Picard-Chebyshev method to the averaged VOP equations is exactly analogous to the previous

discussion for the high precision VOP formulation. McClain (110, 111) provides excellent references for the development of the averaged VOP equations of motion. In this development, the high precision VOP equations for a single disturbing function have the form

$$\frac{da_i}{dt} = \varepsilon G_i(\vec{a},\lambda)$$

$$\text{for } i = 1,2,\ldots,5$$

$$\frac{d\lambda}{dt} = n(a_1) + \varepsilon G_6(\vec{a},\lambda)$$

where

$\vec{a}$ = vector of five slowly varying orbital equinoctial elements

λ = rapidly varying phase element (mean longitude)

$n(a)$ = satellite mean motion

a_1 = a = semimajor axis

ε = small parameter proportional to the disturbing force

A near-identity transformation relating the osculating elements to the averaged elements is assumed to be of the form

$$a_i = \bar{a}_i + \sum_{j=1}^{N} \varepsilon^j \eta_{i,j}(\vec{\bar{a}}, \bar{\lambda}) + O(\varepsilon^{N+1})$$

$$\text{for } i = 1,2,\ldots,5$$

$$\lambda = \bar{\lambda} + \sum_{j=1}^{N} \varepsilon^j \eta_{6,j}(\vec{\bar{a}}, \bar{\lambda}) + O(\varepsilon^{N+1})$$

where the barred variables are averaged elements and the η's represent the short periodic functions. The N is the order of the small parameter ε to be included in the theory. McClain shows that the averaged VOP equations have the form

$$\frac{d\bar{a}_i}{dt} = \sum_{j=1}^{N} \varepsilon^j D_{i,j}(\vec{\bar{a}}) + O(\varepsilon^{N+1}) \qquad (2.82a)$$

$$\text{for } i = 1,2,\ldots,5$$

$$\frac{d\bar{\lambda}}{dt} = n(\bar{a}_1) + \sum_{j=1}^{N} \varepsilon^j D_{6,j}(\vec{\bar{a}}) + O(\varepsilon^{N+1}) \qquad (2.82b)$$

From these equations, it is seen that the right hand sides are functions of the five slowly varying averaged elements only. For a full development of the actual right hand sides for the different possible disturbing functions, see References 47, 48, 54, 82, 104, 111 and 112. Hence, the equations may be written as

$$\frac{d\vec{\bar{a}}^*}{dt} = \vec{F}(\vec{\bar{a}}^*)$$

where $$\vec{\bar{a}}^* = \begin{pmatrix} \vec{\bar{a}} \\ \hline \bar{\lambda} \end{pmatrix}$$

Equation (2.83) is analogous to Equation (2.76). The remaining development is the same as that for the high precision VOP equations and will not be repeated here.

The results of performing these integrations are Chebyshev series representing the averaged element histories over the interval of representation. Because of the smooth, slowly varying nature of these equations, it is expected that low order Chebyshev series will adequately represent the averaged element rates and that long integration steps can be taken. The numerical results of Section 2.4 bear out these conjectures. As previously noted, however, the short periodic contributions must be computed to determine the osculating elements and thence the osculating position and velocity vectors. This is considered beyond the scope of this work.

2.3 Algorithmic Architecture

The importance of the Picard-Chebyshev integration method in the previously described applications does not lie solely in its ability to provide accurate integrations of the equations of motion. The primary significance of this formulation rests on the degree of parallelism inherent in the structure of the algorithm. This section will outline the architecture of the Picard-Chebyshev orbit propagation algorithm, with emphasis on those computations which are uniquely suited to the target SIMD hardware architecture. The iterative flow of the algorithm will be described.

2.3.1 Software Description

The target computer architecture for this effort is the Single Instruction Steam/Multiple Data Stream (SIMD) parallel processor discussed in Chapter 1. This type of architecture is most effective only when applied to algorithms structured such that major blocks of instructions can be executed simultaneously on different streams of data. In these cases, performance improves almost linearly with the number of parallel processing elements used (78). The following description of the algorithm architecture for the Picard-Chebyshev orbit propagation algorithm will display

this structure to a high degree. Section 2.4 will discuss the numerical accuracy of the method and Chapter 4 will describe the potential performance improvements derived from this algorithm architecture.

The orbit propagation problems to which the Picard-Chebyshev algorithm has been applied are described in Section 2.2. The major tasks to be accomplished by the algorithm for each formulation are delineated in that section. The following flow description uses Cowell equations of motion as an example for the discussion. Any modification peculiar to the application of the method for the VOP formulations will be noted.

A brief description of the software modules involved in this implementation is provided in Table 2.2. The major input parameters for the algorithm are the following:

M	$\triangleq$	order of the Chebyshev series representing the integrands
N	$\triangleq$	number of Chebyshev orthogonality points minus one involved in the computation of series coefficients
TL	$\triangleq$	lower limit of integration arc in the true independent variable t
TU	$\triangleq$	upper limit of integration arc in the true independent variable t

Table 2.2 Software Module Descriptions for the Picard-Chebyshev Orbit Propagation Algorithm (Page 1 of 4)

For the GTDS-RD version:

Name	Description
CSER1	Evaluate a function $y(\chi)$, represented as a series of Chebyshev polynomials $T(\chi)$ at a particular point χ'.
CSRSTA	Evaluate the integration state vector at some time t' by evaluating the appropriate Chebyshev series.
EPELMS	Compute the Keplerian or equinoctial elements at some epoch time t_o, given the inertial Cartesian components of position and velocity at that time.
EQUMTN†	Compute the time derivatives of the VOP orbital elements.
FORCES†	Compute the two-body and free-flight perturbative accelerations acting on a spacecraft at a given state.
F2BDY	Compute the two-body acceleration.
KEPLR	Given the mean anomaly and eccentricity for a Keplerian orbit, solve Kepler's equation.

† These are existing GTDS-RD subroutines.

Table 2.2. Software Module Descriptions for the Picard-Chebyshev Orbit Propagation Algorithm (Page 2 of 4)

Name	Description
LGRNGE	Compute the components of the inertial Cartesian position and velocity vectors at some given value of true anomaly for the unperturbed two body problem, given an epoch state vector and epoch true anomaly.
PCHEBY	Integrate by the method of Picard iteration the components of a vector differential equation whose right hand sides have been approximated by series of Chebyshev polynomials.
VCOINT	Compute the integral of a vector whose components are approximated by Chebyshev series.
VMATTX	Compute the matrix of Chebyshev polynomials $T_o(\chi)$ to $T_M(\chi)$ evaluated at the points of orthogonality χ_o through χ_N.
VORTPT	Compute the N + 1 points χ_o to χ_n at which the Chebyshev polynomials of degree M or less possess the property of orthogonality with respect to summation.

Table 2.2. Software Module Descriptions for the Picard-Chebyshev Orbit Propagation Algorithm (Page 3 of 4)

For the stand-alone†† version:

Name	Description
AERO	Compute the contribution to the total spacecraft acceleration vector due to atmospheric drag.
AKTOAE	Convert from Keplerian elements to equinoctial elements.
ELMRPT	Compute Keplerian or equinoctial elements at each integration output time and control the ephemeris printout.
ELRATE	Same as 'EQUMTN' above.
EPELM	Same as 'EPELMS' above.
EQELPV	Convert a set of equinoctial orbital elements to inertial Cartesian position and velocity.
FORCE	Same as 'FORCES' above.

†† Note: Also included in the stand-alone software are subroutines 'CSER1', 'CSRSTA', 'F2BDY', 'KEPLR', 'LGRNGE', 'VCOINT', 'VCOVAL', 'VMATTX' and 'VORTPT' described above.

Table 2.2. Software Module Descriptions for the Picard-Chebyshev Orbit Propagation Algorithm (Page 4 of 4)

Name	Description
LEGPRM	Compute the derivatives of the Legendre polynomials by using a recursion relation.
PAKTAE	Compute the partial derivatives of a set of Keplerian orbit elements with respect to their corresponding equinoctial elements.
PEQTX	Compute the partial derivatives of the vector of equinoctial orbital elements with respect to the position and velocity vectors.
VTEST	Same as 'PCHEBY' above.

C $\triangleq$ convergence criterion for the method

$\vec{r}_0, \dot{\vec{r}}_0$ $\triangleq$ epoch initial conditions for the integration span. It should be noted that for the high precision VOP equations, the initial condition vector would be $\vec{a}_0$ and for the averaged VOP equations it would be $\bar{\vec{a}}_0$.

The program begins by error-checking the input parameters. Then a one-time initialization step is performed by calling subroutines VORTPT and VMATTX. Subroutine VORTPT computes the orthogonality points

$$\chi_n \in [-1,+1] \text{ for } n = 0,1,2,\ldots,N$$

and

$$t_n \in [TL,TU] \text{ for } n = 0,1,2,\ldots,N$$

where, from Equation (2.50c)

$$\chi_n = -\cos \frac{n\pi}{N}$$

and, from Equation (2.54b)

$$t_n = \frac{1}{2} [\chi_n (TU - TL) + TU + TL]$$

These are the points along the arc to be integrated at which the right hand sides of the equations of motion must be evaluated and subsequently approximated by Chebyshev series. Note than once N is specified, the locations of the points χ_n (and hence t_n) are independent of each other and their locations can be computed in parallel.

Subroutine now uses the points χ_n to compute the matrix [TX] as described in Figure 2.2. Here is another candidate area for parallel processing. Equations (2.25a) and (2.25b) show that $T_0(\chi) = 1$ and $T_1(\chi) = \chi$. Higher order Chebyshev polynomials may be calculated via the three term recursion relation of Equation (2.28)

$$T_{m+1}(\chi) = 2\chi T_m(\chi) - T_{m-1}(\chi), \; m \geq 2$$

with starting values given by

$$T_0(\chi) = 1$$

$$T_1(\chi) = \chi$$

$$[TX] = \begin{bmatrix} T_0(\chi_0) & T_0(\chi_1) & \cdots & T_0(\chi_N) \\ T_1(\chi_0) & T_1(\chi_1) & \cdots & T_1(\chi_N) \\ \vdots & \vdots & \cdots & \vdots \\ T_M(\chi_0) & T_M(\chi_1) & \cdots & T_M(\chi_N) \end{bmatrix}$$

Figure 2.2. Structure of the [TX] Array

Using these facts, each column of the [TX] array can be computed independently and, if the candidate parallel processor had N + 1 elements, simultaneously. The [TX] array can now be used in the iterative process to determine the coefficients of the series approximations to the acceleration components for the Cowell Equations [Equation (2.65)] or to the element rate components for the VOP formulations [Equation (2.80b)].

The program now computes the zeroth iterate of the solution, which is required to initiate the Picard method. The program calls subroutine EPELMS which computes the orbital elements (either Keplerian or equinoctial) associated with the lower limit of the current integration arc TL.

As previously discussed, the zeroth iterate for the Cowell formulation is taken to be the position and velocity of the unperturbed Kepler problem. Therefore, at each time orthogonality point t_n, the mean anomaly is computed. The routine KEPLR is then called to solve Kepler's equation for the eccentric and true anomalies at each time t_n. Subroutines LGRNGE is called to compute the position and velocity vectors at t_n using the Lagrange F, G, F_t and G_t coefficients (see Appendix A). Finally, subroutine F2BDY is called to compute the two body acceleration at t_n. For the VOP formulations, Equation (2.79) indicates the form of the zeroth iteration. In this case, the first five components of the vector of orbital elements are assumed to remain constant and equal to their values at the start time (TL) of the integration arc. Thus

$$a_i^{[0]}(t_n) = a_i(TL)$$

$$\text{for } i = 1,2,\ldots,5$$

$$\text{and } n = 0,1,\ldots,N$$

The rapidly varying phase element, however, must be computed at each orthogonality point t_n by the relation

$$a_6^{[0]}(t_n) = a'_6(t_n - TL) + a_6(TL)$$

where a'_6 is found from Equations (2.75a), (2.75b) or (2.75c), depending on the element set used in the VOP formulation. Again, the computations at any orthogonality point t_n are independent of those at any other and hence can be completed in parallel.

As the iterative procedure progresses, the k^{th} iterate of the integration state is stored in array [STK] as shown in Figure 2.3. The [STK] array represents the integration state vector at the N + 1 orthogonality points for the k^{th} iteration of the Picard method. These components are saved and later compared to the state vector components at the $k+1^{st}$ iteration to test for convergence of the method.

Using the mathematical models of the right hand sides of the differential equations, expressed functionally by Equations (2.61a), (2.76) or (2.83), the rows of the [STK] array can be used to compute the forcing functions at the N + 1 orthogonality points t_n. The components of these force evaluations are stored in array [FORC] as indicated in Figure 2.4. Each row of the [FORC] array can be computed in parallel with any other row, using as input to the force models the corresponding row in array [STK].

COWELL FORMULATION

$$[STK] \triangleq \begin{bmatrix} r_1(t_0) & r_2(t_0) & r_3(t_0) & \dot{r}_1(t_0) & \dot{r}_2(t_0) & \dot{r}_3(t_0) \\ r_1(t_1) & r_2(t_1) & r_3(t_1) & \dot{r}_1(t_1) & \dot{r}_2(t_1) & \dot{r}_3(t_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ r_1(t_N) & r_2(t_N) & r_3(t_N) & \dot{r}_1(t_N) & \dot{r}_2(t_N) & \dot{r}_3(t_N) \end{bmatrix}$$

VOP FORMULATION

$$[STK] \triangleq \begin{bmatrix} a_1(t_0) & a_2(t_0) & a_3(t_0) & a_4(t_0) & a_5(t_0) & a_6(t_0) \\ a_1(t_1) & a_2(t_1) & a_3(t_1) & a_4(t_1) & a_5(t_1) & a_6(t_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1(t_N) & a_2(t_N) & a_3(t_N) & a_4(t_N) & a_5(t_N) & a_6(t_N) \end{bmatrix}$$

Figure 2.3. Structure of [STK] Array for Storage of k^{th} Iterate of Integration State

COWELL FORMULATION

$$\vec{F}[t_n, \vec{r}(t_n), \dot{\vec{r}}(t_n)], \; n = 0,1,\ldots,N$$

$$[\text{FORC}] \triangleq \begin{bmatrix} \overbrace{\ddot{r}_1(t_0) \quad \ddot{r}_2(t_0) \quad \ddot{r}_3(t_0)} & 0 & 0 & 0 \\ \ddot{r}_1(t_1) & \ddot{r}_2(t_1) & \ddot{r}_3(t_1) & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ddot{r}_1(t_N) & \ddot{r}_2(t_N) & \ddot{r}_3(t_N) & 0 & 0 & 0 \end{bmatrix}$$

VOP FORMULATION

$$\vec{F}[t_n, \vec{a}(t_n)], \quad n = 0,1,\ldots,N$$

$$[\text{FORC}] \triangleq \begin{bmatrix} \overbrace{\dot{a}_1(t_0) \quad \dot{a}_2(t_0) \quad \dot{a}_3(t_0) \quad \ldots \quad \dot{a}_6(t_0)} \\ \dot{a}_1(t_1) \quad \dot{a}_2(t_1) \quad \dot{a}_3(t_1) \quad \ldots \quad \dot{a}_6(t_1) \\ \vdots \quad \vdots \quad \vdots \quad \ldots \quad \vdots \\ \dot{a}_1(t_N) \quad \dot{a}_2(t_N) \quad \dot{a}_3(t_N) \quad \ldots \quad \dot{a}_6(t_N) \end{bmatrix}$$

Figure 2.4. Structure of [FORC] Array for Storage of k^{th} Iterate of Force Functions

The main iterative loop in the algorithm will now be described. Subroutine VCOVAL is called to compute the coefficients of the Chebyshev series approximations for the components of the integrand vector. This can be thought of as a specialized matrix product as shown below. The general form for the m^{th} coefficient in the Chebyshev series representing the i^{th} component of the total acceleration in the Cowell formulation is given by Equation (2.65). This may be written as

$$A^{*}_{i,m} = \frac{(TU - TL)}{N} \sum_{n=0}^{N}{}'' F_i[t_n, \vec{r}(t_n) \dot{\vec{r}}(t_n)] T_m(\chi_n) \tag{2.84}$$

for m = 0,1,...,M

and i = 1,2,3

where the " superscript on the summation means that the first and last terms are factored by 2.

In the VOP formulation the F_i in the above equation is replaced by the element rate mathematical model $F_i[t_n, \dot{\vec{a}}(t_n)]$, as shown in Equation (2.80b), and the index i = 1,2,...,6. Equation (2.84) above has the form of a vector inner produce in which the m^{th} row in the [TX] array is dotted with the i^{th} column of the [FORC] array, but with the first and

last terms factored by two. Hence, all M + 1 coefficients for each Chebyshev series representing each integrand component are found by

$$[A] = \frac{(TU - TL)}{N} [TX] * [FORC]'' \qquad (2.85)$$

where the double prime of the [FORC] array means that the first and last entry in each column are factored by two. The [A] array takes the form shown in Figure 2.5, where, for the Cowell problem, the last three columns are all zeros.

With the series coefficients for the integrand components stored in the columns of [A], the simple integration formulae of Equations (2.67) and (2.70) (Cowell formulation) or (2.81b) (VOP formulation) can be applied to these columns to determine the coefficients for the integrated series except for the zeroth order coefficients. These zeroth order coefficients are found by adjusting them to satisfy the appropriate initial conditions. The coefficients of the integrated series are computed by subroutine VCOINT and are stored in array [BC] as shown in Figure 2.6. It should be noted that each integration raises the order of the resulting series by one.

$$[A] \triangleq \begin{bmatrix} A^*_{1,0} & A^*_{2,0} & A^*_{3,0} & \cdots & A^*_{6,0} \\ A^*_{1,1} & A^*_{2,1} & A^*_{3,1} & \cdots & A^*_{6,1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ A^*_{1,M} & A^*_{2,M} & A^*_{3,M} & \cdots & A^*_{6,M} \end{bmatrix}$$

Figure 2.5. Structure of [A] Array Containing Integrand Series Coefficients

The integration program now must evaluate the resulting integrated series at the N + 1 Chebyshev orthogonality points. This is done by N + 1 calls to subroutine CSRSTA, the Chebyshev series evaluator. The results are stored in array [STK1] in the same form as array [$\dot{STK}$]. As the integration state vector components are evaluated, they are compared to the corresponding components found in the previous iteration (stored in [STK]). If the relative error is greater than some input convergence criterion, then logical flags are set to indicate that the method has not converged. The integration state vectors stored in [STK1] are used in the next iteration as input for the force model evaluations at the N + 1 Chebyshev orthogonality points. For all iterations other than zero, the force function

COWELL FORMULATION

$$[BC] \triangleq \begin{bmatrix} B_{1,0} & B_{2,0} & B_{3,0} & C_{1,0} & C_{2,0} & C_{3,0} \\ B_{1,1} & B_{2,1} & B_{3,1} & C_{1,1} & C_{2,1} & C_{3,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ B_{1,M} & B_{2,M} & B_{3,M} & C_{1,M} & C_{2,M} & C_{3,M} \\ B_{1,M+1} & B_{2,M+1} & B_{3,M+1} & C_{1,M+1} & C_{2,M+1} & C_{3,M+1} \\ \underbrace{0 \quad 0 \quad 0}_{\text{Velocity Series}} & & & \underbrace{C_{1,M+2} \quad C_{2,M+2} \quad C_{3,M+2}}_{\text{Position series}} & & \end{bmatrix}$$

VOP FORMULATION

$$[BC] \triangleq \begin{bmatrix} B_{1,0} & B_{2,0} & B_{3,0} & \cdots & B_{6,0} \\ B_{1,1} & B_{2,1} & B_{3,1} & \cdots & B_{6,1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ B_{1,M} & B_{2,M} & B_{3,M} & \cdots & B_{6,M} \\ B_{1,M+1} & B_{2,M+1} & B_{3,M+1} & \cdots & B_{6,M+1} \end{bmatrix}$$

Figure 2.6. Structure of [BC] Array for Storage of Integrated Series Coefficients

mathematical models will include all selected perturbation sources, whereas in the zeroth iteration only the two body force models were used.

The program now tests for convergence in the current iteration. It checks the logical flags associated with each integration state component. If they are all true then the $k+1^{st}$ iteration is considered to have converged. Because of the nonlinear nature of the equations of motion, the program looks for three successive solutions that agree to within the convergence criterion. That is, the $k-1^{st}$ and k^{th} solutions must agree to within the convergence criterion at the N + 1 orthogonality points AND the k^{th} and $k+1^{st}$ solutions must do the same. The program checks the appropriate logical variables and if the above criteria are met, the program is terminated. If not, the results from the $k+1^{st}$ iteration are shifted to the storage locations for iteration k and a new iteration is begun by returning to subroutine VCOVAL and evaluating the new series coefficients using the forceing function evaluations from the previous iteration.

When convergence is achieved, the program checks to see if the total integration span has been completed. If so, execution is terminated. If not, the integration

state vector components at the upper limit of the current arc (t = TU or χ = +1) become the initial conditions for the next arc and the Picard-Chebyshev algorithm is restarted.

2.3.2 Algorithm Parallelism

The preceding discussion indicated that several portions of the algorithm contained computations amenable to parallel processing. Figure 2.7 presents a functional flow diagram of the Picard-Chebyshev orbit propagation algorithm. In this diagram, the areas of parallel execution are clearly indicated. The "width" (W) of a parallel processor is defined to be the number of identical processing elements under the control of the main control unit. From Figure 2.7, if the width of the parallel processor is greater than or equal to the number of orthogonality points (N + 1), then each column of the array [TX] can be computed in parallel during the one-time initialization step. Similarly, in the computation of the zeroth iteration, a parallel processor of width W = N + 1 affords the ability to compute the required N + 1 two body force evaluations in parallel. During the iterative loop, this parallel processor of width W = N + 1 allows parallel evaluation of the integration state Chebyshev series, the relative error and the integrand vector at the N + 1 orthogonality points for each iteration.

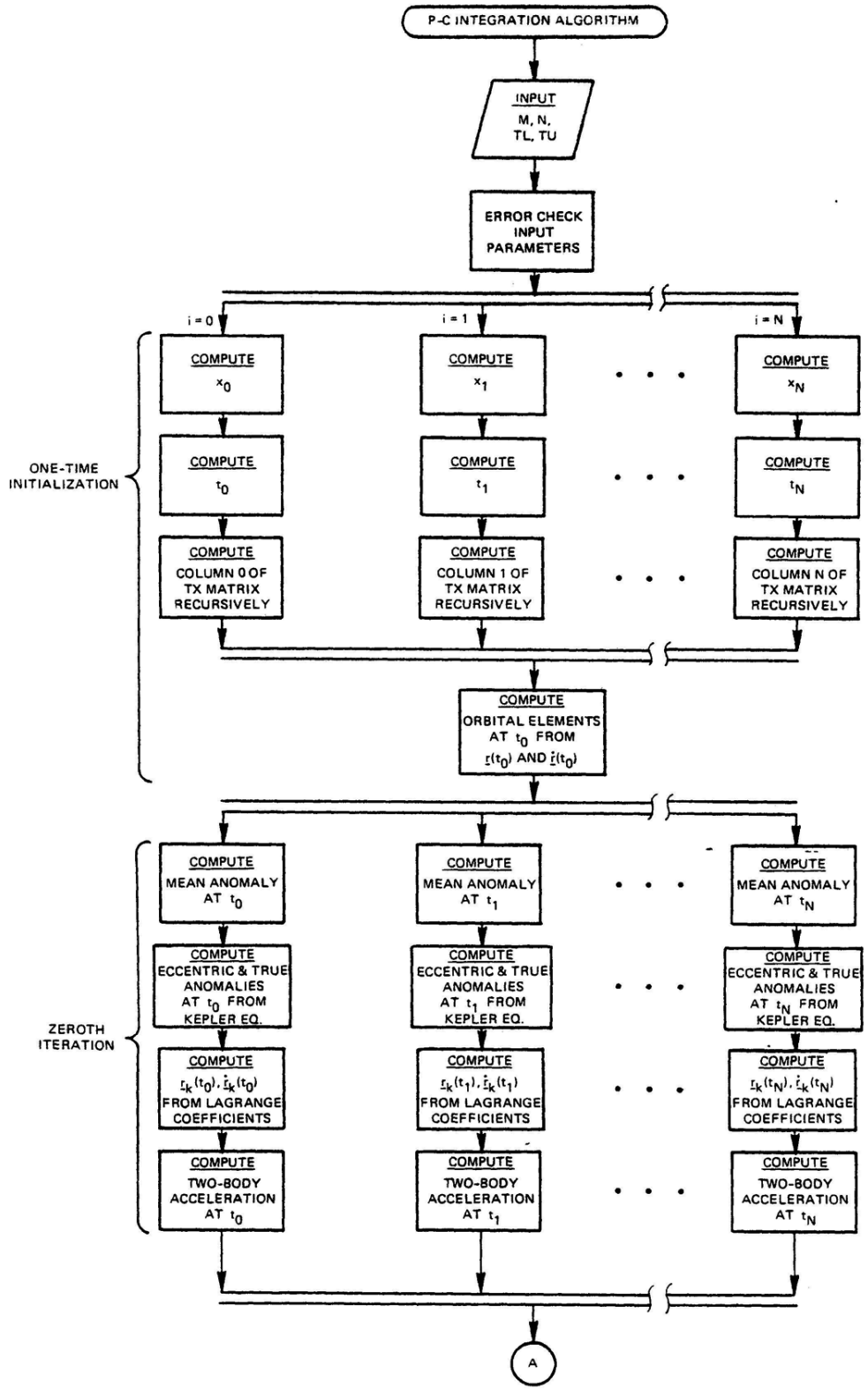


Figure 2.7. Picard-Chebyshev Parallel Orbit Propagation Algorithm (Page 1 of 2)

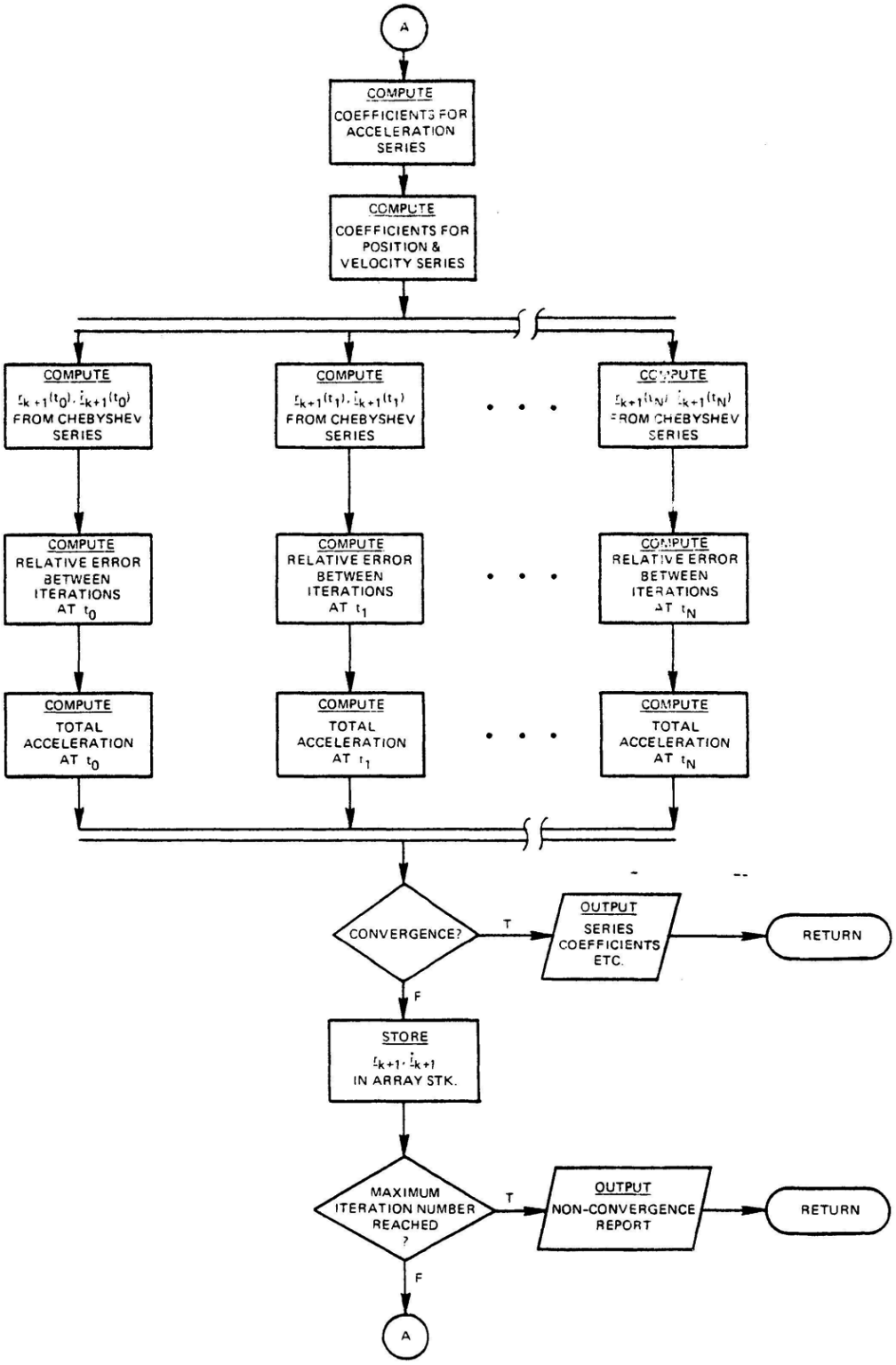


Figure 2.7. Picard-Chebyshev Parallel Orbit Propagation Algorithm (Page 2 of 2)

This will allow these N + 1 computations to be computed in the effective time required to have one of these computations performed by a sequential processor of equivalent characteristics (instruction execution time, memory access time, etc.). The potential performance improvement occasioned by this high degree of parallelism in the Picard-Chebyshev algorithm will be discussed further in Chapter 4.

2.4 Numerical Results

The Picard-Chebyshev integration algorithm was implemented in two different software versions to test the numerical accuracy of the method. First, the algorithm was implemented in the research and development version of the Goddard Trajectory Determination System (GTDS-RD). This system (46) is a highly flexible orbit determination system and a version of this software is currently resident at the Charles Stark Draper Laboratory as a software test bed for precision orbit propagation/determination studies. The second version of the Picard-Chebyshev integrator is a stand-alone version which served as a baseline for the later implementation of the parallel orbit estimation algorithm to be described in Chapter 3.

The initial implementation of the Picard-Chebyshev method in the GTDS-RD software was undertaken for three reasons. First, the software interfaces for the Picard-Chebyshev integration of the Cowell equations of motion already existed due to the previous work of Feagin. Test runs comparing the previous P-C method to that proposed in this work indicate some accuracy improvement with the proposed method. Second, the GTDS-RD system has extensive models for the perturbed VOP element rates for both the high precision and averaged VOP formulations. This greatly simplified the incorporation of the Picard-Chebyshev method into the VOP formulations, and the resulting implementation showed some pleasantly surprising accuracy and performance characteristics. Third, the GTDS-RD package contains a relatively sophisticated compare capability, allowing two trajectories to be compared and certain difference quantities and histories to be plotted. This provides a straightforward method of evaluating the Picard-Chebyshev integrations by directly comparing them against precise Cowell integrations using the same initial conditions and force models. The modifications to GTDS-RD which were required to implement the Picard-Chebyshev orbit propagation method are described in Appendix C.

The stand-alone capability was implemented to serve as a baseline orbit propagator for the parallel orbit estimation algorithm. This version has available to it much simpler force models than the GTDS-RD implementation. The gravitational perturbing potential is represented only by zonal harmonics J_2 through J_8 (selectable). The drag perturbation may be included, but the atmosphere is assumed to be spherical and rotating as a rigid body about the rotating Earth. The atmospheric density model is assumed to have an exponential form. No tesseral or sectoral spherical harmonics are included in the gravitational model and no third body effects are considered. A description of the stand-alone version is included as Appendix D.

2.4.1 General Test Case Descriptions

The comparison of the Picard-Chebyshev orbit propagation algorithm to the baseline high precision Cowell integration has been accomplished for two general sets of initial conditions. The first test case is termed a low altitude (perigee height = 200 km) near-circular case with an eccentricity of approximately 0.01. The epoch Keplerian elements and the associated Cartesian position and velocity components are listed in Table 2.3. The second general

Table 2.3

Low Altitude Circular Test Case Epoch Conditions

EPOCH STATE		EPOCH ELEMENTS	
r_x =	-464.856 km	a =	6644.586 km
r_y =	6667.800 km	e =	0.01
r_z =	574.231 km	i =	67.985 deg
v_x =	-2.8381186 km/sec	ω =	200.674 deg
v_y =	-0.7871898 km/sec	Ω =	91.977 deg
v_z =	7.0830275 km/sec	M =	164.317 deg

PERIOD = 90 min.

Table 2.4

Low Altitude Elliptical Test Case Epoch Conditions

EPOCH STATE		EPOCH ELEMENTS	
r_x =	772.925 km	a =	8520.320 km
r_z =	8413.367 km	e =	0.238
r_y =	-2488.811 km	i =	68.064 deg
v_x =	-2.975946 km/sec	ω =	93.811 deg
v_y =	0.407087 km/sec	Ω =	91.564 deg
v_z =	6.118241 km/sec	M =	275.328 deg

PERIOD = 130 min.

case is considered to be a low altitude (perigee height = 116 km) elliptical case with an eccentricity of 0.238. Table 2.4 lists the epoch values for the integration initial conditions for this case.

Within each of the above test cases, several different runs were made using various combinations of stepsizes, equation of motion formulations, order of Chebyshev approximating series, number of orthogonality points and type of perturbation sources. Initial runs were made comparing the P-C integrator to the analytical two body solution provided by the Lagrange F, G, F_t and G_t coefficients. These comparisons provide a heuristic demonstration of the numerical error propagation in the algorithm. For the circular test case, runs were then made using zonal harmonics through J_8 and comparing with a Cowell integration of the same problem using a thirty second stepsize. Drag was then added to the perturbation field and another set of comparisons made. Similar runs were made for the elliptical test case, but the baseline for comparison was taken to be a 200 step/revolution time regularized Cowell integration, due to the eccentricity of the orbit. The varying test cases for the low altitude circular test case are outlined in Table 2.5, while those for the elliptical case are presented in Table 2.6.

Table 2.5. Low Altitude Circular Test Cases for Picard-Chebyshev Integrator

PICARD-CHEBYSHEV INTEGRATOR: SUMMARY OF TEST RUNS

RUN #	EQUATION SET	STEPSIZE		M†	N††	FIELD	DRAG	SPAN
		COWELL	P-C					
1	COWELL	30 s.	5400 s.	64	69	8x0	NO	3 HRS.
2	COWELL	30 s.	5400 s.	64	69	8x0	YES	3 HRS.
3	COWELL	30 s.	2700 s.	64	69	8x0	YES	3 HRS.
4	VOP-EQ(λ)	30 s.	5400 s.	48	69	8x0	YES	3 HRS.
5	VOP-EQ(λ)	30 s.	2700 s.	48	69	8x0	YES	3 HRS.
6	VOP-EQ(F)	30 s.	2700 s.	48	69	8x0	YES	3 HRS.
7	VOP-EQ(λ)	30 s.	5400 s.	22	49	8x0	YES	3 HRS.
8	AVE. VOP-EQ	1 DAY	40 DAY	36	49	8x0	NO	30 DAYS
9	AVE. VOP-EQ	1 DAY	40 DAY	36	49	8x0	NO	180 DAYS

† M = order of Chebyshev approximating polynomial

†† N = number of Chebyshev orthogonality points minus one

Table 2.6. Low Altitude Elliptical Test Cases for Picard-Chebyshev Integrator

PICARD-CHEBYSHEV INTEGRATOR: SUMMARY OF TEST RUNS

RUN #	EQUATION SET	STEPSIZE COWELL	STEPSIZE P-C	M†	N††	FIELD	DRAG	SPAN
1	COWELL	200/REV	8100 s.	64	69	8x0	NO	4.5 HRS.
2	COWELL	200/REV	8100 s.	64	69	8x0	YES	4.5 HRS.
3	COWELL	200/REV	4050 s.	64	69	8x0	YES	4.5 HRS.
4	COWELL	200/REV	2025 s.	64	69	8x0	YES	4.5 HRS.
5	VOP-EQ(λ)	200/REV	8100 s.	48	69	8x0	YES	4.5 HRS.
6*	VOP-EQ(λ)	200/REV	8100 s.	48	69	8x0	YES	4.5 HRS.
7*	VOP-EQ(λ)	200/REV	4050 s.	48	69	8x0	YES	4.5 HRS.
8	VOP-EQ(λ)	200/REV	8100 s.	48	69	6x6	NO	4.5 HRS.

† M = order of Chebyshev approximating polynomial

†† N = number of Chebyshev orthogonality points minus one

* In these runs, epoch mean anomaly was set to zero degrees.

2.4.2 Analytical Comparison Test Results

In order to determine the nature of the numerical error propagation in the Picard-Chebyshev integrator, a series of test runs was made to compare numerical integrations of the two body problem (Cowell Formulation) to known analytical solutions for this problem. The analytical solution technique used was the method of Lagrange F, G, F_t and G_t coefficients. A derivation of this method is presented in Appendix A. Differences between the numerically integrated position and velocity vector components and the analytical solution were computed and plotted. The initial conditions for the comparisons were the low altitude circular test case conditions given in Table 2.3. The Picard-Chebyshev integrator used a Chebyshev series of order 48 to fit the acceleration components and 70 orthogonality points. The integration stepsize was varied in different runs.

In the first run, the integration stepsize was taken to be 5400 seconds (1 satellite revolution) and the comparison was carried out over 24 hours with output every two minutes. Figures 2.8 through 2.10 present the time history of the velocity error. These are the errors in the first integrals of the Chebyshev series representing

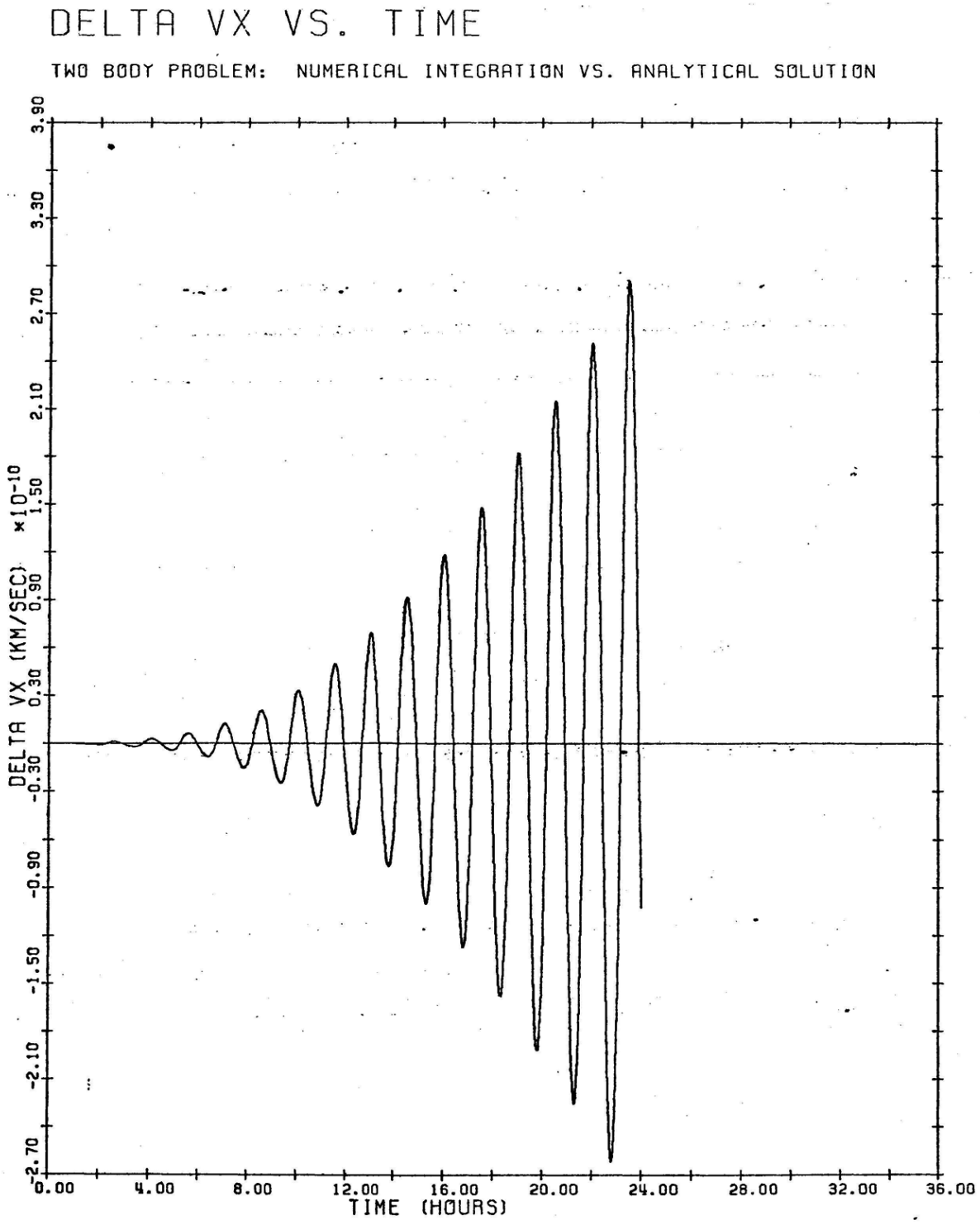


Figure 2.8. Difference in "x" Velocity Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)

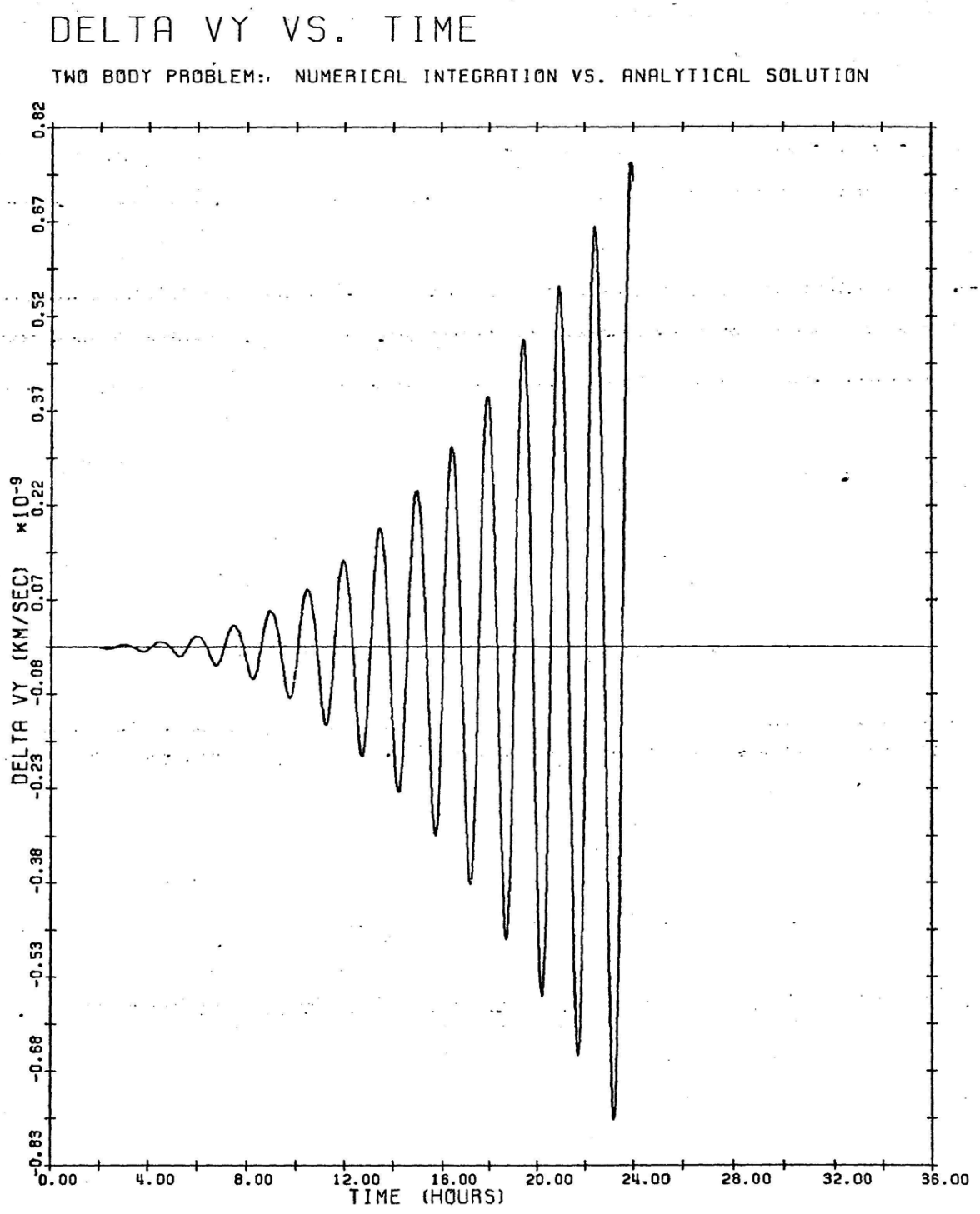


Figure 2.9. Difference in "y" Velocity Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)

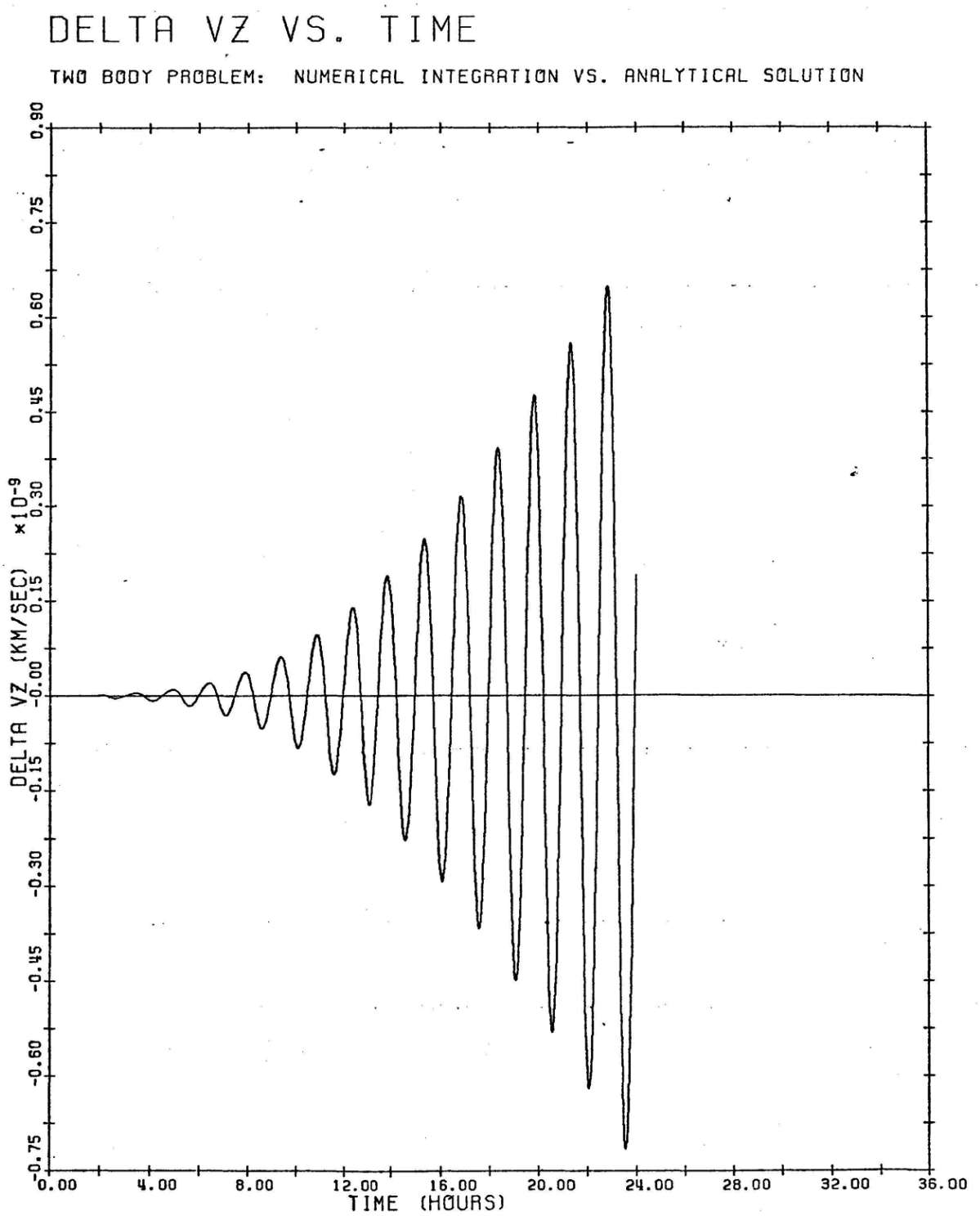


Figure 2.10. Difference in "z" Velocity Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)

the acceleration components. Several interesting characteristics are seen from these error curves. First, the velocity errors appear to grow without bound and to have an envelope that is growing quadratically with time. The error has an oscillatory nature with the primary frequency being the satellite frequency. The position component errors given in Figures 2.11 through 2.13 show the same characteristics. Additionally, when the Chebyshev series for the acceleration components are evaluated and compared against the accelerations computed from the analytical two body model, error histories with similar characteristics result. These plots are given in Figures 2.14 through 2.16.

If, for any given component, the error curves for acceleration and velocity are superimposed, the error histories are shifted in phase by 90°. This is to be expected, since the velocity error is in essence an integral of the acceleration error. Integration of a quasi-periodic function, such as the acceleration error, results in a phase shift of 90°. A similar result is seen if the velocity and position error curves for a given component are superimposed.

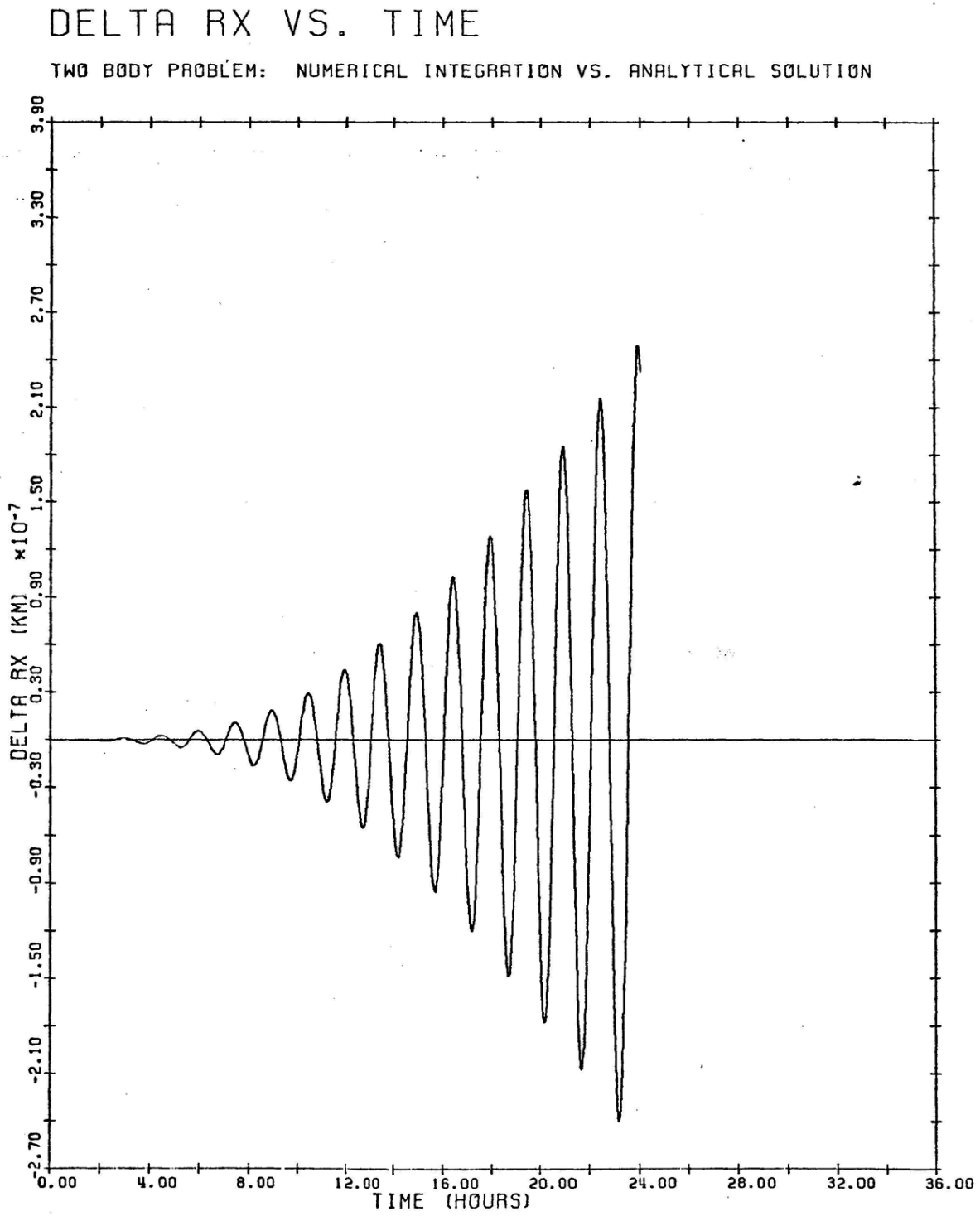


Figure 2.11. Difference in "x" Position Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)

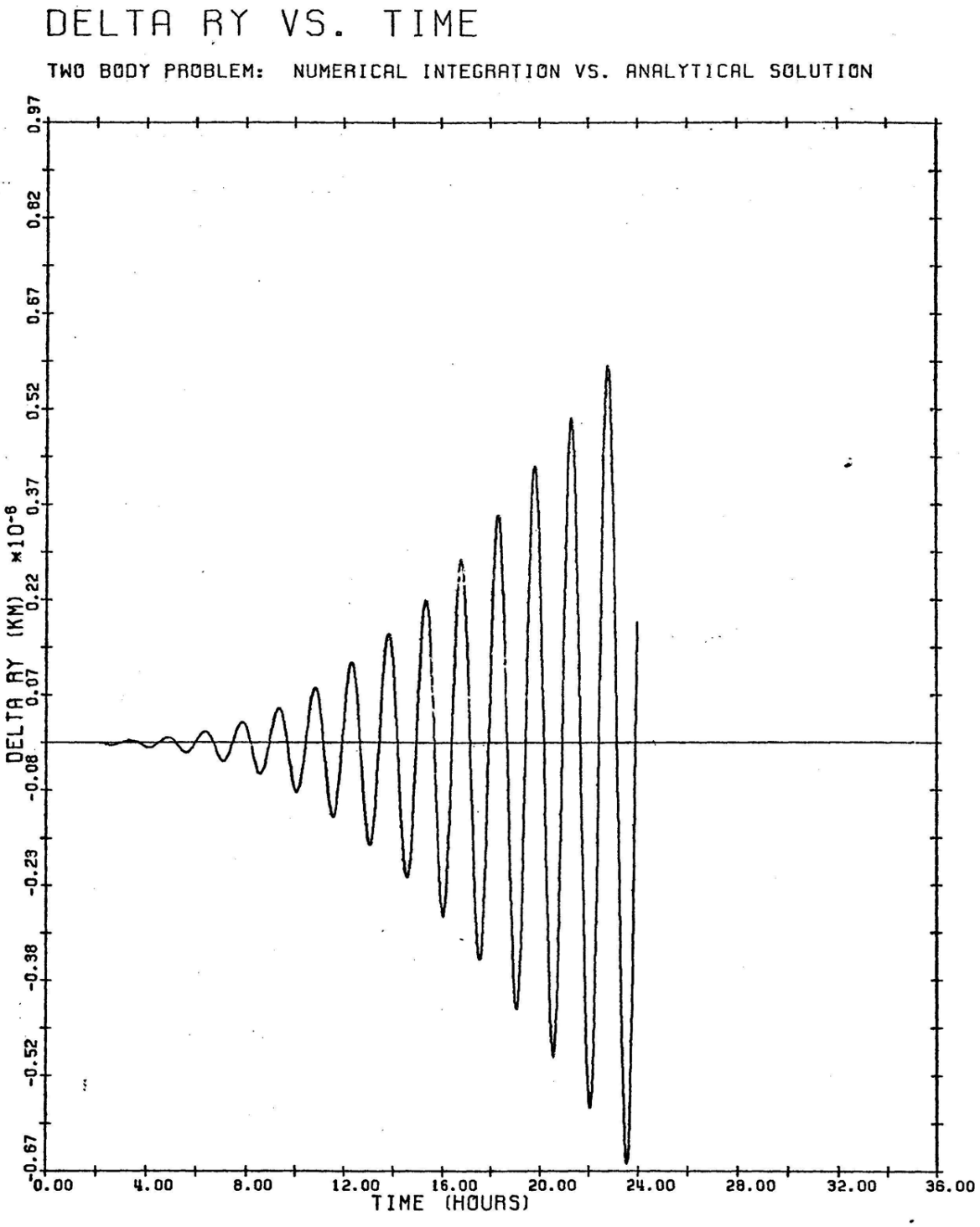


Figure 2.12. Difference in "y" Position Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)

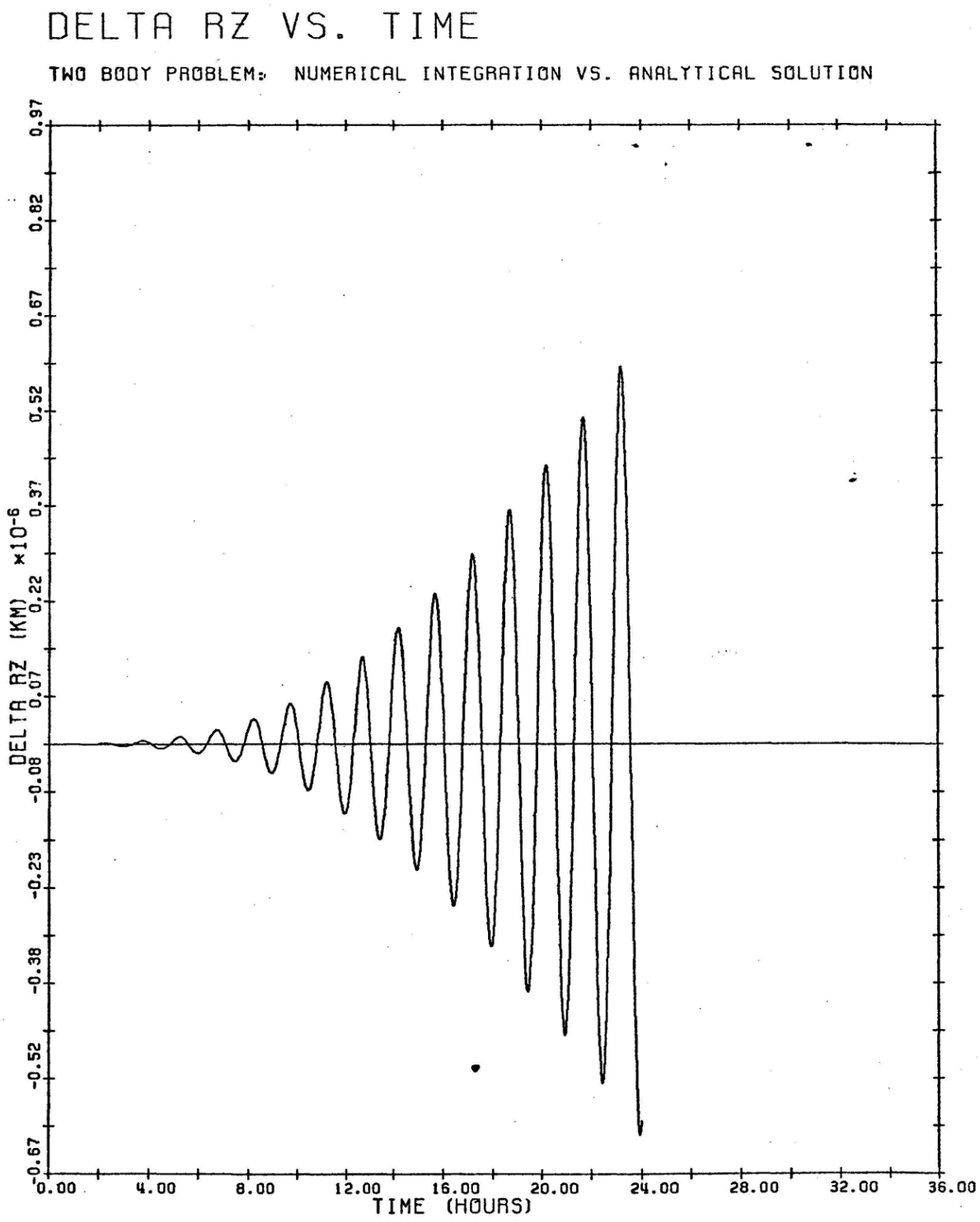


Figure 2.13. Difference in "z" Position Component: Picard-Chebyshev Numerical Integration vs. Analytical Solution (Integration Stepsize = 5400 sec.)

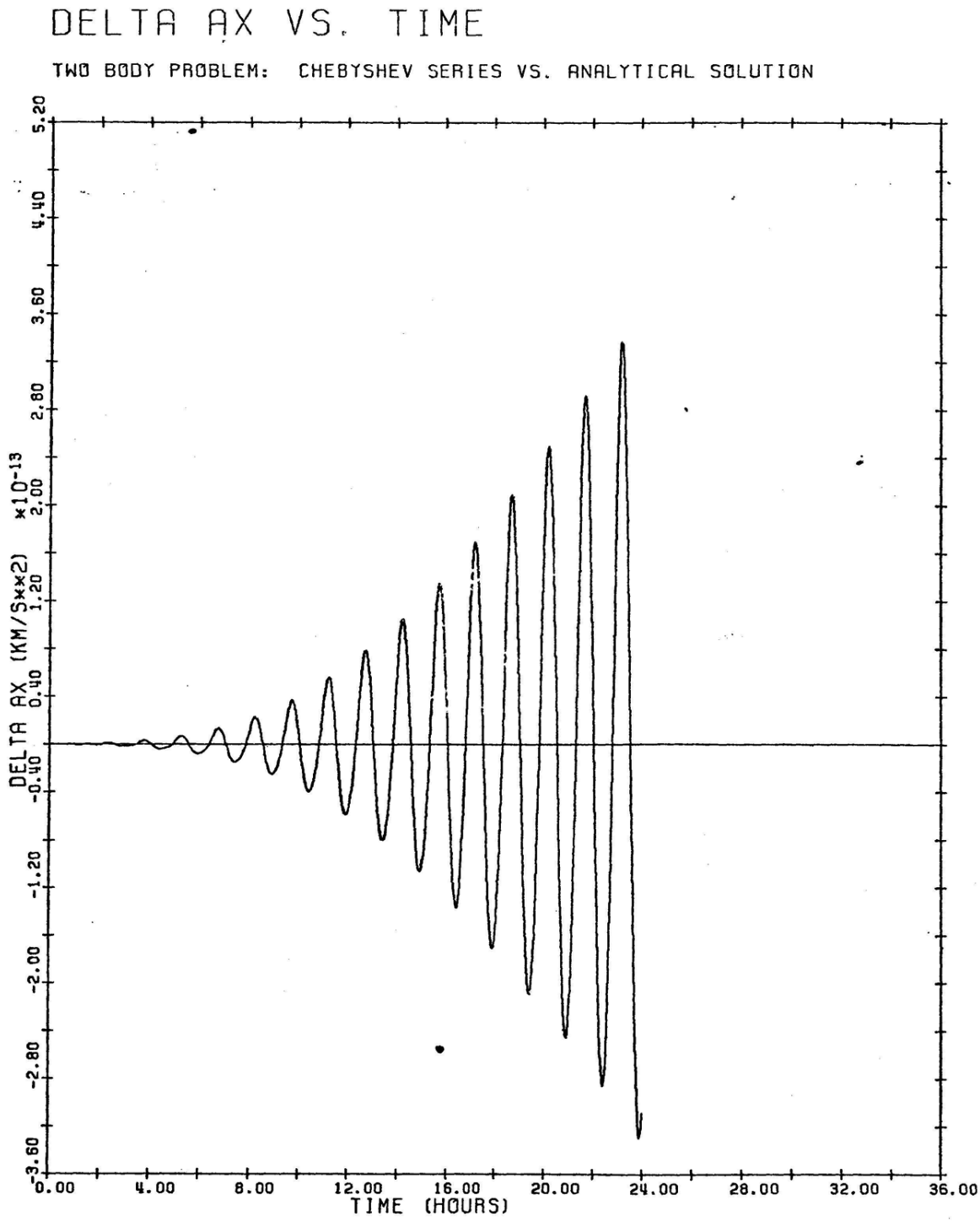


Figure 2.14. Difference in "x" Acceleration Component: Chebyshev Series Fit versus Analytical Solution

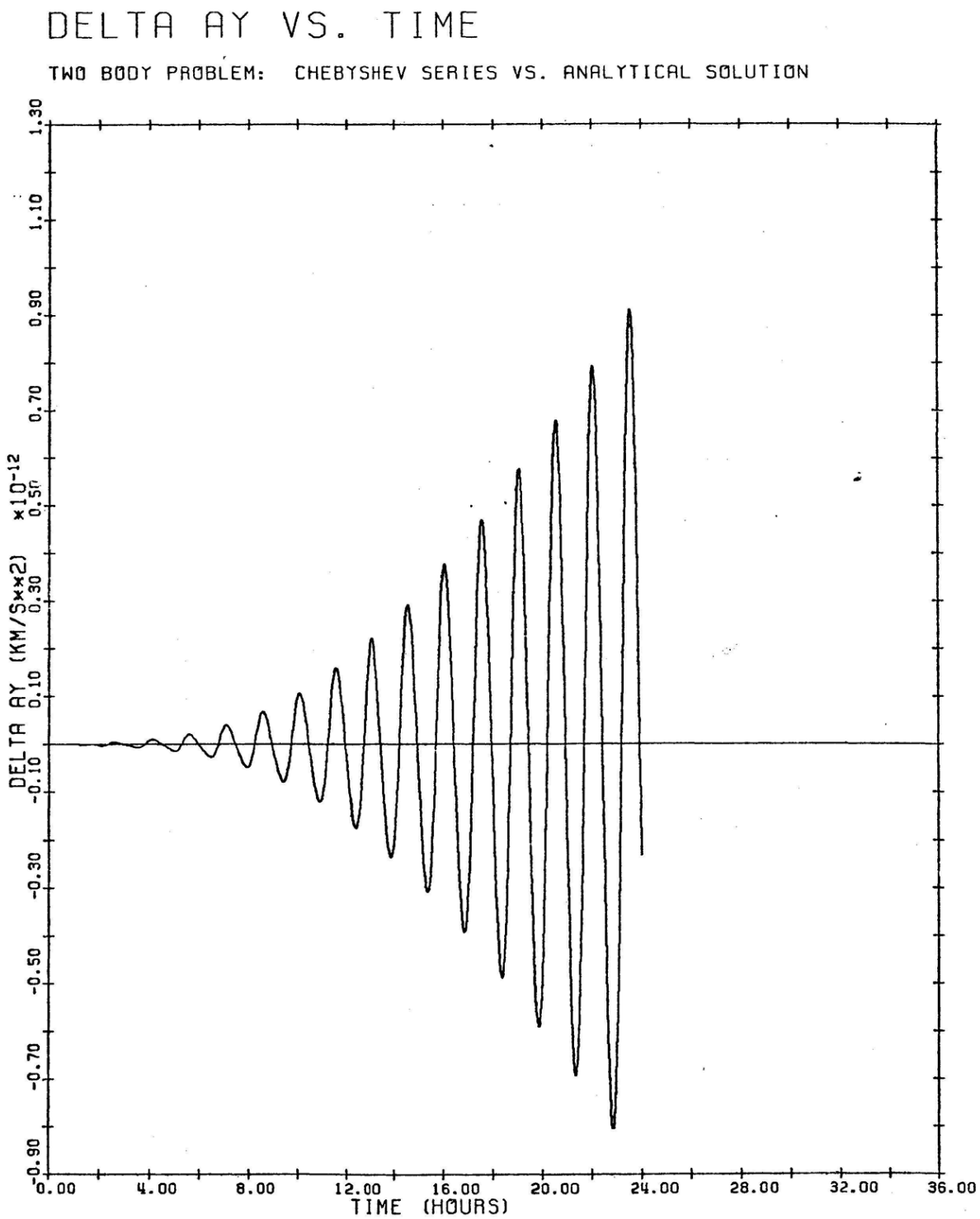


Figure 2.15. Difference in "y" Acceleration Component: Chebyshev Series Fit versus Analytical Solution

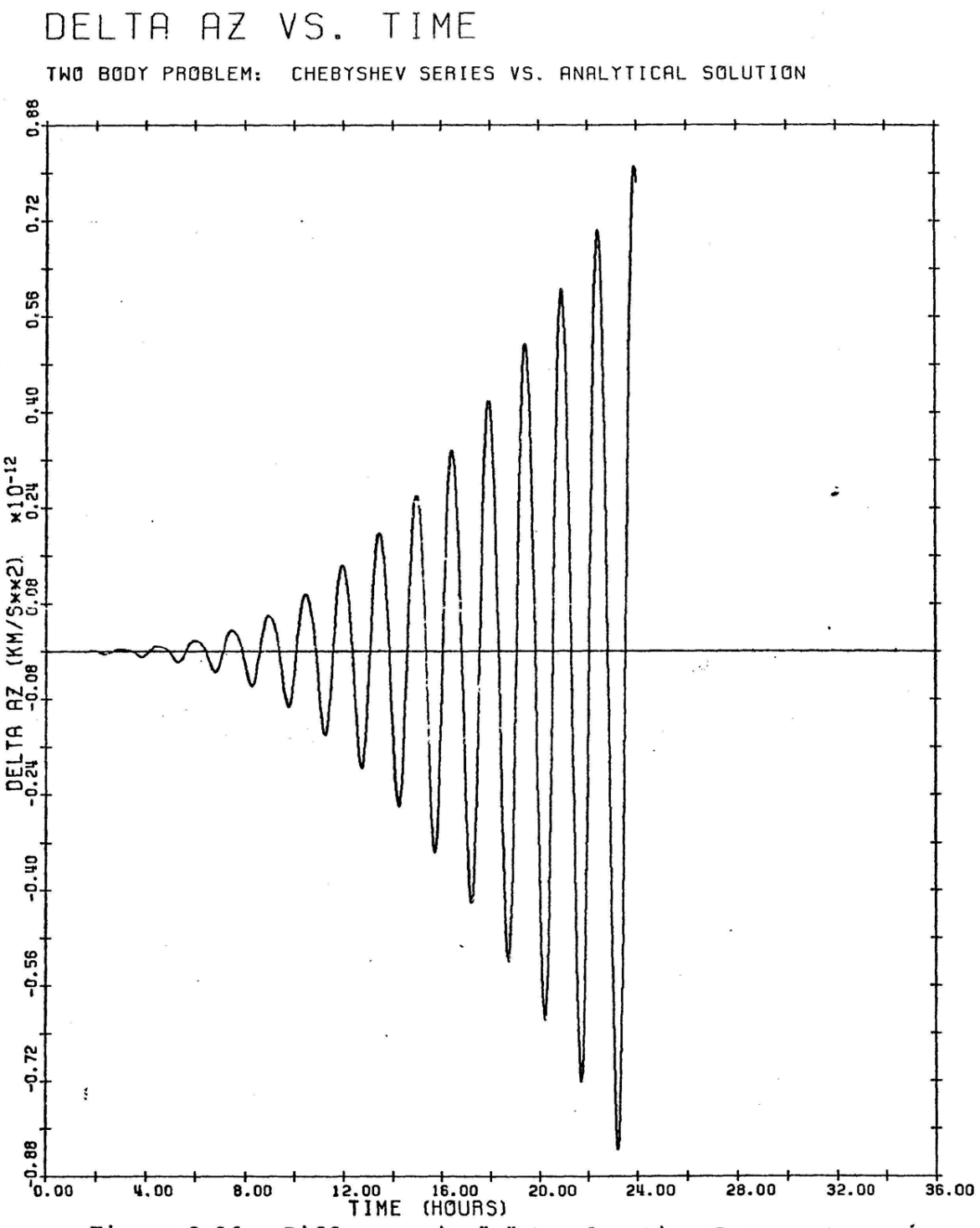


Figure 2.16. Difference in "z" Acceleration Component: Chebyshev Series Fit versus Analytical Solution

It is felt that the primary contribution to this error propagation is the initial condition error induced at the beginning of each integration arc in the Picard-Chebyshev method. In the first arc the initial condition error is zero, because the position and velocity series are required to match exactly the given epoch conditions. However, because the Chebyshev fit to the acceleration components over the first arc is not exact and because of numerical round-off error, there is a slight error in the position and velocity components at the end of the first integration arc. Figure 2.17 and 2.18 plot the velocity and position errors over the first 90 minute integration arc. The numerical noise is clearly evident in these plots. The errors in the position and velocity components at the end of the first integration arc become errors in the initial conditions for the next arc. These initial condition errors are reflected in the series coefficients for the next arc as erroneous values for the zeroth order coefficients $B_{i,0}$ and $C_{i,0}$ $(i = 1,2,3)$ for that arc. The propagation of these errors as more integration arcs are taken is indicated by Figures 2.8 through 2.13. Detailed plots of the acceleration, velocity and position component errors over the last integration arc in the 24 hour compare span are given by Figures 2.19 through 2.21. The nearly periodic nature of the errors is evident. Additionally, comparisons

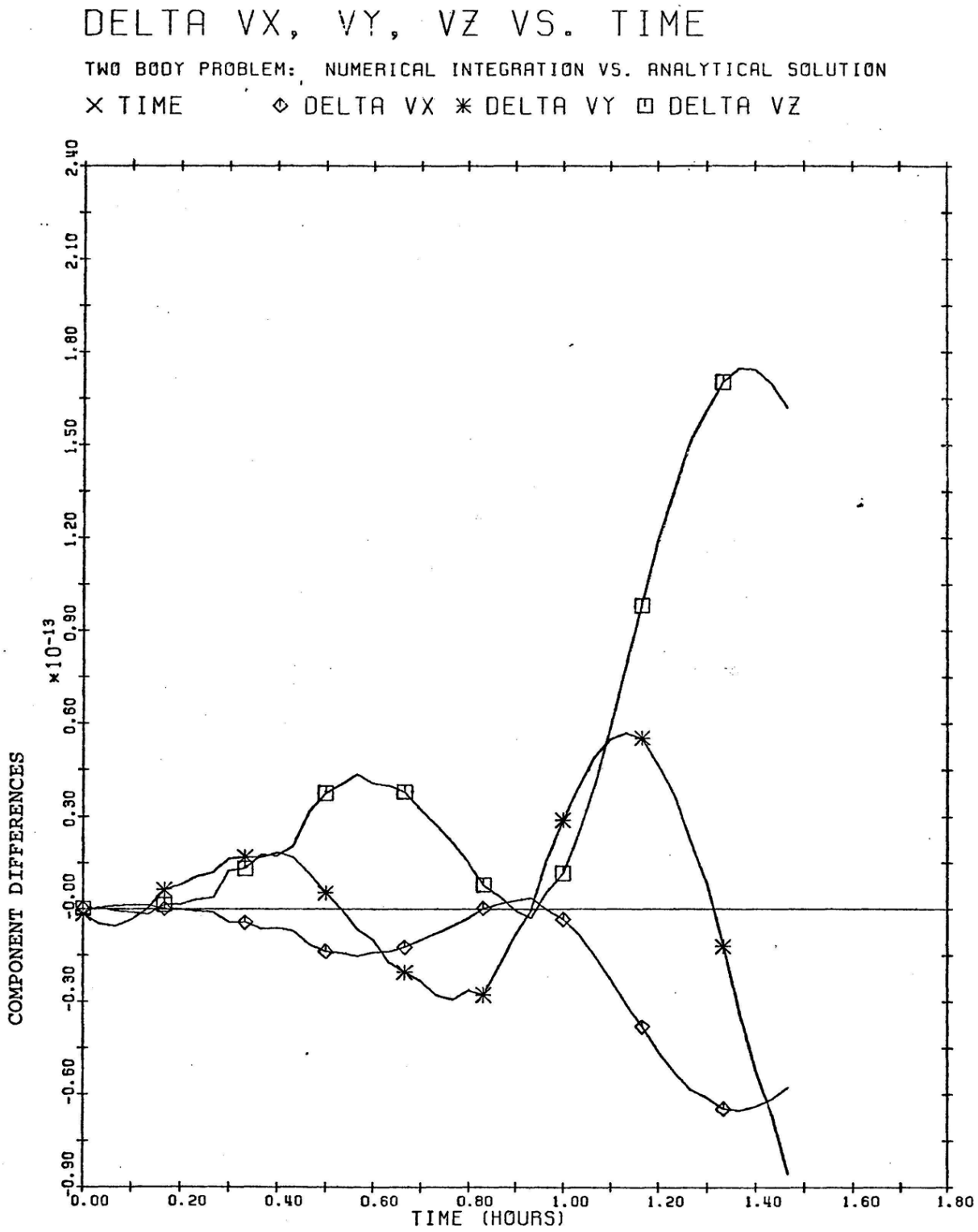


Figure 2.17. Velocity Component Differences During First Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution

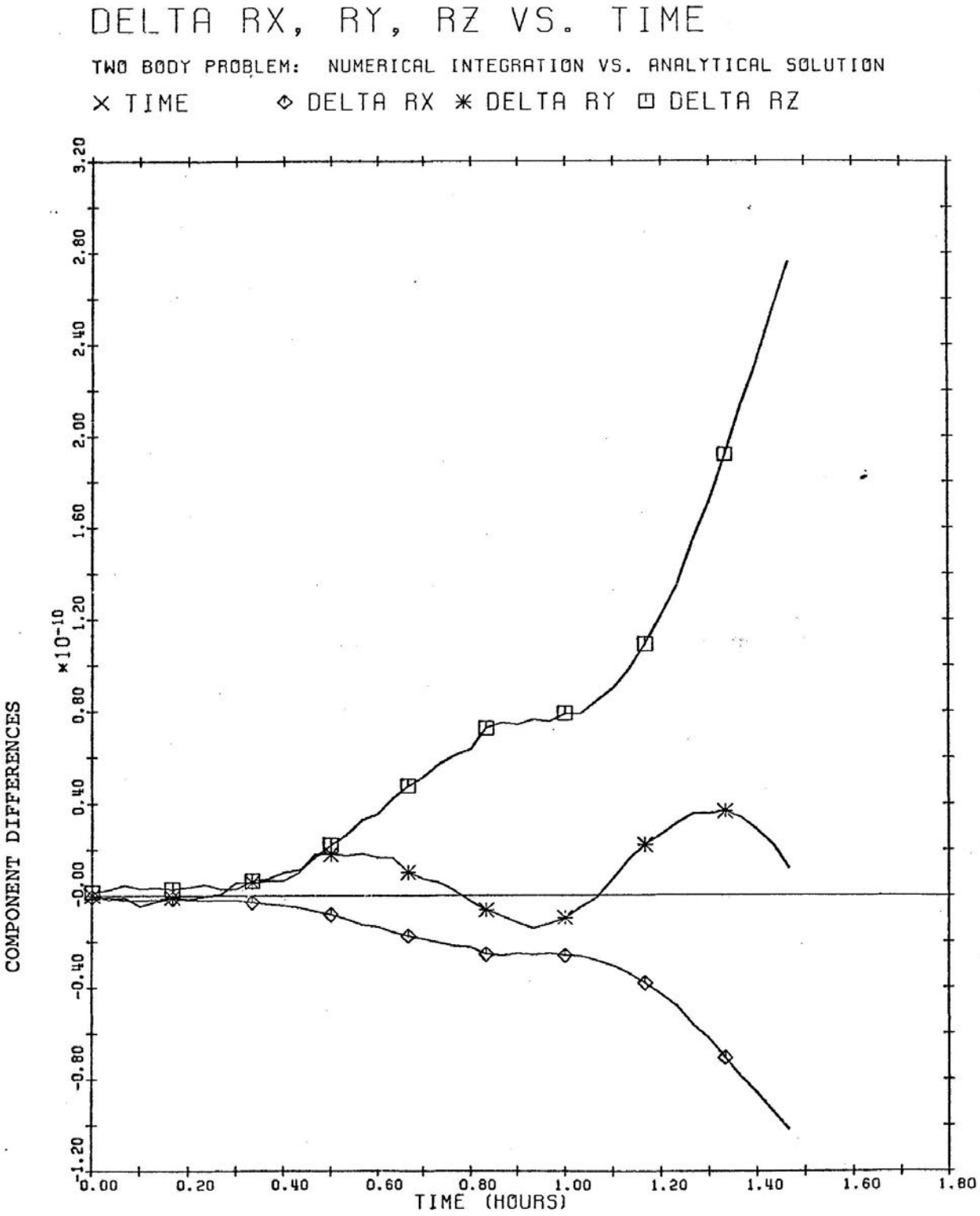


Figure 2.18. Position Component Differences During First Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution

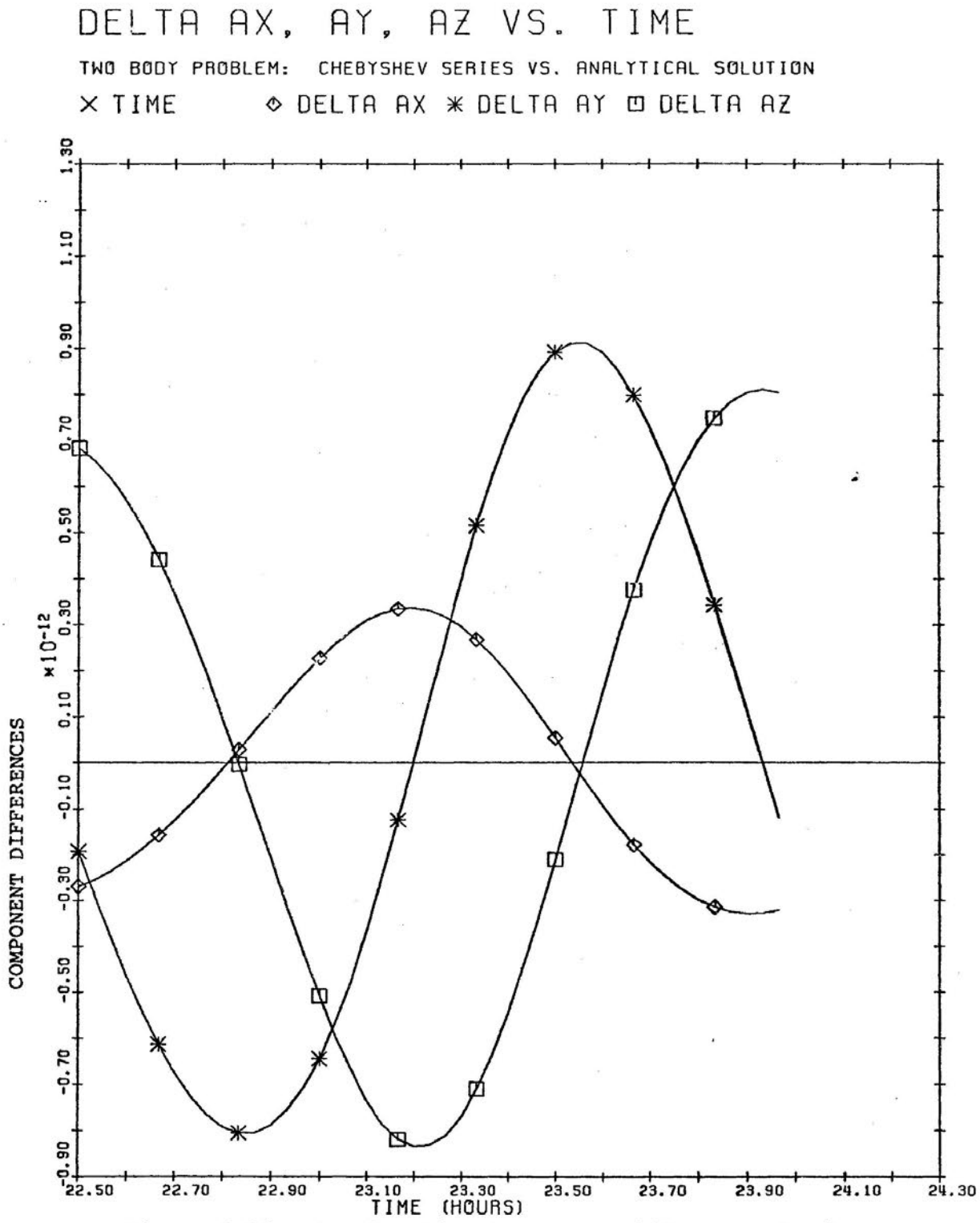


Figure 2.19. Acceleration Component Differences During Final Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution

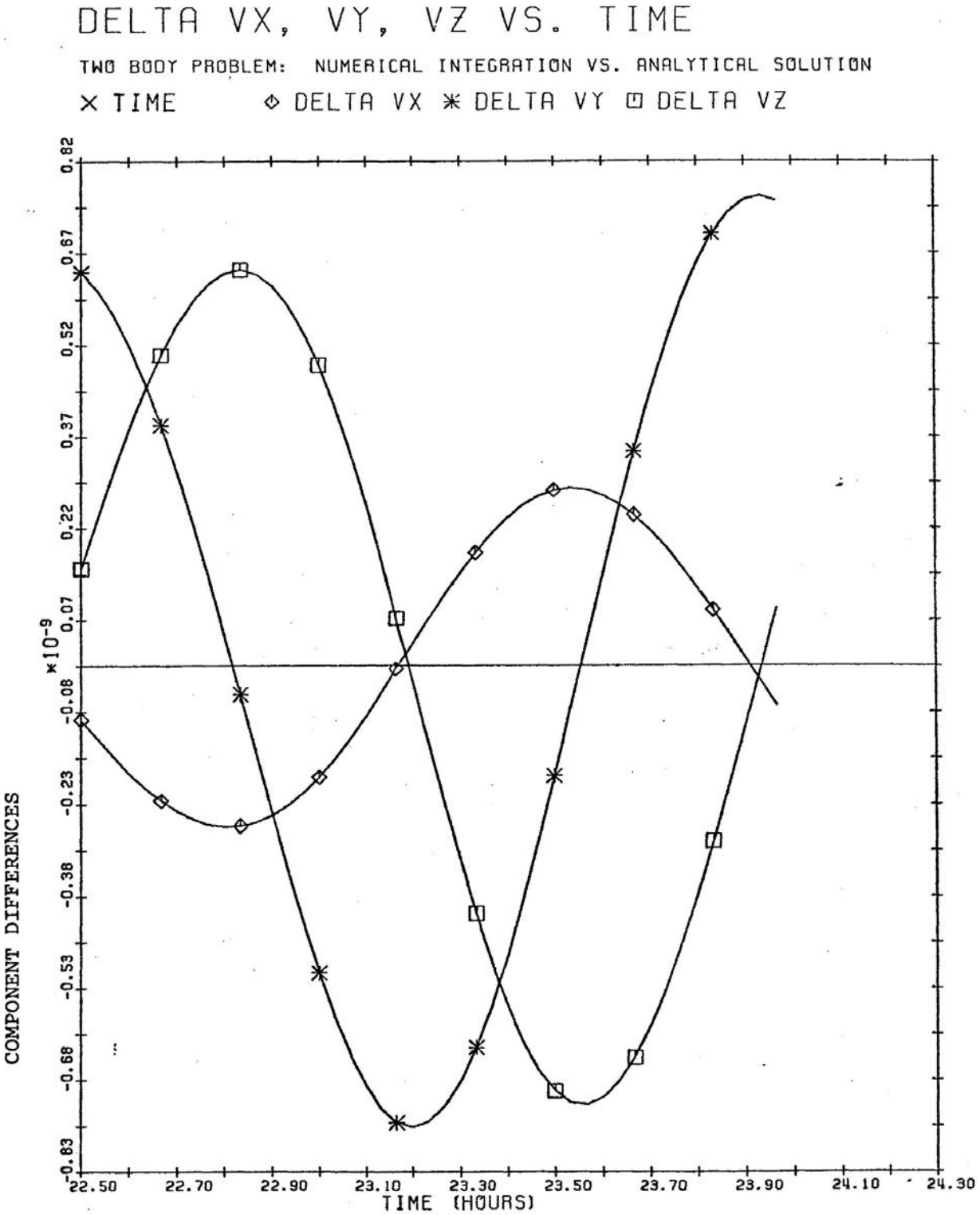


Figure 2.20. Velocity Component Differences During Final Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution

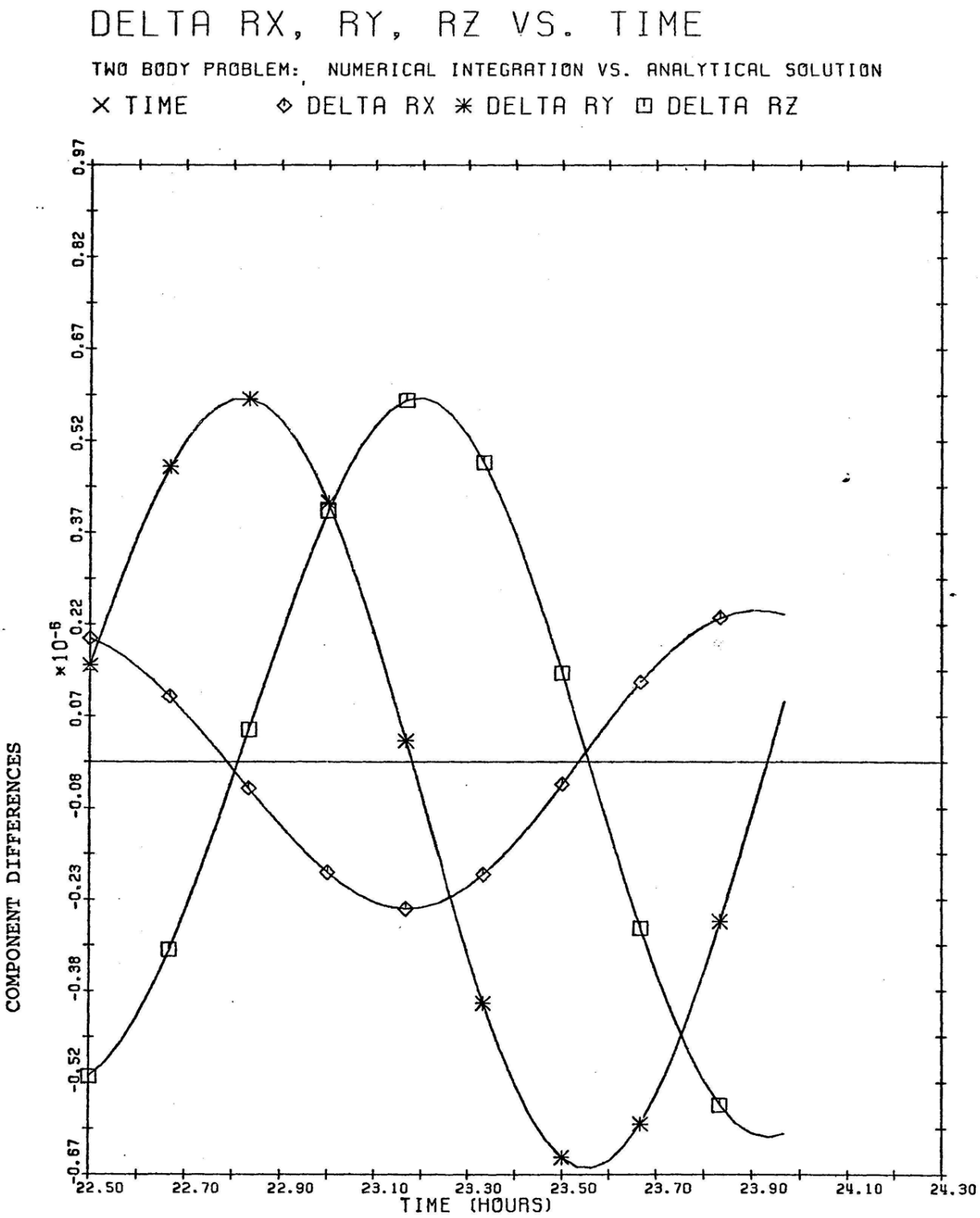


Figure 2.21. Position Component Differences During Final Ninety Minute Integration Arc: Picard-Chebyshev Integration versus Analytical Solution

of the plots clearly show that the velocity errors are shifted 90° in phase from the acceleration errors and the position errors are shifted 90° in phase from the velocity errors, as would be expected.

The error behavior detailed in the previous figures for the Picard-Chebyshev integrator is not unlike that for other types of numerical integrators. For example, the GTDS-RD software package was used to numerically integrate the Cowell equations of motion over 24 hours. The integrator selected was a 12^{th} order summed Cowell/Adams predictor-corrector method with a 60 second integration stepsize and output every two minutes. It should be noted that the 60 second stepsize is very conservative for an integration of the two body problem, where the only frequency is the satellite orbital frequency. At each output point, the two body solution is computed by the GTDS-RD subroutine 'TWOBDY' using the Lagrange f and g series solution. The position and velocity components from the analytical solution are subtracted from their respective values computed by the Cowell integrator and the results are plotted. Figures 2.22 through 2.24 present the velocity component differences and Figures 2.25 through 2.27 give the position component errors. The general characteristics of these error plots are essentially the same as

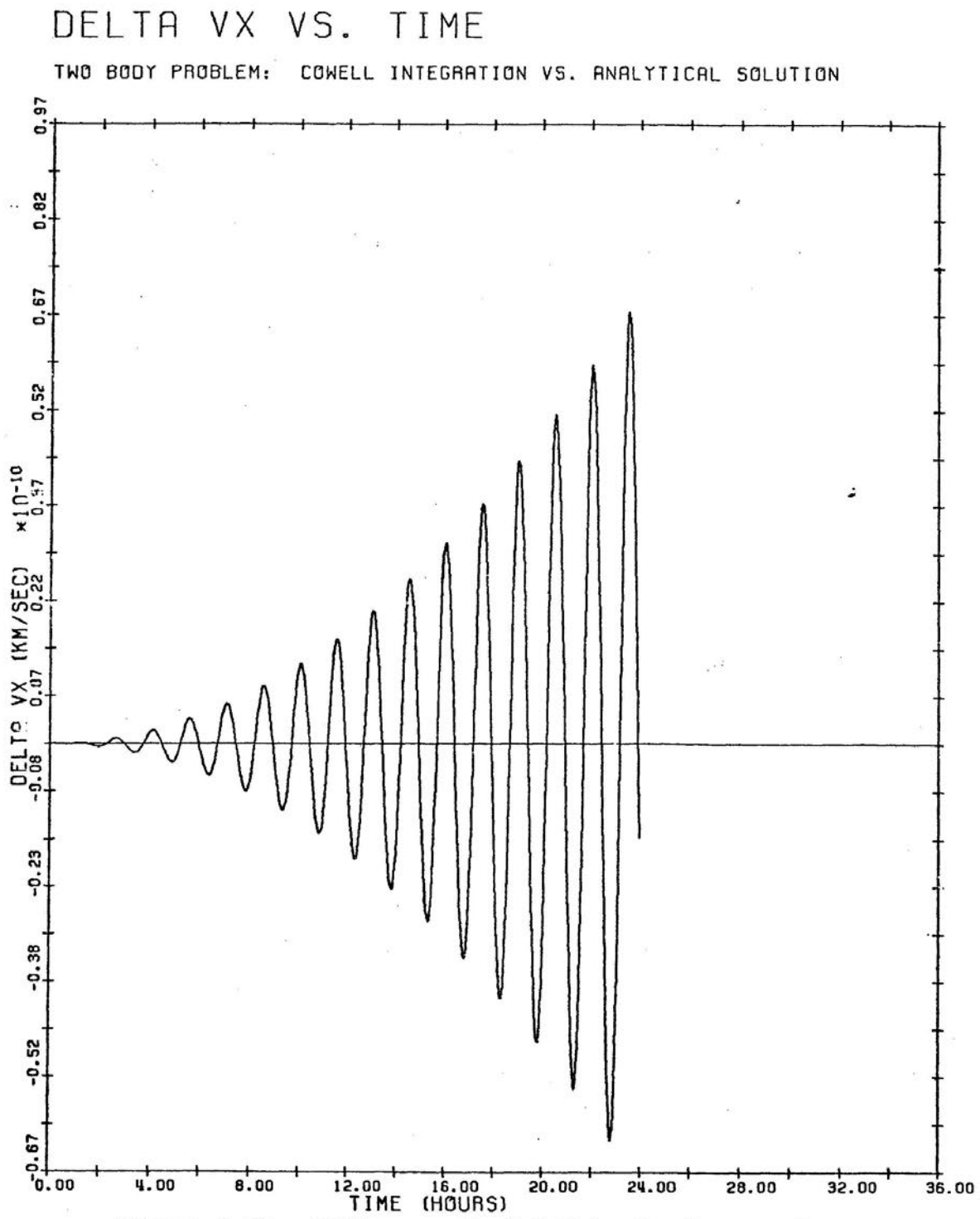


Figure 2.22. Difference in "x" Velocity Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)

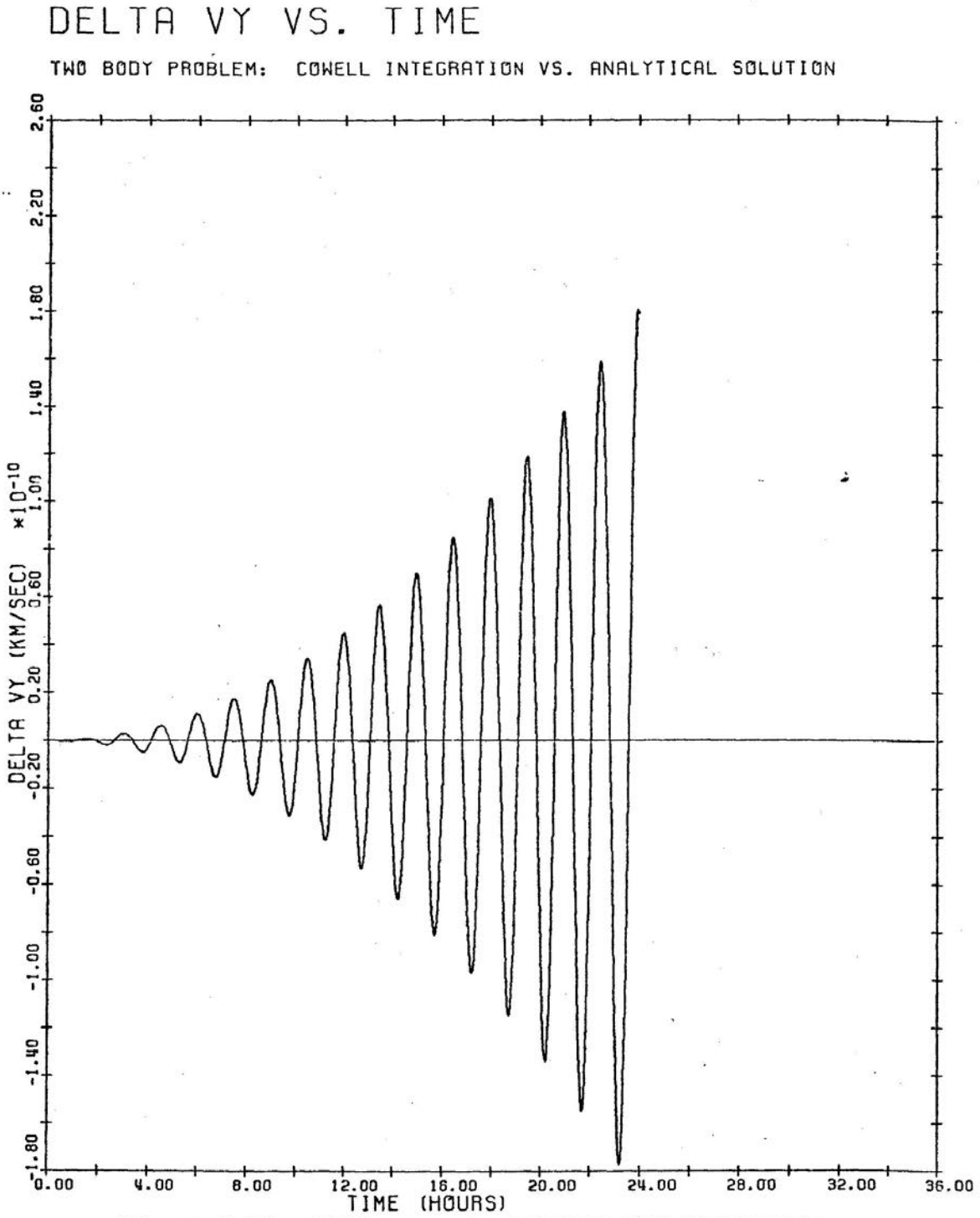


Figure 2.23. Difference in "y" Velocity Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)

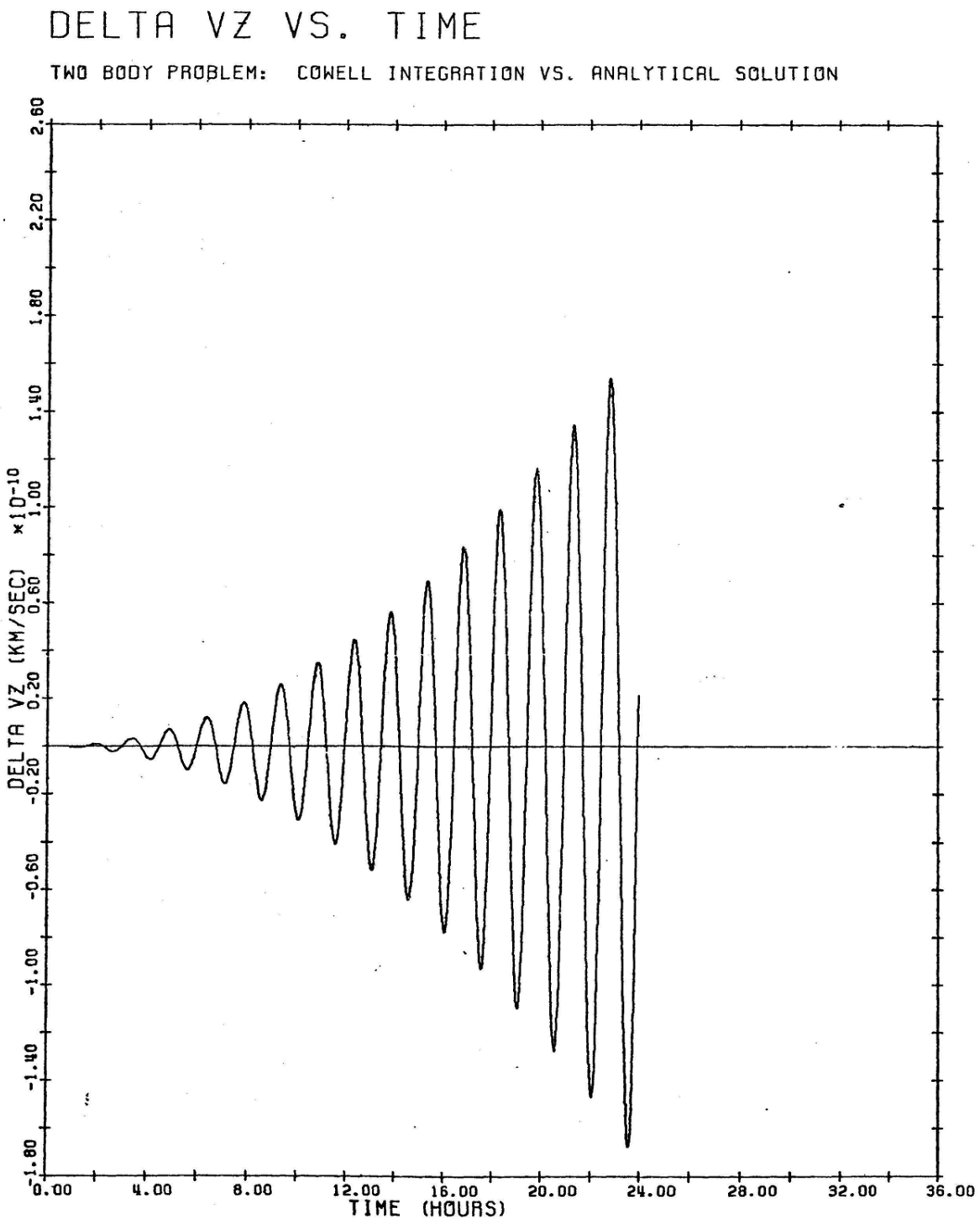


Figure 2.24. Difference in "z" Velocity Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)

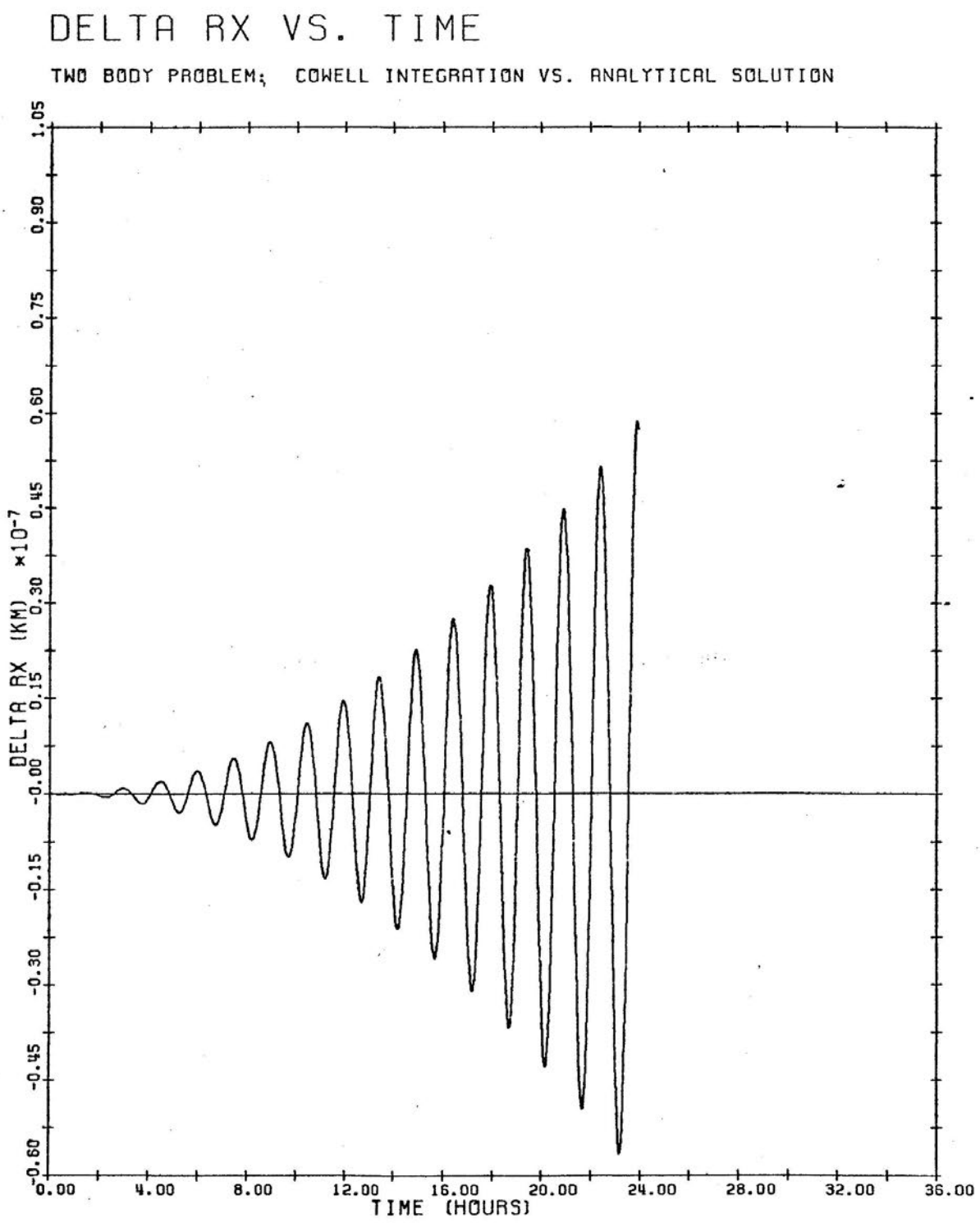


Figure 2.25. Difference in "x" Position Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)

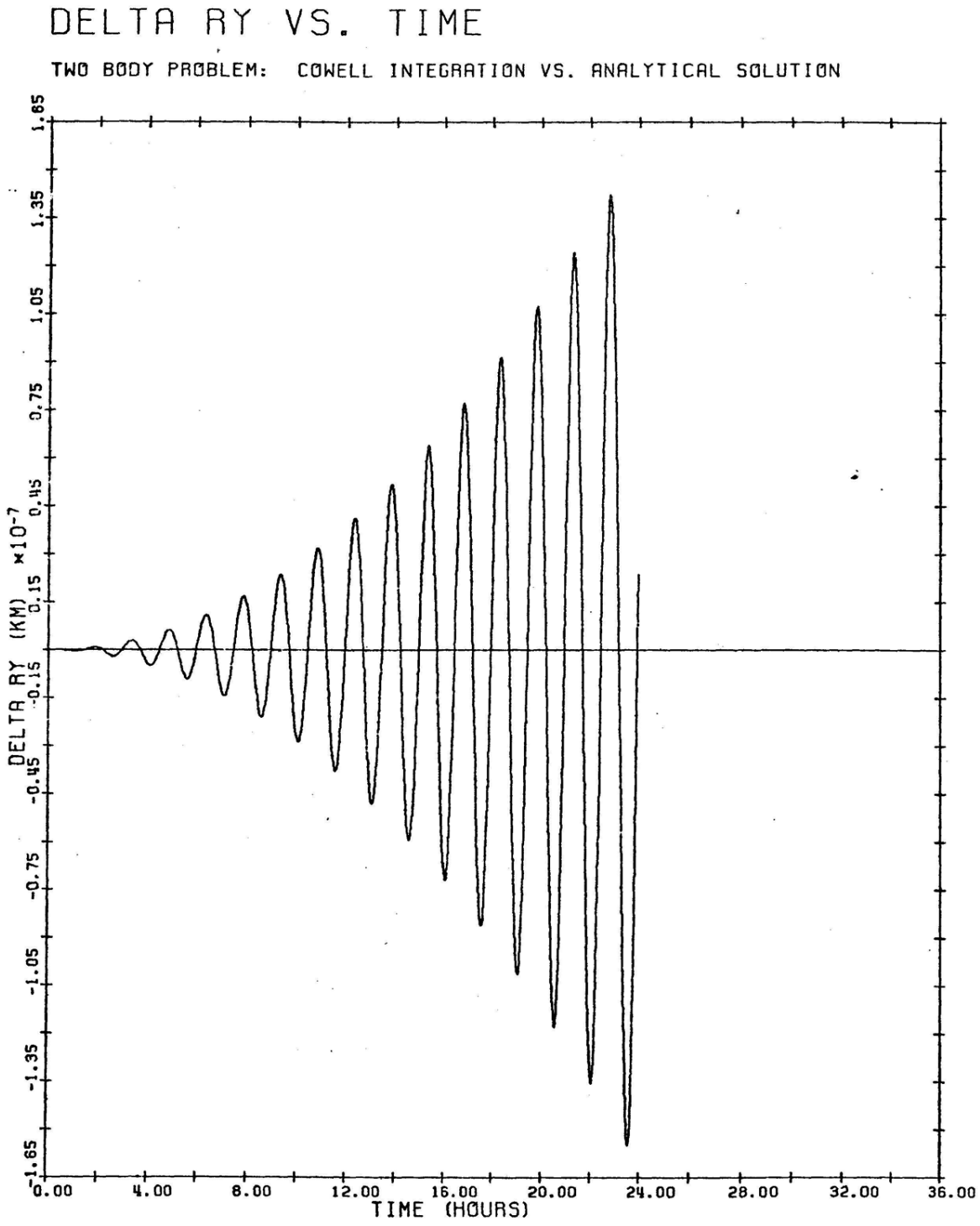


Figure 2.26. Difference in "y" Position Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)

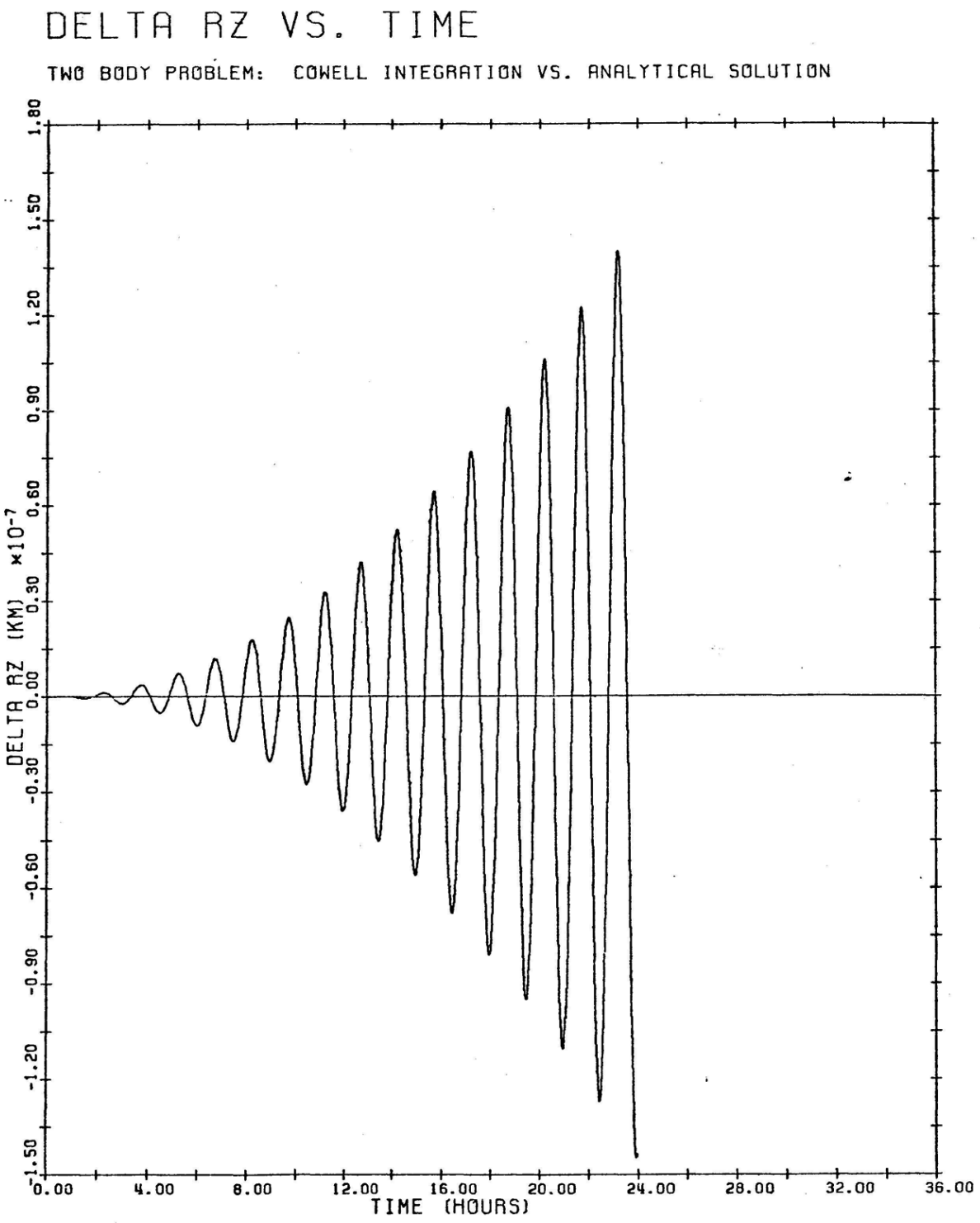


Figure 2.27. Difference in "z" Position Component: Cowell Numerical Integration versus Analytical Solution (Integration Stepsize = 60 sec.)

the characteristics of the error plots for the P-C integrator given in the previous figures. The oscillatory nature of the errors is again evident as is the unbounded growth.

When Figures 2.22 through 2.27 are compared against Figures 2.8 through 2.13 respectively, the Cowell errors are approximately one order of magnitude smaller during the 24 hour compare span. This is believed to be due to the length of the Picard-Chebyshev integration arc. Empirical results from the Picard-Chebyshev algorithm have indicated that an integration arc of one satellite revolution is generally the longest time span for which the algorithm will converge with reasonable accuracy. Shorter integration arcs will in general produce more accurate results. Figures 2.28 through 2.33 depict the velocity and position errors for a Picard-Chebyshev integration of the two body problem using an integration arc of one-half the satellite period (2700 sec.). Comparison of these results with the Cowell results indicate that the maximum differences in the Picard-Chebyshev results during the 24 hour compare span are of the same order as those of the Cowell results. Similarly, Figures 2.34 through 2.39 present the same data for the case of an integration arc of one-fourth the satellite period (1350 sec.). These errors

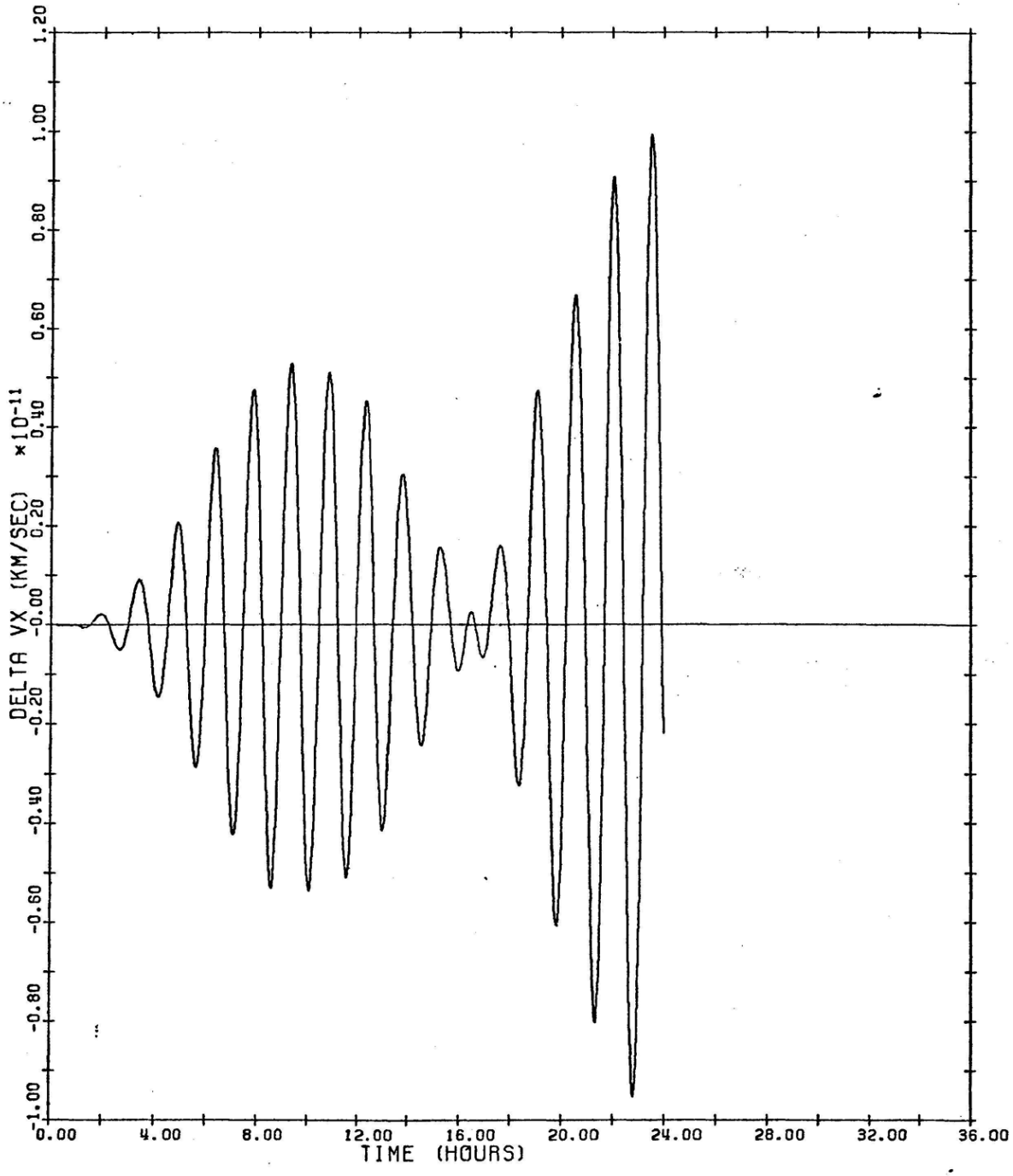


Figure 2.28. Difference in "x" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)

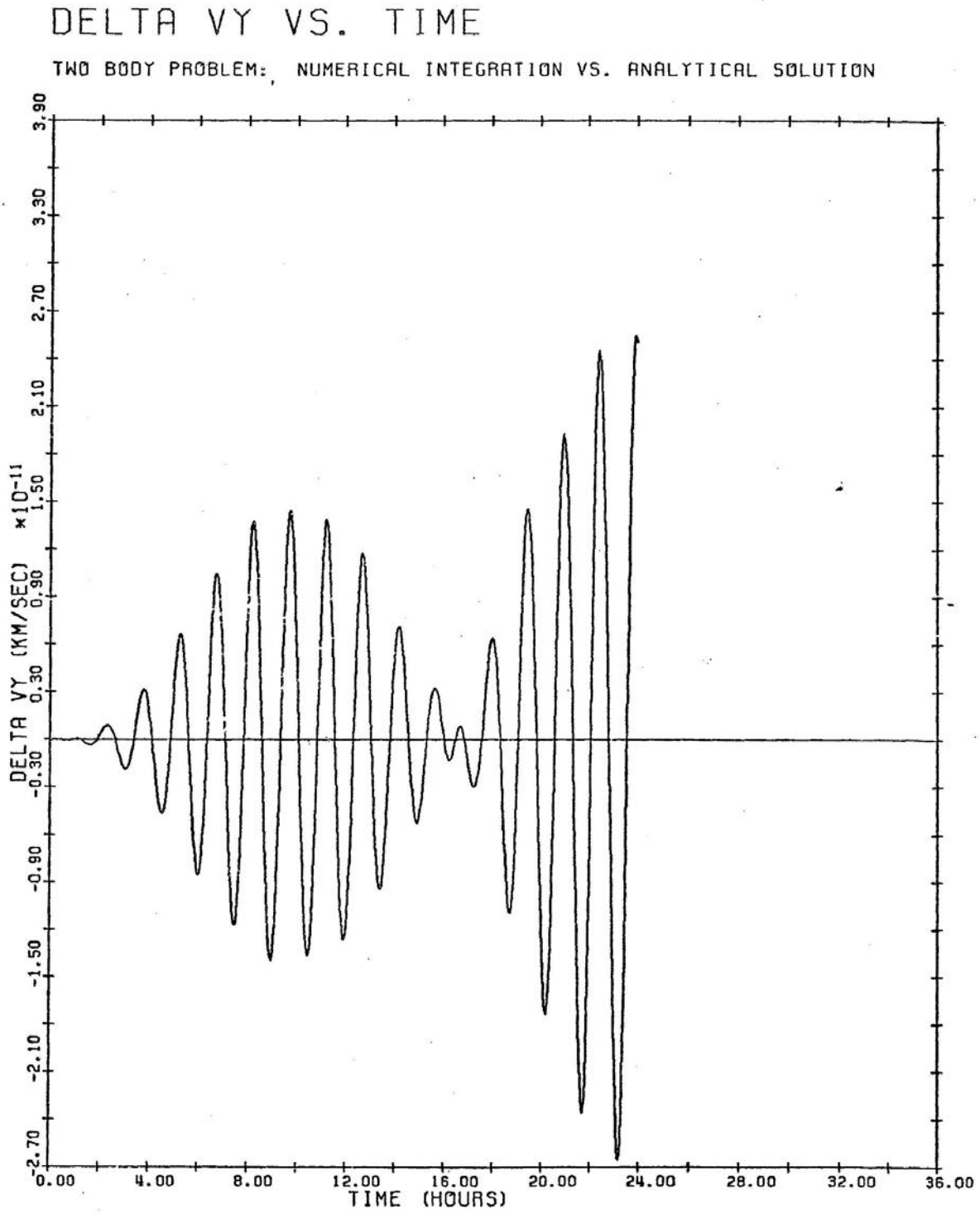


Figure 2.29. Difference in "y" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)

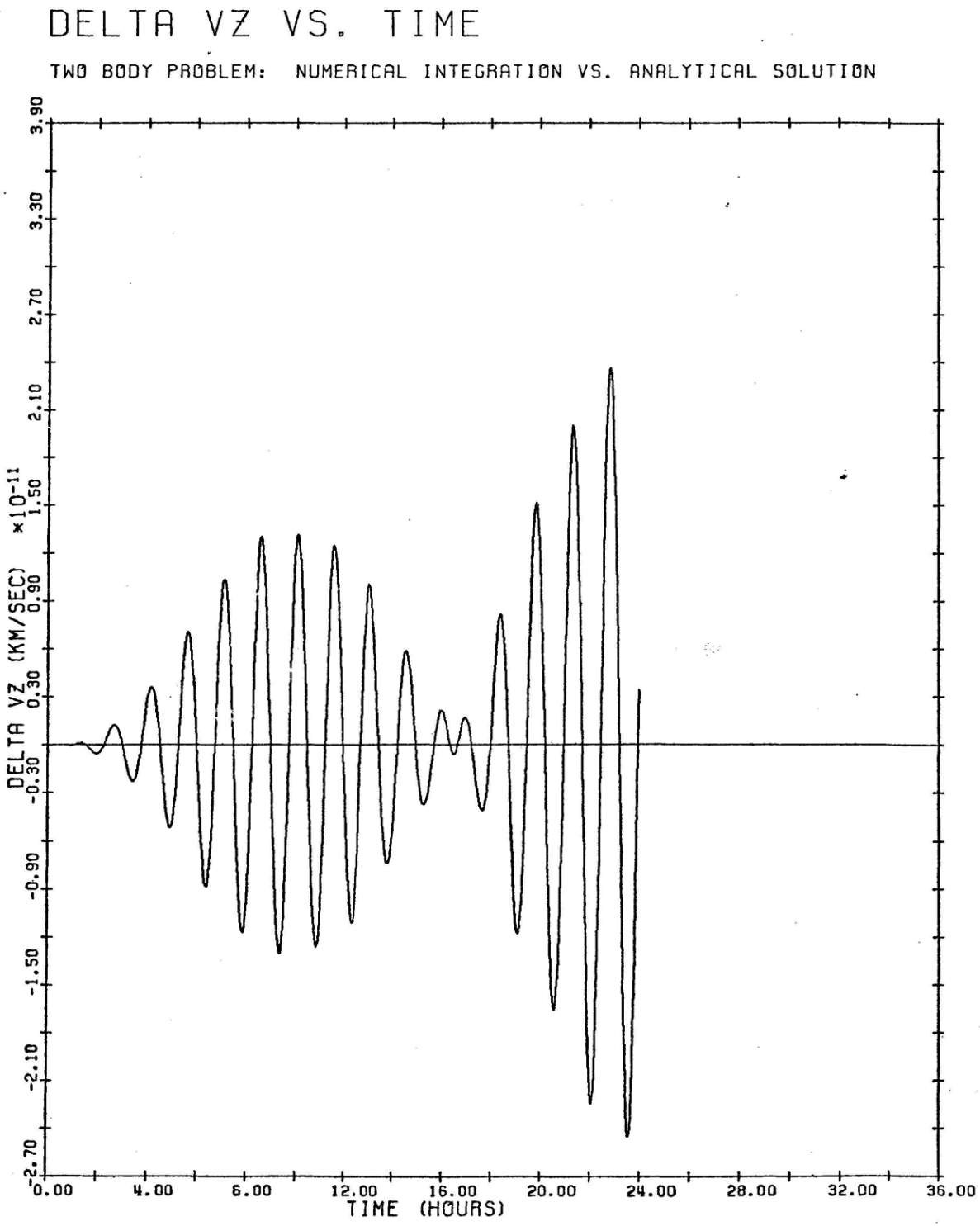


Figure 2.30. Difference in "z" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)

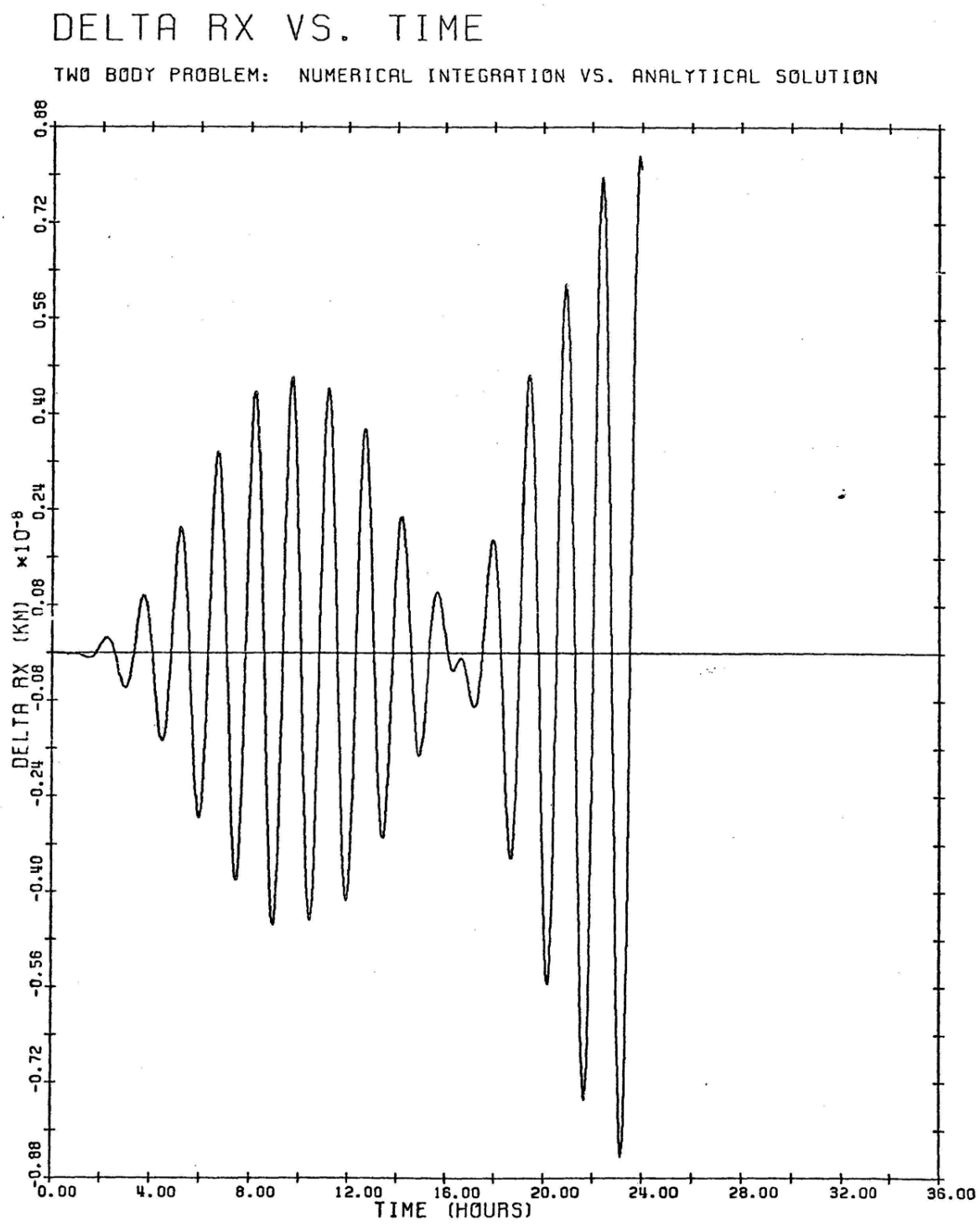


Figure 2.31. Difference in "x" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)

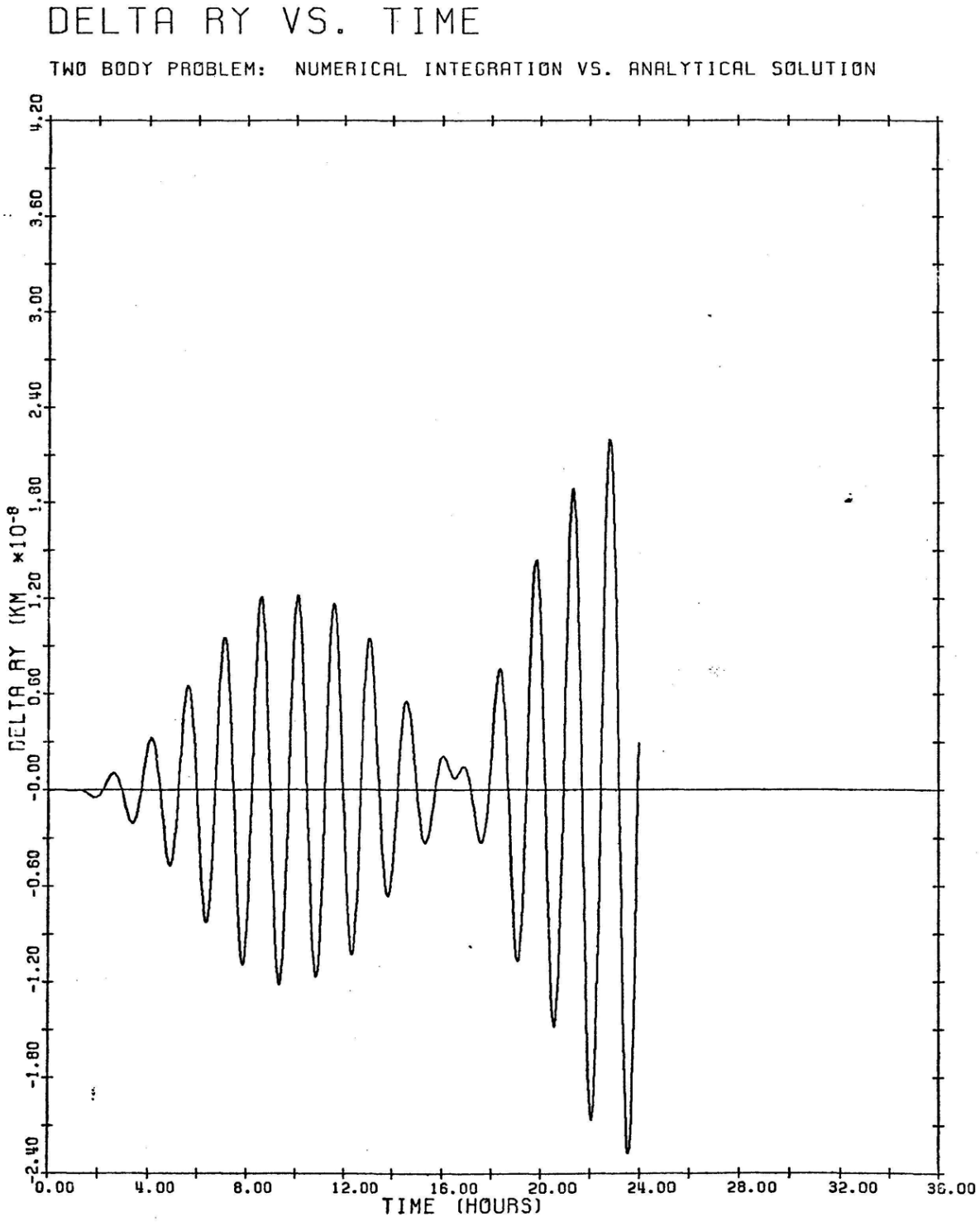


Figure 2.32. Difference in "y" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)

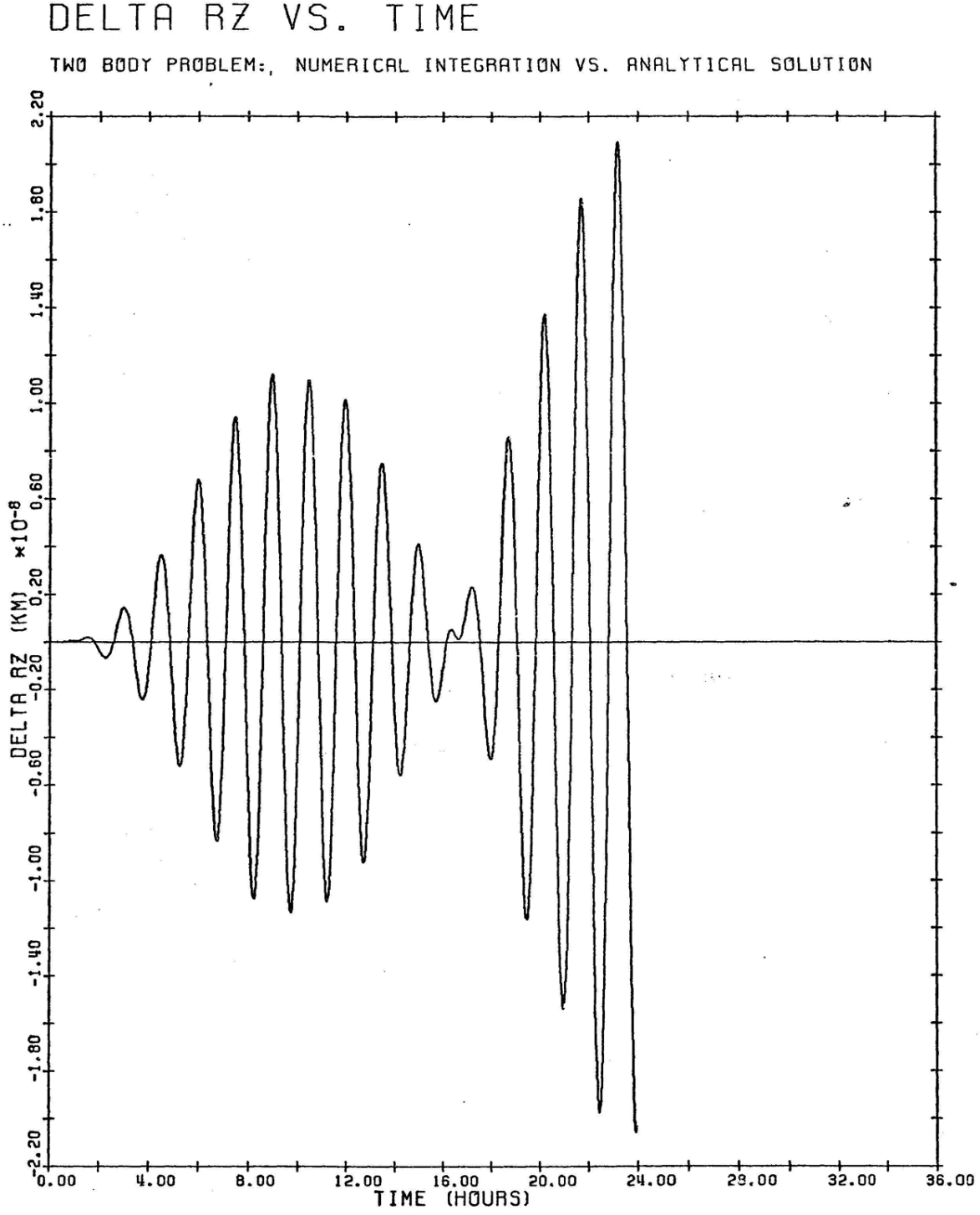


Figure 2.33. Difference in "z" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 2700 sec.)

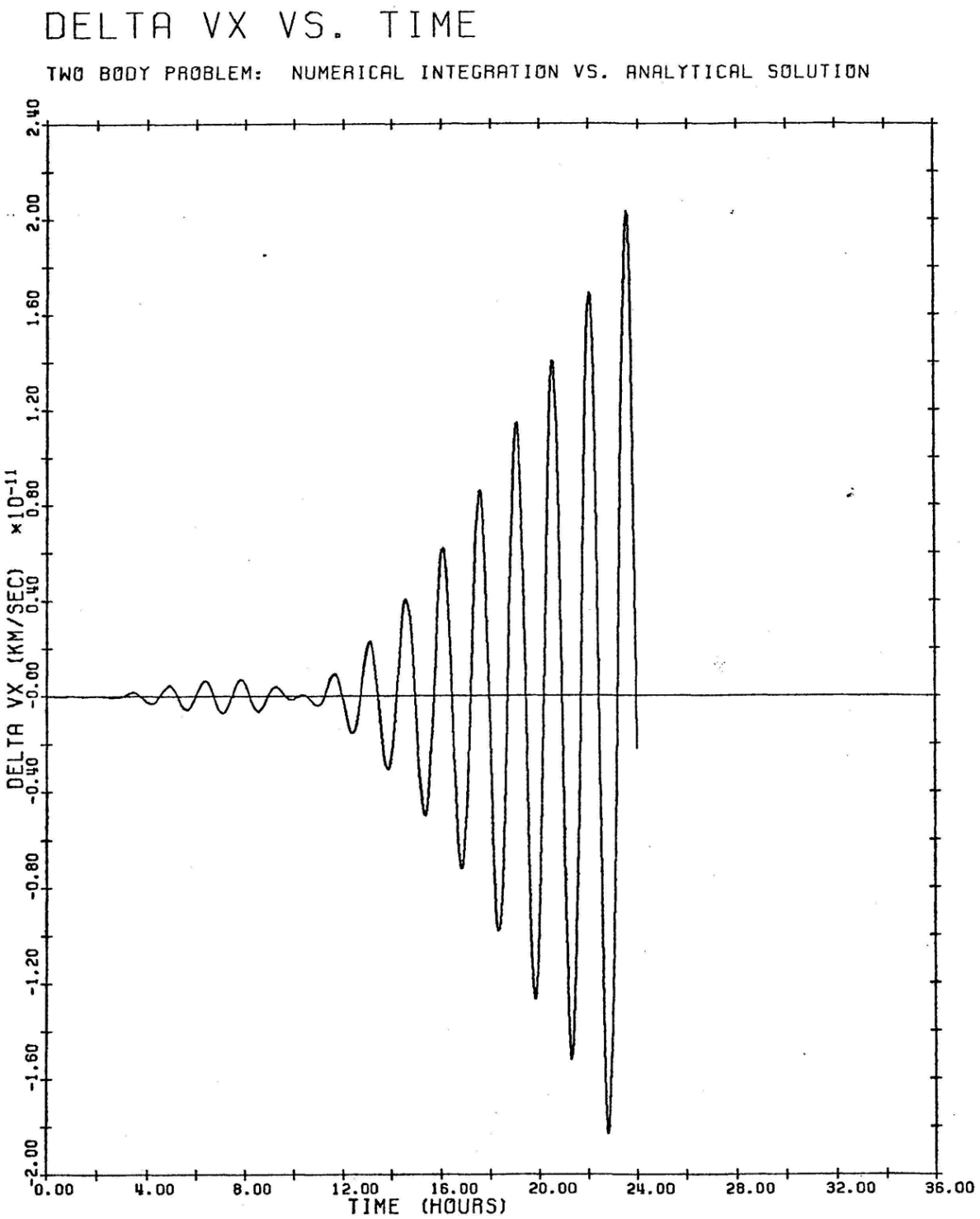


Figure 2.34. Difference in "x" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)

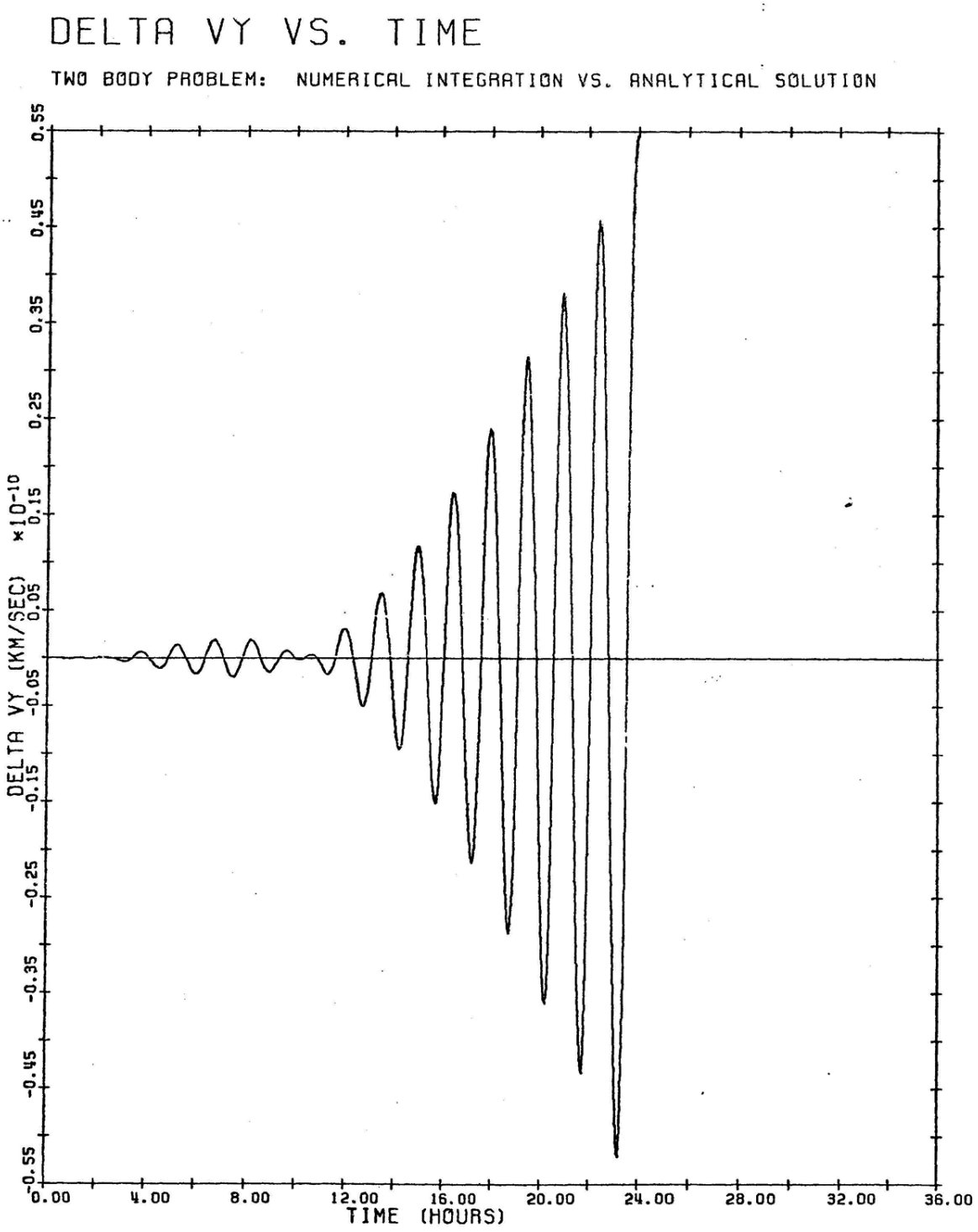


Figure 2.35. Difference in "y" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)

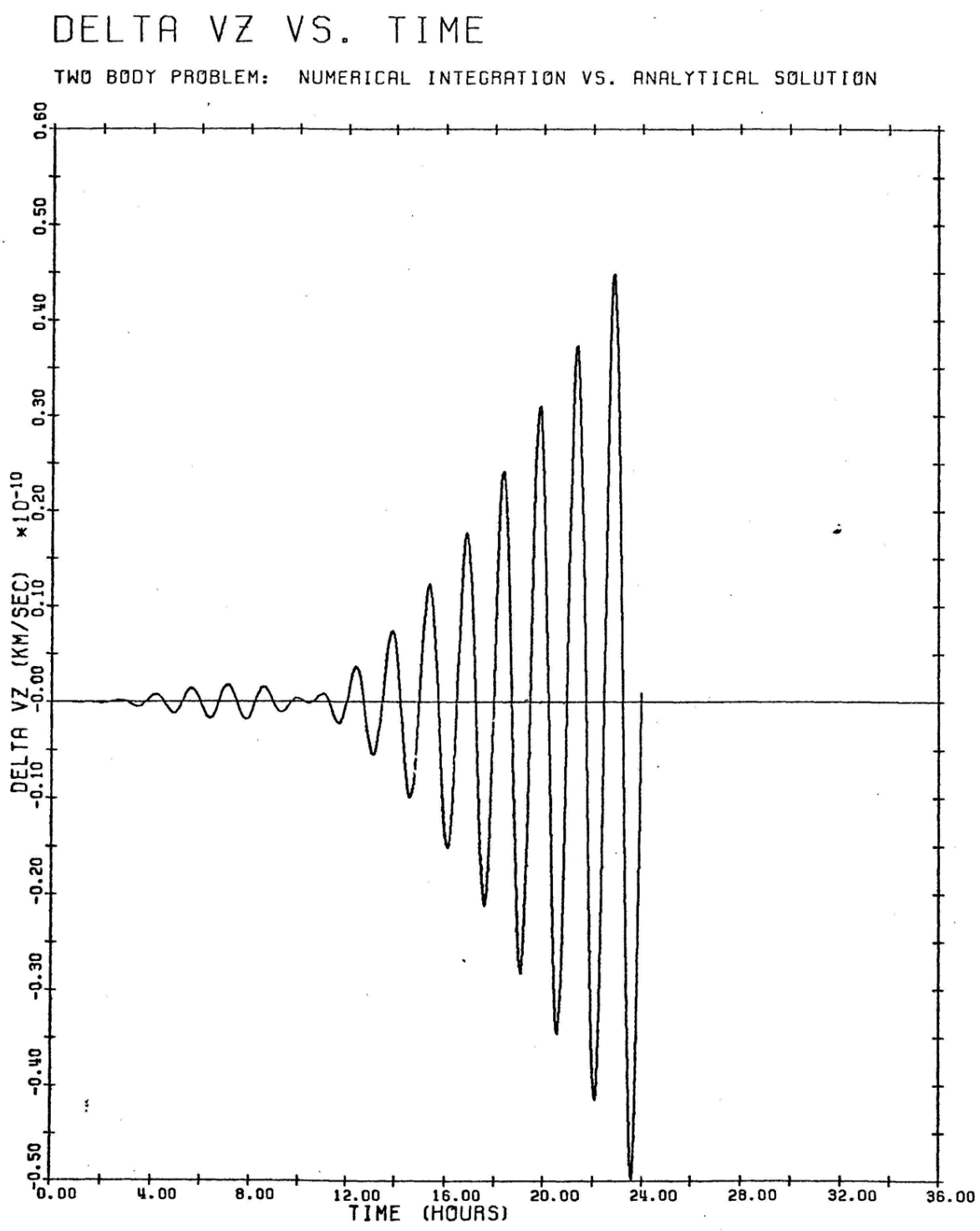


Figure 2.36. Difference in "z" Velocity Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)

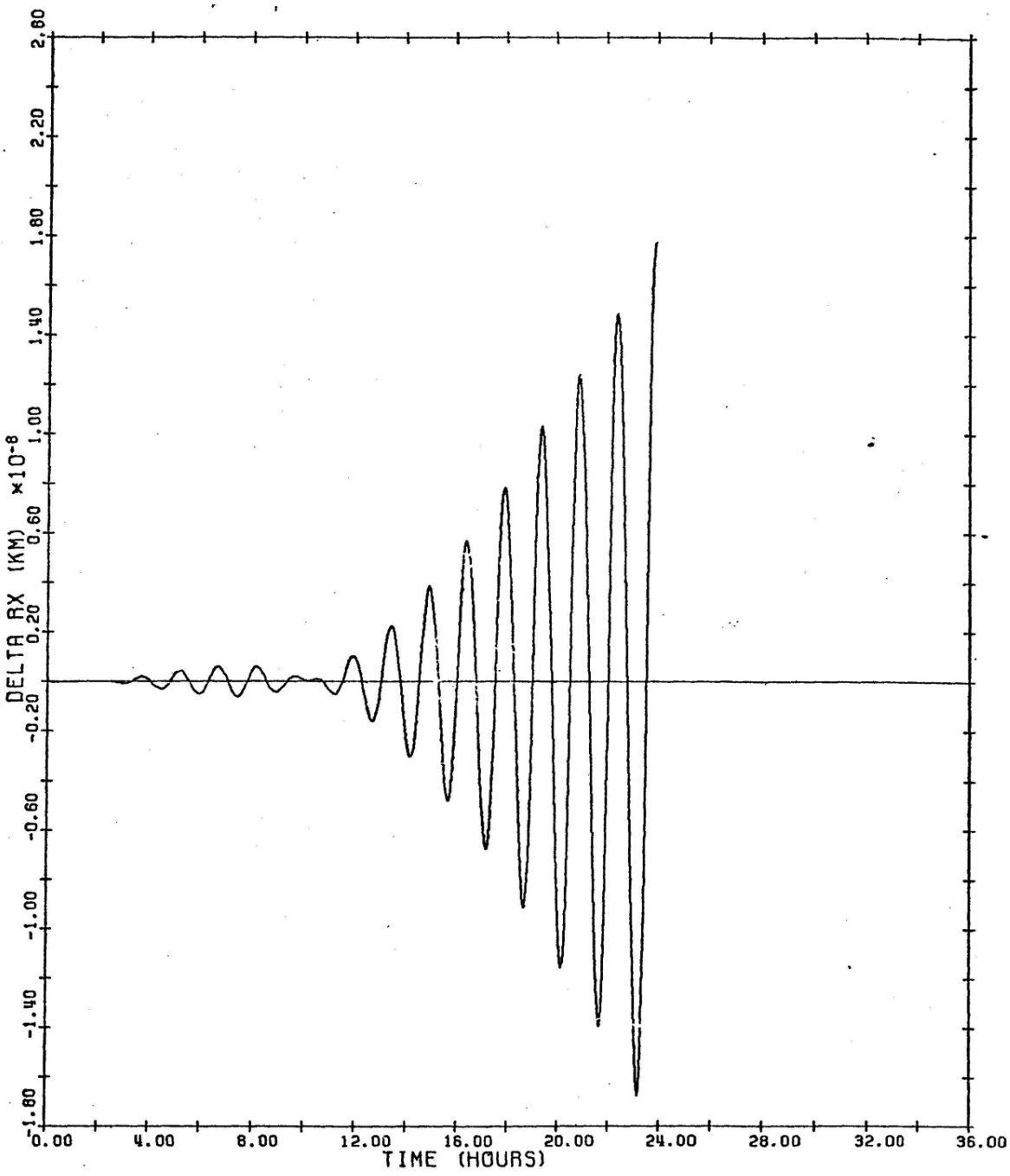


Figure 2.37. Difference in "x" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)

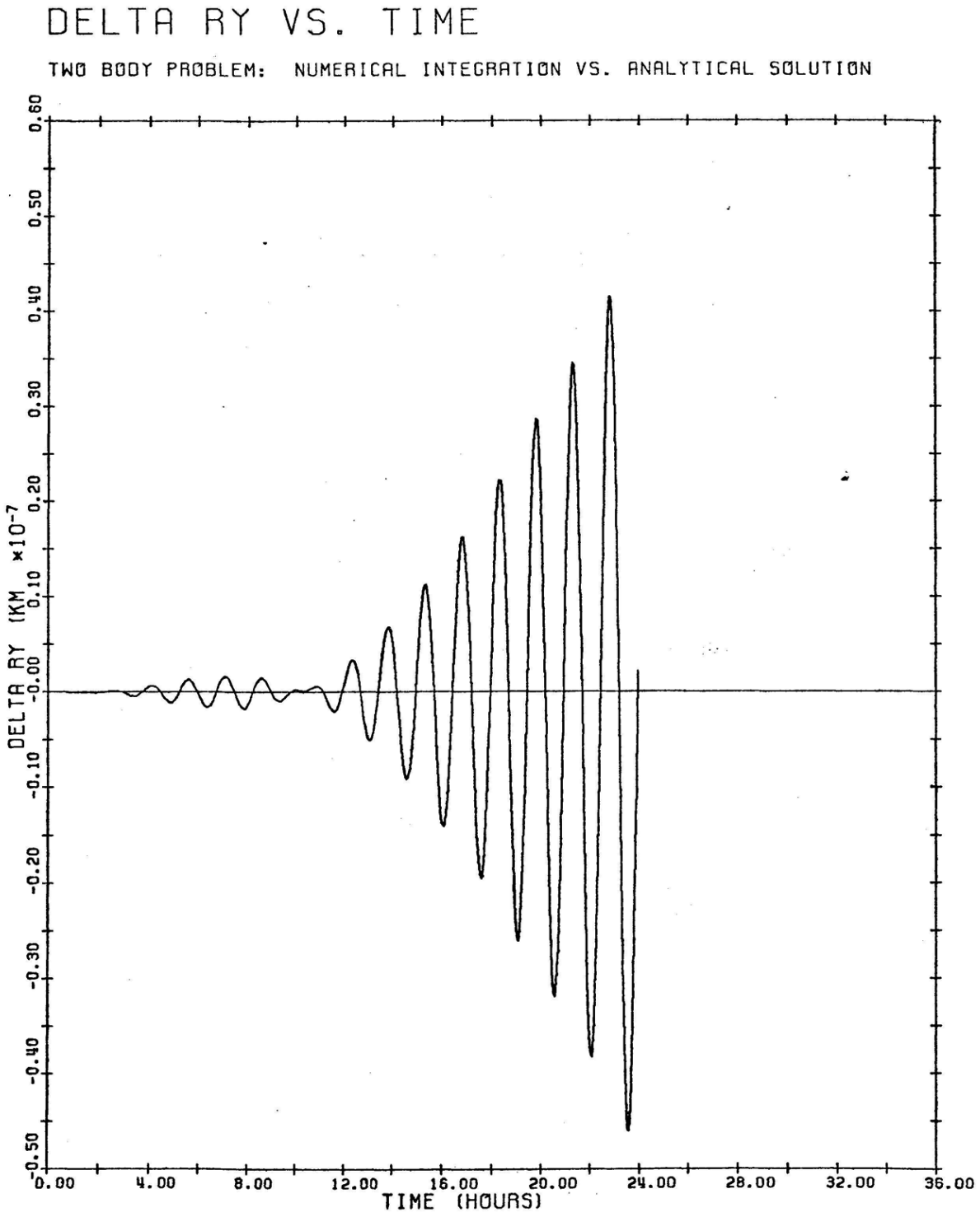


Figure 2.38. Difference in "y" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)

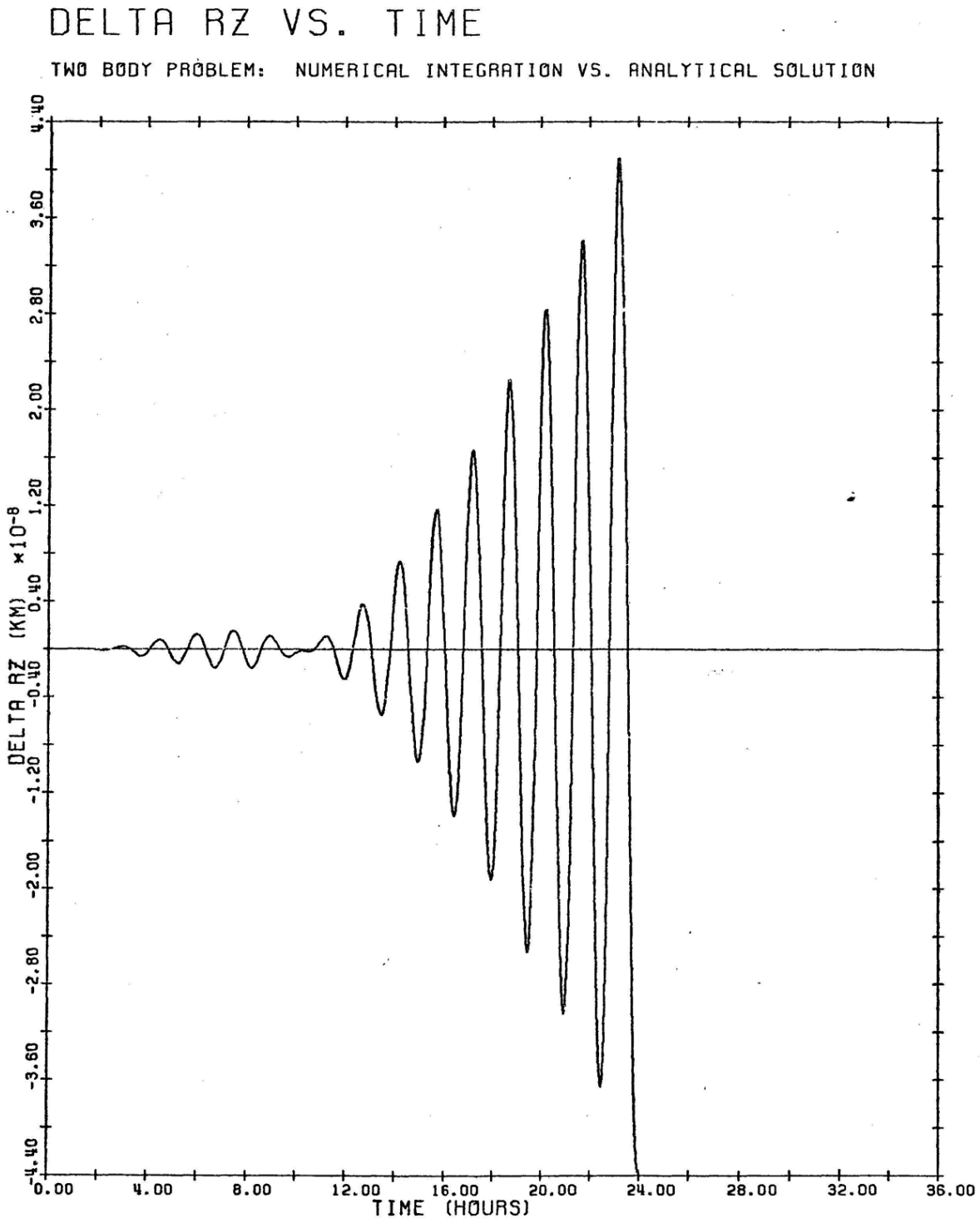


Figure 2.39. Difference in "z" Position Component: Picard-Chebyshev Numerical Integration versus Analytical Solution (Integration Stepsize = 1350 sec.)

appear to be nearly one order of magnitude less than the Cowell errors during the compare span.

It should be noted that further reduction in the integration stepsize does not significantly reduce the magnitude of maximum differences over the 24 hour compare span. For example, a Picard-Chebyshev integration run using a stepsize of one-sixteenth of a satellite period yielded maximum differences in the position and velocity components that were of the same order of magnitude as those produced by the run using an integration arc of one fourth the satellite period. Empirical results such as those given above indicate that the Picard-Chebyshev stepsize should not generally be reduced beyond one-fourth of a satellite period. It must also be remembered that reducing the length of the integration arc penalizes the algorithm performance and diminishes the advantage gained through the parallel architecture. This performance characteristic will be further described in Chapter 4.

2.4.3 Low Altitude Circular Test Results

The epoch conditions for the low altitude circular test case are given in Table 2.3 and descriptions of the various test runs for this test case are presented in

Table 2.5. For the high precision formulations, the GTDS-RD version of the Picard-Chebyshev integrator was used to propagate the equations of motion and this integrated trajectory was compared to a baseline trajectory provided by the GTDS-RD high precision Cowell integrator with a 30 second stepsize. In the case of the averaged equations of motion, the Picard-Chebyshev integration was compared against the GTDS-RD multistep integration of the averaged equations using a one day stepsize. The high precision trajectories were compared over two orbital periods, while the averaged trajectories were compared over much longer spans.

Table 2.7 presents a summary of the numerical results from these trajectory comparisons. Each row in the table gives the maximum sampled radial, cross track, along track and delta-r errors computed for a particular test run.

In test run 1, the Cowell equations were integrated by the Picard-Chebyshev algorithm using a 5400 second (1 orbital period) integration stepsize. Zonal harmonics J_2 through J_8 were included in the force model, but drag effects were not included. Table 2.7 shows that the Picard-Chebyshev integrator agreed extremely well with the precise

Table 2.7. Order of the Maximum Sampled† Position Differences for the Low Altitude Circular Test Cases

RUN #	RADIAL	CROSS TRACK	ALONG TRACK	DELTA-R
1	10^{-4} mm	10^{-4} mm	10^{-3} mm	10^{-3} mm
2	1 cm	10^{-1} mm	10 cm	10 cm
3	1 cm	10^{-1} mm	1 cm	1 cm
4	1 cm	10^{-1} mm	10 cm	10 cm
5	1 cm	10^{-1} mm	1 cm	1 cm
6	1 cm	10^{-1} mm	1 cm	1 cm
7	1 m	10 cm	1 m	1 m
8	10^{-2} mm	10^{-1} mm	10^{-1} mm	10^{-1} mm
9	10^{-2} mm	1 mm	1 mm	1 mm

† Sampling Rates: runs 1-6 = 1 sample/min

run 7 = 1 sample/2 min

run 8 = 1 sample/8 min

run 9 = 1 sample/2 days

Cowell integration for this case. When drag is included in the force model, however, the results are somewhat different. Test run 2 repeats the conditions of test run 1, but with drag included. Table 2.7 indicates that the dominant error components (along track and delta-r) increased four orders of magnitude over their values in the drag-free case. Although this is a rather dramatic change, the actual along track difference between the Cowell and Picard-Chebyshev integrations after three hours for the drag case is only 10 cm. Reduction of the stepsize by a factor of two produces an order of magnitude reduction in the compare differences, as seen in Table 2.7 for test run 3.

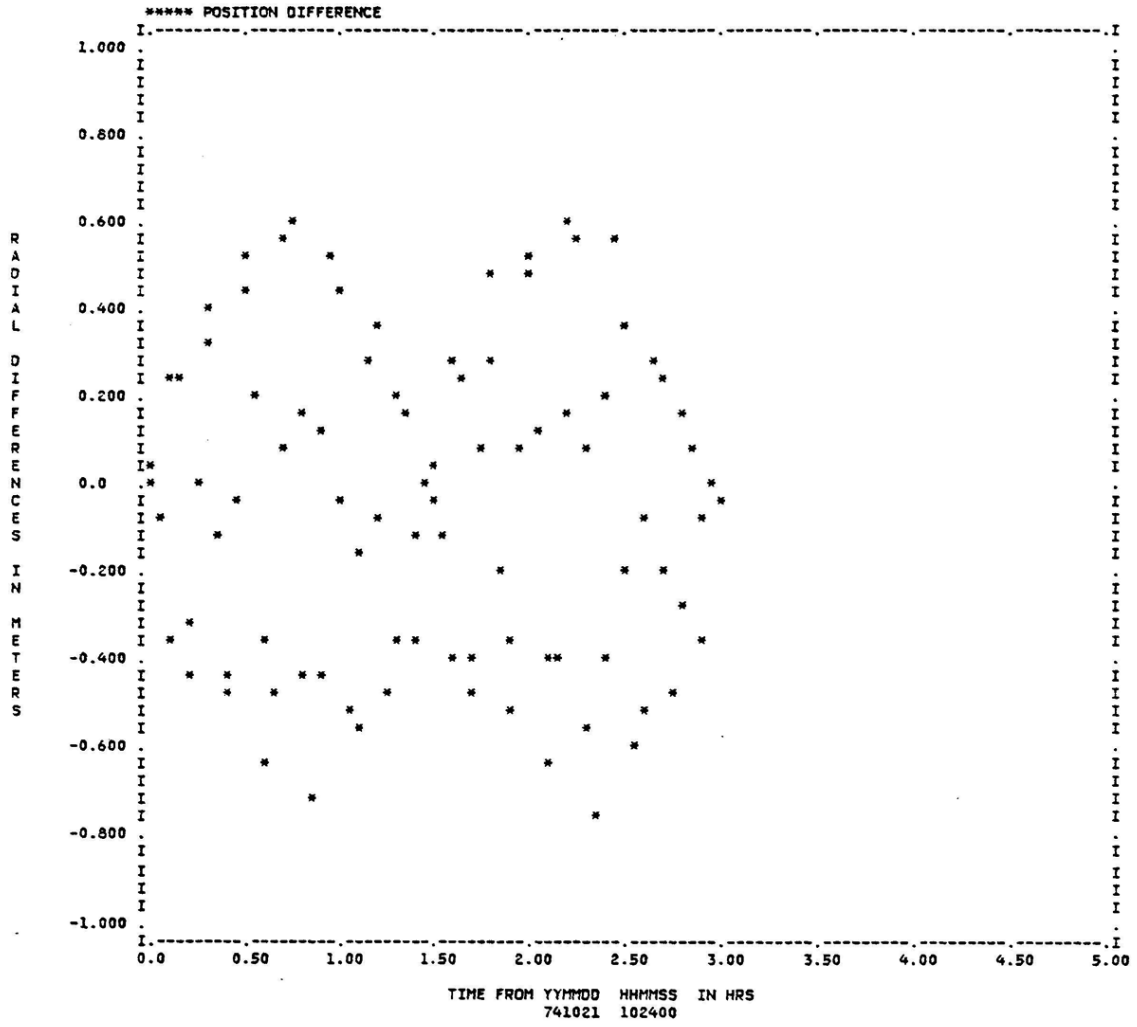
The first three test runs indicated that the Picard-Chebyshev algorithm provided very accurate integrations of the Cowell equations of motion for the low altitude circular test case, even in the presence of the drag perturbation. It was also desired to evaluate the Picard-Chebyshev integration capability for the Variation of Parameters formulation of the equations of motion. In test run 4, the Picard-Chebyshev integrator was used with the equinoctial VOP equations with mean longitude as the fast variable. The remaining conditions for this run were the same as those for test run 2. The test run 4 comparison results given in Table 2.7 show that the Picard-Chebyshev integra-

tion of this VOP formulation gives results as accurate as the previous integration of the Cowell equations. In test run 5, the integration stepsize was cut in half and the accuracy results are very comparable to the Cowell results of test run 3. An additional run using the conditions of test run 5, but integrating the equinoctial VOP equations with eccentric longitude as the fast variable, gave exactly the same accuracy results as test runs 3 and 5. However, it should be noted from Table 2.5 that these comparable accuracies were achieved with a significant reduction in the order of the Chebyshev polynomials approximating the integrands of the equations of motion. Additionally, the Picard-Chebyshev algorithm converged in significantly fewer iterations for the VOP formulations than it did for the Cowell formulation. More will be said about this characteristic in Chapter 4.

A final test run for the VOP equations was made to examine the effect on accuracy of reducing both the order of the Chebyshev approximating series and the number of Chebyshev orthogonality points. Test run 7 used Chebyshev approximating series of order 22 and 50 orthogonality points in the integration of the equinoctial VOP equations (mean longitude). The integration stepsize was 1 satellite orbital period. Table 2.7 shows that, after two satellite

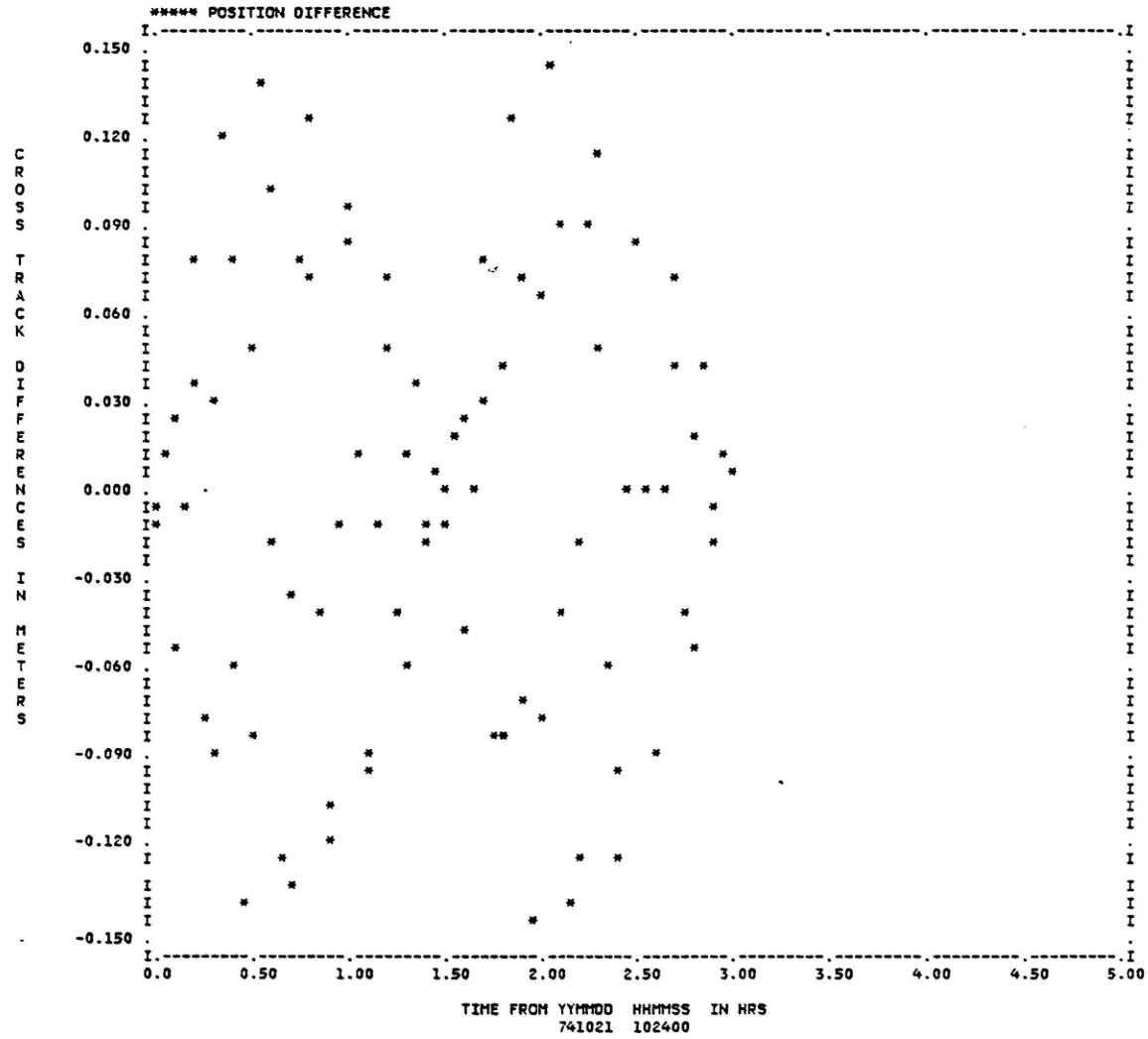
orbital periods, the Picard-Chebyshev and the Cowell integrations differ by about 1-2 meters in radial, along track and delta-r differences. Figures 2.40 through 2.42 present plots of the radial, cross track and along track position differences over the three hour compare span for test run 7. It can be seen that the differences remain relatively well-behaved, with only the along track difference indicating any significant secular growth. This is generally attributable to the mismatch between the mean motion computed from the Cowell reference trajectory and that computed from the Picard-Chebyshev integration. Figures 2.43 through 2.45 depict the same difference information for the velocity. These difference plots show the same behavior as the position differences, except that the radial velocity difference contains the dominant secular growth component due to the mean motion difference. Additionally, Figures 2.46 through 2.51 plot the histories of the equinoctial elements over the three hour compare span. These are the integration state components being integrated by the Picard-Chebyshev integrator in this test run. The elements from the Cowell integration are plotted with an asterisk symbol (*) while those elements from the Picard-Chebyshev integration are plotted with a plus symbol (+). When the two data items to be plotted at any output time agree to within the resolution of the printer-

Figure 2.40. Radial Position Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)



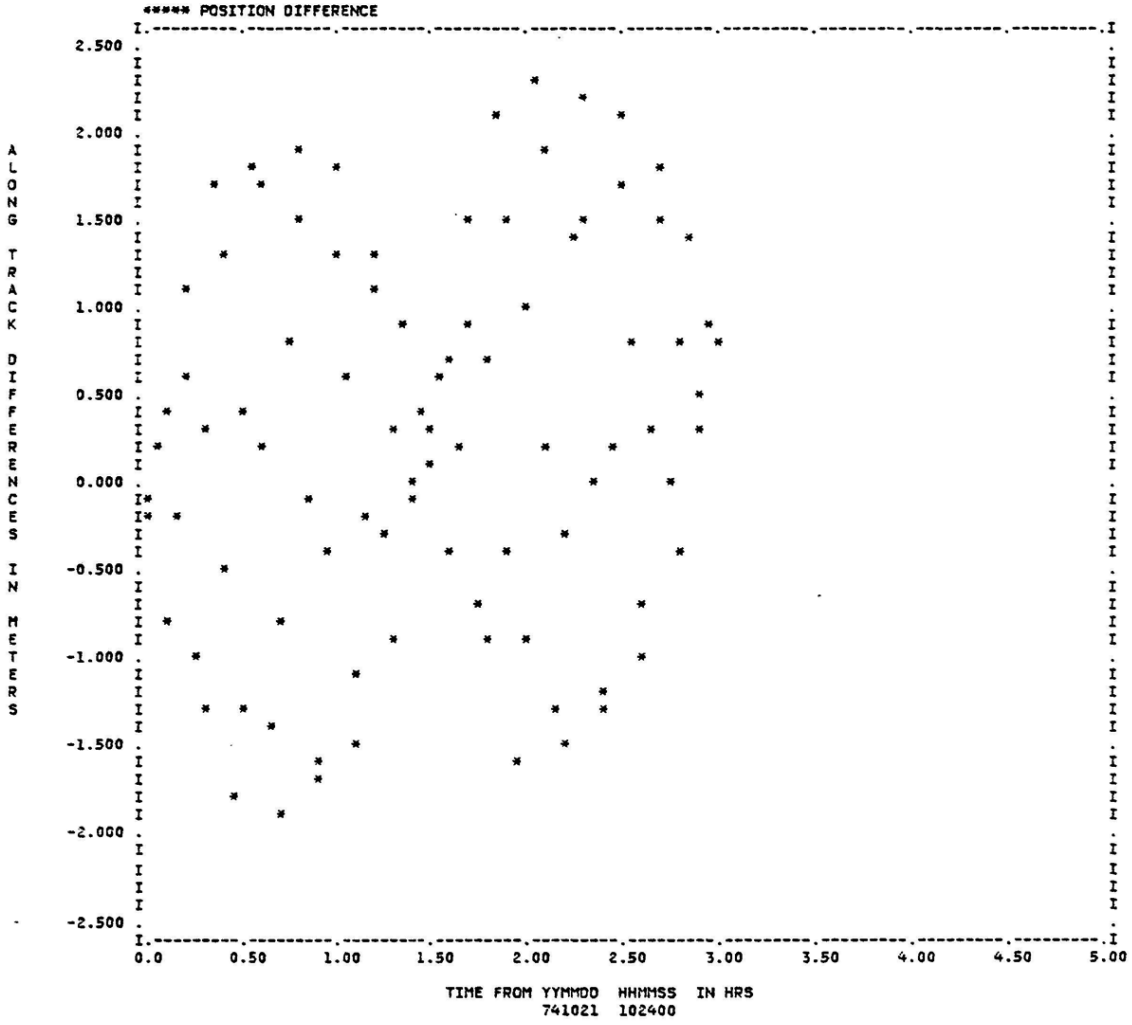
	Picard-Chebyshev (M=22, N=49)	Cowell
Stepsize:	5400 sec	30 sec
Field:	8x0	8x0
Drag:	yes	yes
Formulation:	Equinoctial VOP (λ)	Cowell

Figure 2.41. Cross Track Position Differences for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)



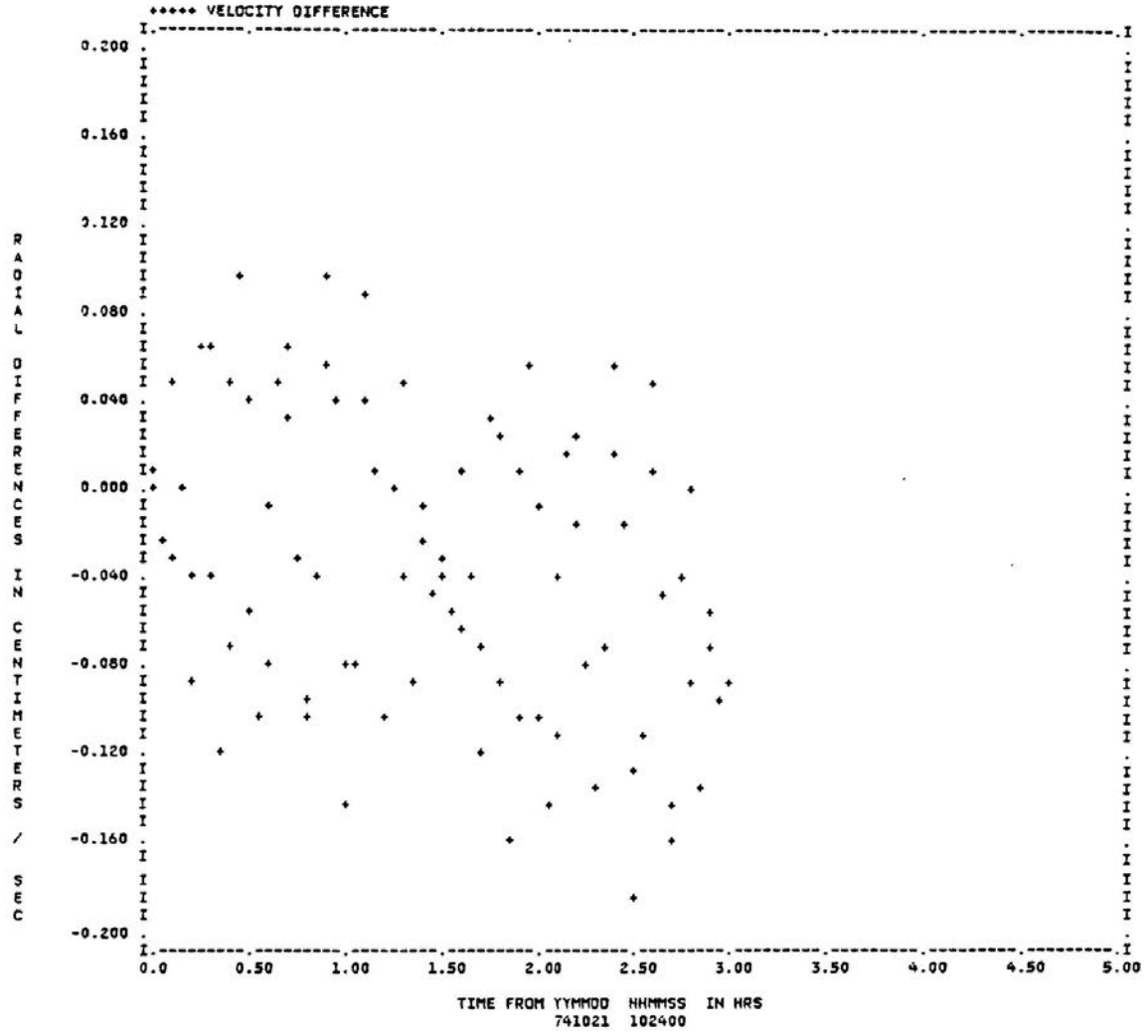
See Figure 2.40 for detailed integrator description.

Figure 2.42. Along Track Position Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)



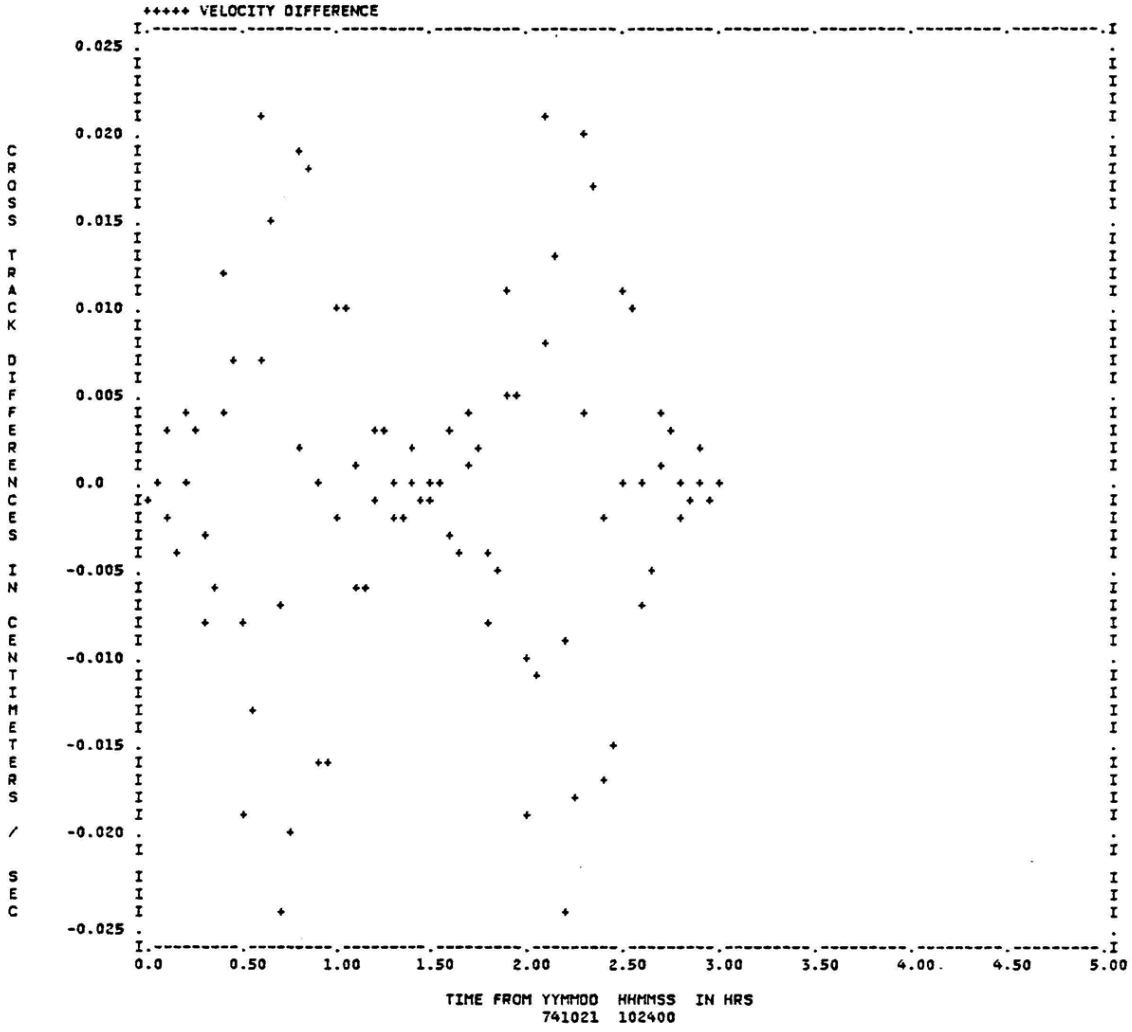
See Figure 2.40 for detailed integrator description.

Figure 2.43. Radial Velocity Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)



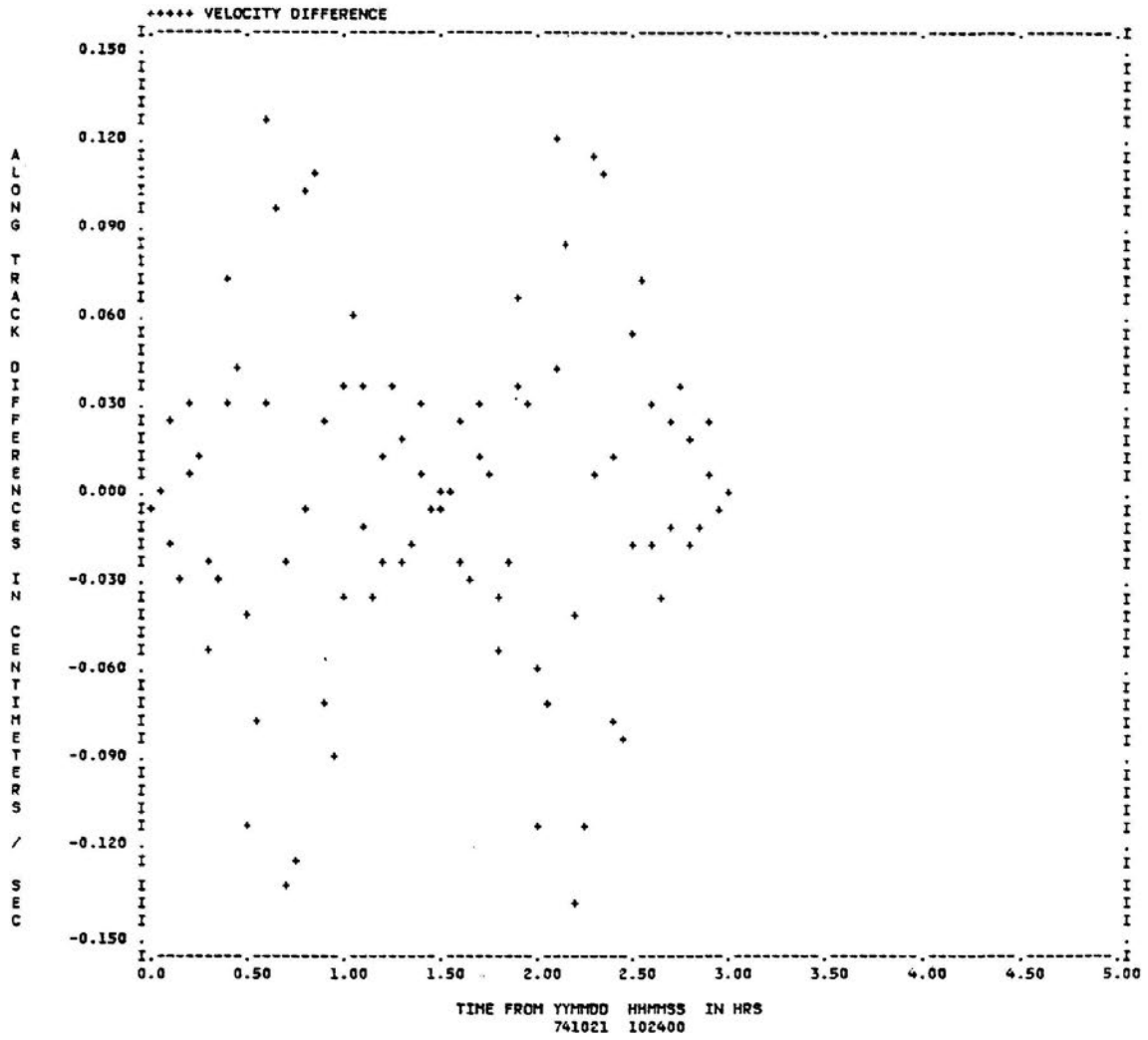
See Figure 2.40 for detailed integrator description.

Figure 2.44. Cross Track Velocity Difference for Low Altitude Circular Test Case: Picard-Chebyshev Integration versus Cowell Integration (Test Run 7)



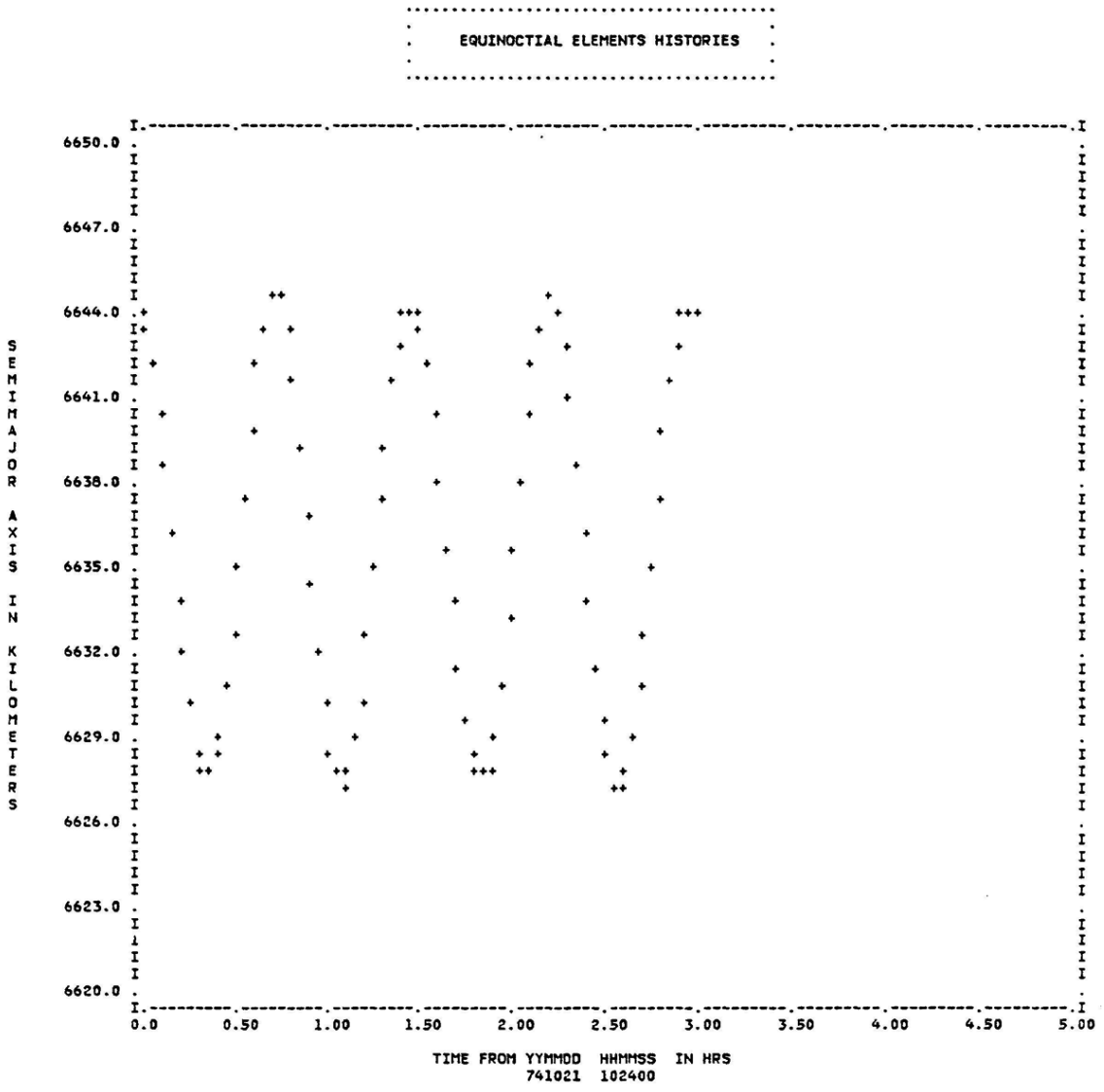
See Figure 2.40 for detailed integrator description.

Figure 2.45. Along Track Velocity Difference for Low Altitude Circular Test Case: Picard Chebyshev Integration versus Cowell Integration (Test Run 7)



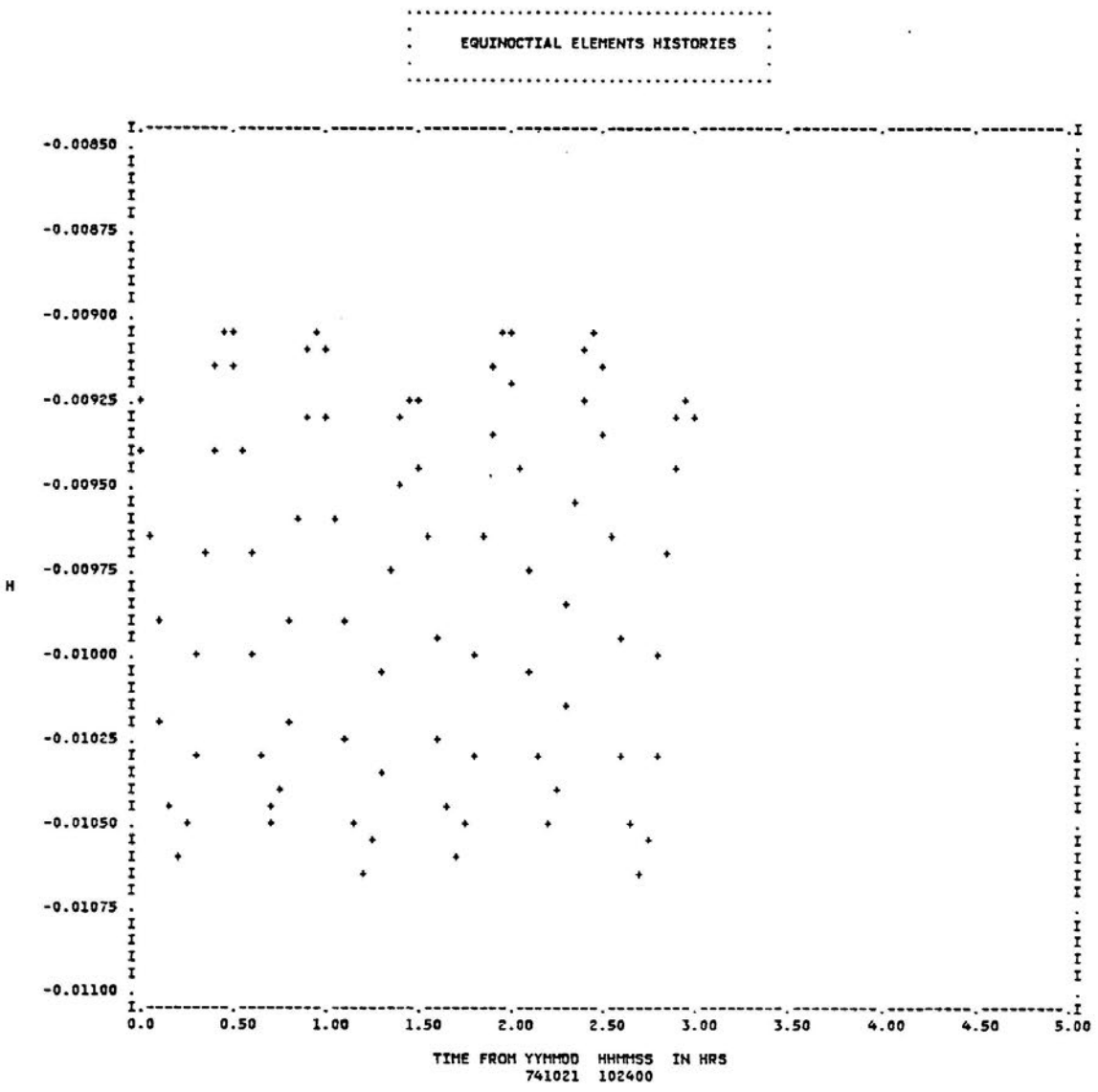
See Figure 2.40 for detailed integrator description.

Figure 2.46. Semimajor Axis Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)



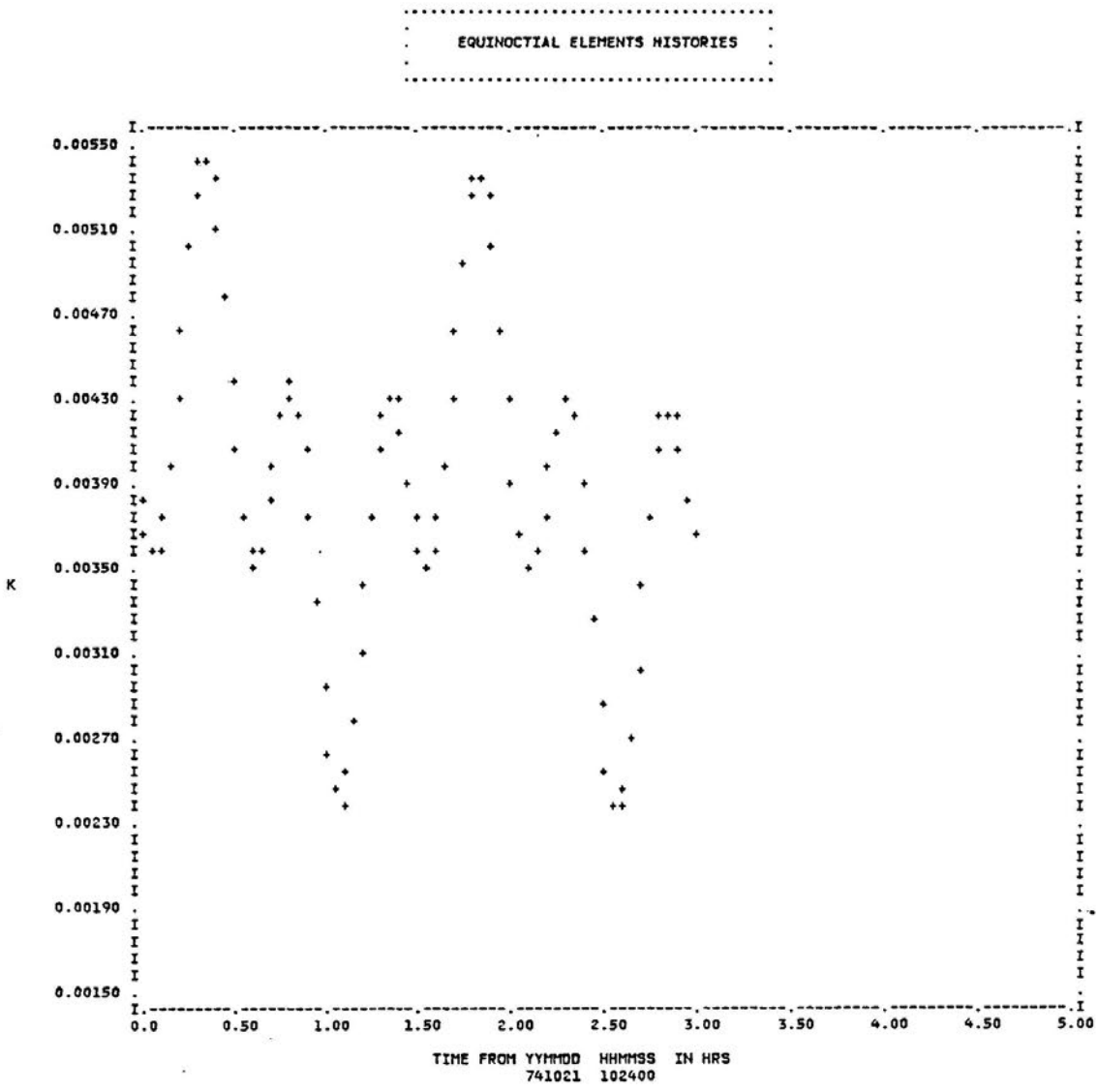
	Picard-Chebyshev (M=22, N=49)	Cowell
Stepsize:	5400 sec	30 sec
Field:	8x0	8x0
Drag:	yes	yes
Formulation:	Equinoctial VOP (λ)	Cowell

Figure 2.47. Equinoctial Element "h" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)



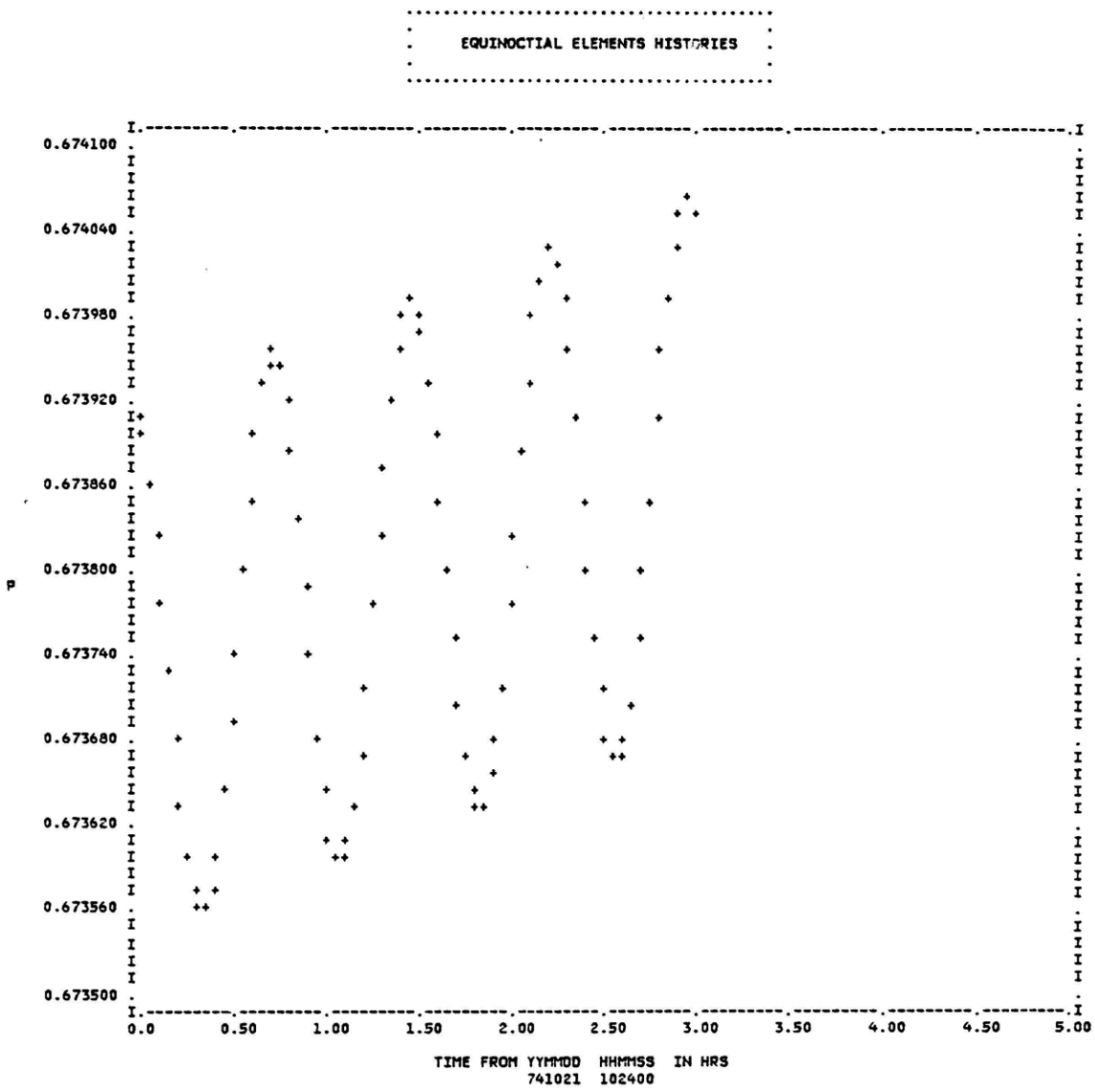
See Figure 2.46 for detailed integrator description.

Figure 2.48. Equinoctial Element "k" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)



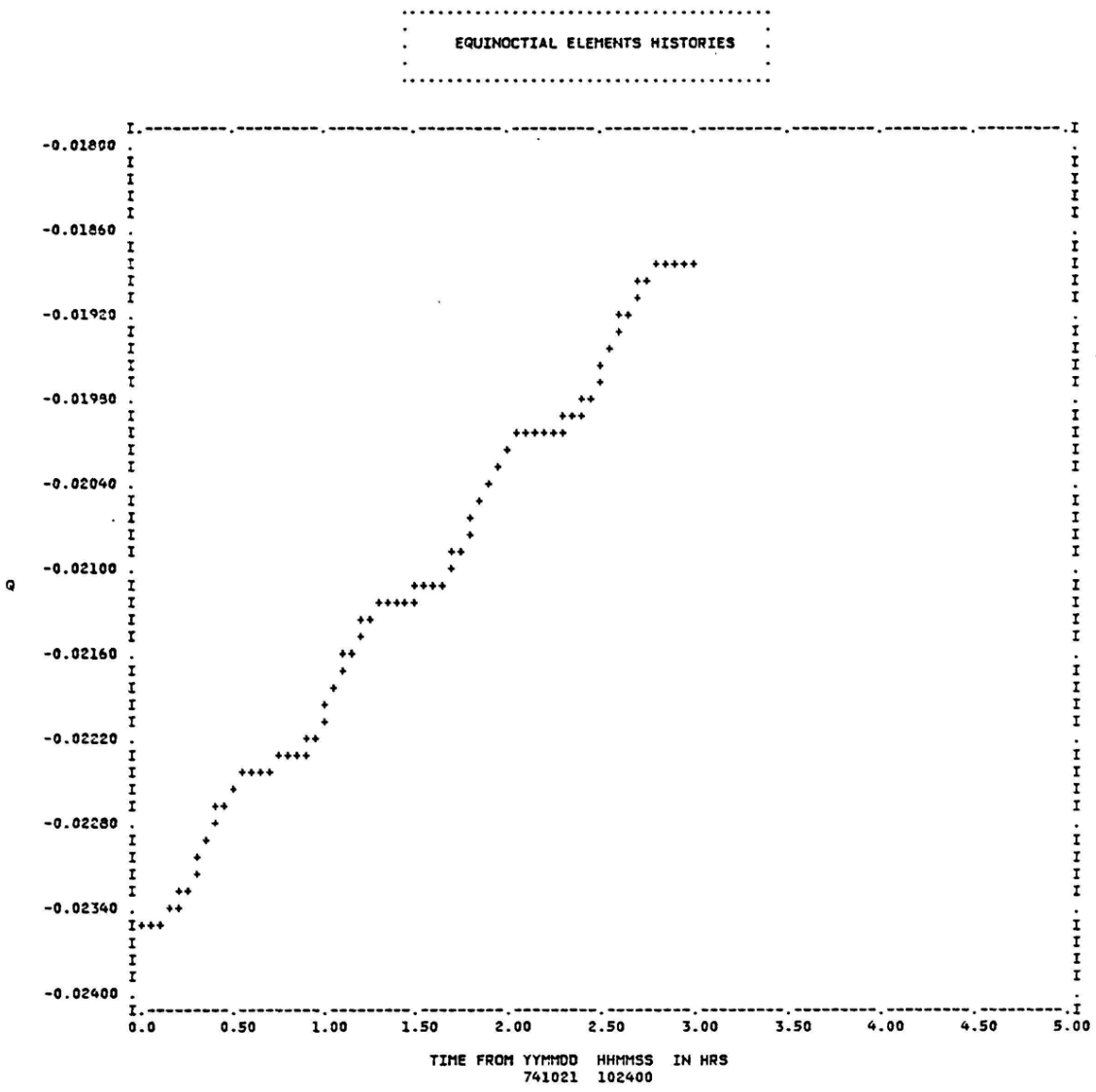
See Figure 2.46 for detailed integrator description.

Figure 2.49. Equinoctial Element "p" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)



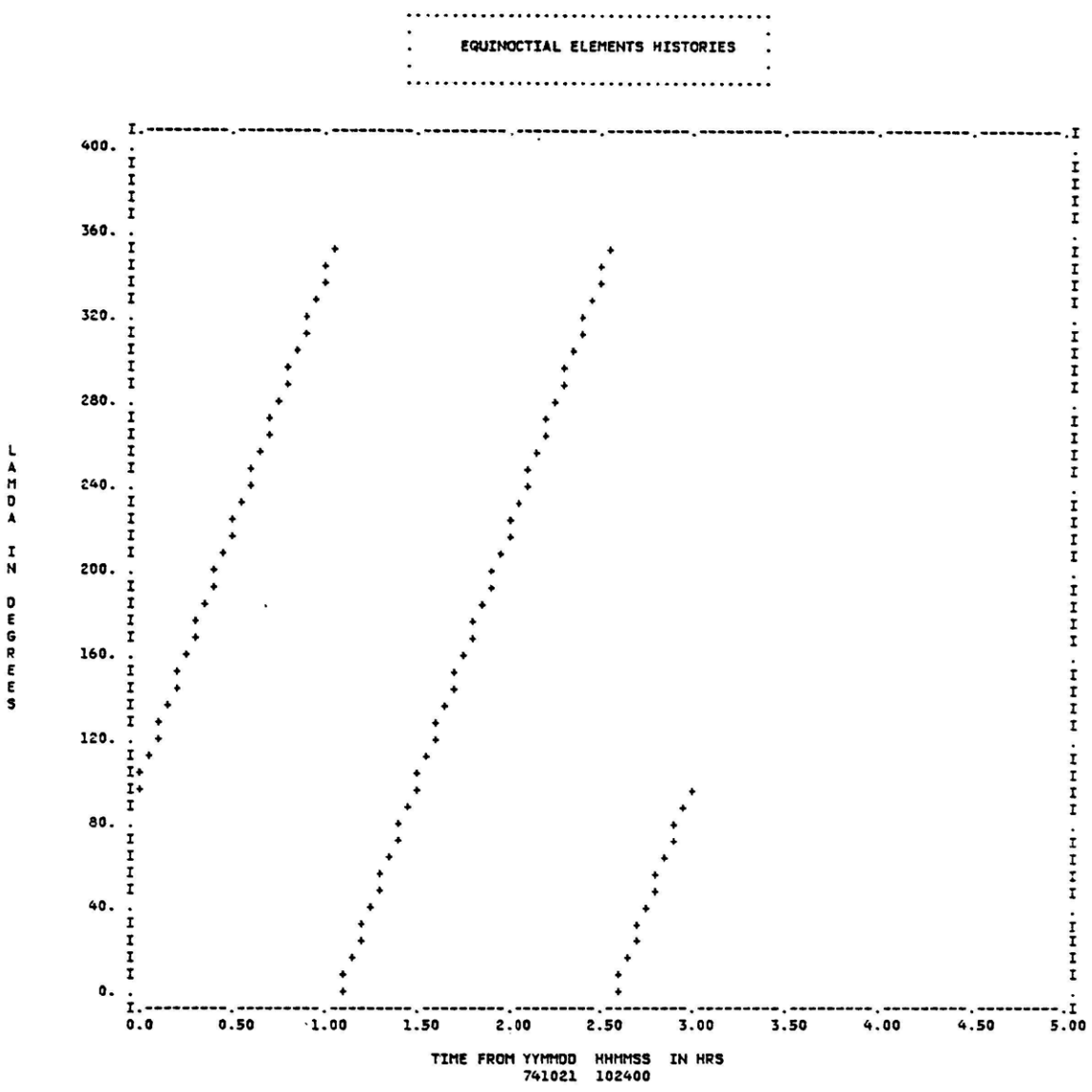
See Figure 2.46 for detailed integrator description.

Figure 2.50. Equinoctial Element "q" Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)



See Figure 2.46 for detailed integrator description.

Figure 2.51. Mean Longitude Comparison for 3 Hour Compare Span for Low Altitude Circular Test Case (Test Run 7)



See Figure 2.46 for detailed integrator description.

plotter, then the plus sign (+) will override the asterisk (*). These figures show that the Picard-Chebyshev integration algorithm produces orbital elements that agree very favorably with those produced by the high precision Cowell integration. It is important to note that the near-circular nature of this test case results in element histories that are very well behaved. For example, the semimajor axis plot shows the predominant zonal and drag short periodic behavior superimposed on a slight secular decrease in the semimajor axis due to the nonconservative drag force. This plot does not reflect any significant perigee "spike" in the element history that might become apparent in orbits with higher eccentricities. The smooth nature of the element rates makes them easy to approximate with low-order Chebyshev polynomial series. Additionally, the low altitude circular nature of the orbit implies that the drag force acts relatively uniformly about the orbit. Therefore, an accurate fit to the perturbed element rates can be achieved using low-order Chebyshev series and with a reduced number of orthogonality points for the required coefficient evaluations.

Two runs were made to verify the capability of the Picard-Chebyshev parallel orbit propagation algorithm to integrate the averaged VOP equations of motion. These

equations are very smooth, since the short periodic components of the motion have been numerically or analytically averaged out. With these frequencies removed it is then possible to take larger integration stepsizes. The GTDS-RD system, for example, uses a nominal integration stepsize of 1 day in the multistep integration of the averaged equations of motion. It was desired in test runs 8 and 9 to see how large an integration stepsize could be taken in the Picard-Chebyshev algorithm for the averaged equations. After some experimentation, it appeared that a stepsize of 40 days was the longest that could be taken and still insure algorithm convergence in a reasonable number of iterations. It is felt that this is due to the difficulty in fitting accurately with a Chebyshev series the secular rate in the mean longitude over many satellite revolutions. For example, a 16 revolution/day satellite, such as the low altitude circular test case given here, traverses 1280π radians in a 40 day integration step. Variables with this type of behavior are somewhat difficult to fit accurately and seem to limit the size of the Picard-Chebyshev integration step. However, as indicated by the results of Table 2.7, a 40 day stepsize produces results which agree extremely well with the GTDS-RD multistep integration of the averaged equations of motion over compare spans of 30 and 180 days.

The above results indicate that the Picard-Chebyshev integrator can produce ephemeris information that is comparable in accuracy to current accepted high precision integration techniques for orbits of low eccentricity. Further testing was required to determine the capability of the algorithm to accurately propagate orbits of higher eccentricity in which the integrands might not be such smoothly varying functions. Therefore, a set of test runs was made for a low altitude elliptical test case.

2.4.4 Low Altitude Elliptical Test Results

The epoch conditions for the low altitude elliptical test case are given in Table 2.4 and the descriptions of the various test runs for this test case are presented in Table 2.6. The numerical results for these test runs are summarized in Table 2.8, which presents the maximum sampled position differences found in comparing the Picard-Chebyshev integration and the time regularized Cowell integration based on the same reference epoch conditions.

In test run 1, only zonal harmonics J_2 through J_8 were included in the force model for the integration. The Cowell equations were integrated with an integration stepsize of 1 satellite orbital period (2.25 hours). The

Table 2.8. Order of the Maximum Sampled† Position Differences for the Low Altitude Elliptical Test Case

RUN #	RADIAL	CROSS TRACK	ALONG TRACK	DELTA-R
1	10^{-1} mm	10^{-2} mm	1 mm	1 mm
2	100 m	10 cm	1 km	1 km
3	10 m	1 cm	100 m	100 m
4	1 m	1 mm	10 m	10 m
5	100 m	10 cm	1 km	1 km
6*	10 m	1 cm	10 m	10 m
7*	1 m	1 mm	1 m	1 m
8	10 cm	1 mm	10 cm	10 cm

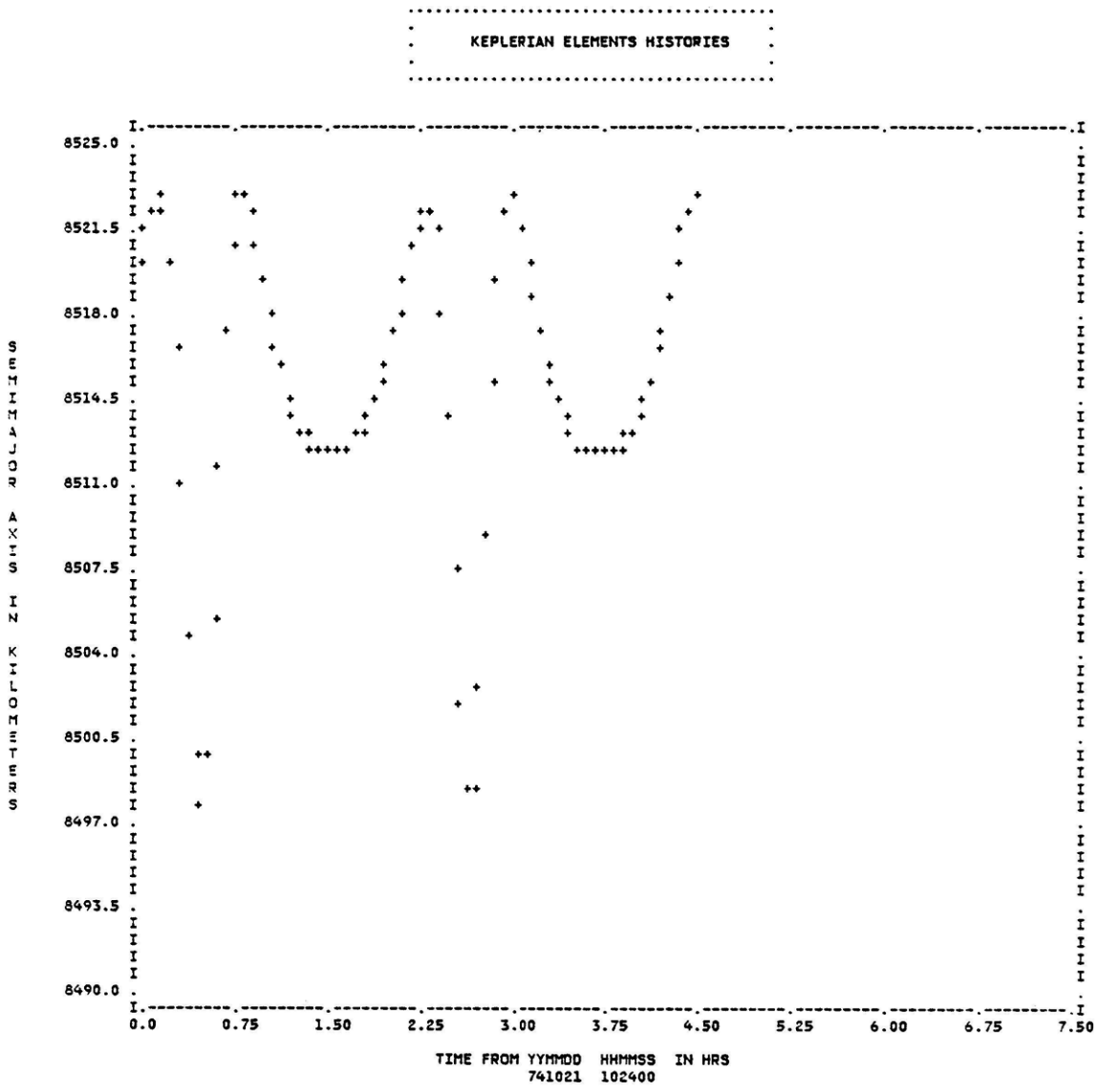
* In these runs, epoch mean anomaly was set to zero degrees.

† Sampling rates: Runs 1-8 = 1 sample/3 min

comparison was taken over two satellite periods, or 4.5 hours. As shown in Table 2.8, the comparison with the 200 step/revolution time regularized Cowell integration is still excellent after 4.5 hours. However, it must be noted that these differences are three to four orders of magnitude greater than those encountered in the corresponding run for the near-circular test case given in Table 2.7. This indicates that the orbital eccentricity can influence the accuracy of the Picard-Chebyshev integration algorithm.

Figures 2.52 through 2.54 plot the time histories of the semimajor axis, eccentricity and inclination for test run 1. Based on the epoch conditions of Table 2.4, the first perigee passage takes place about 0.5 hours from epoch while the second perigee passage takes place about 2.75 hours from epoch. Figures 2.52 through 2.54 indicate that the orbital elements are highly perturbed during the perigee passage. This indicates that the element rates change rapidly in this region. Additionally, this severe perturbation is confined to a relatively small region about perigee, causing a sort of "spike" in the element histories and their associated rates. A good example of how sharp this "spike" can be may be found in Figure 2-42 of Green (82) for a similar eccentric test case.

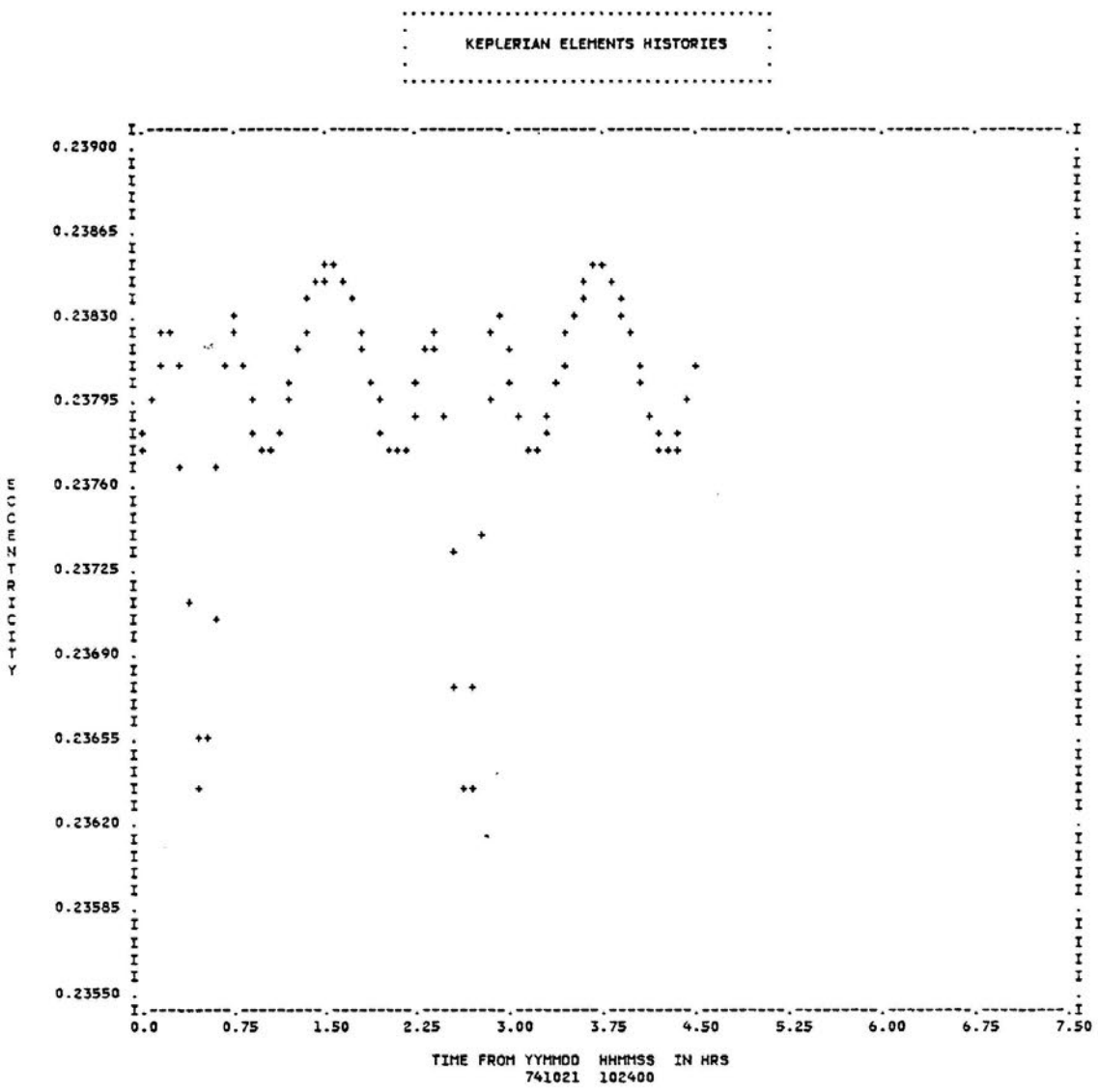
Figure 2.52. Semimajor Axis History for Two Revolutions of the Low Altitude Elliptical Test Case



Epoch conditions: Table 2.4

Force Model: J_2-J_8, no drag

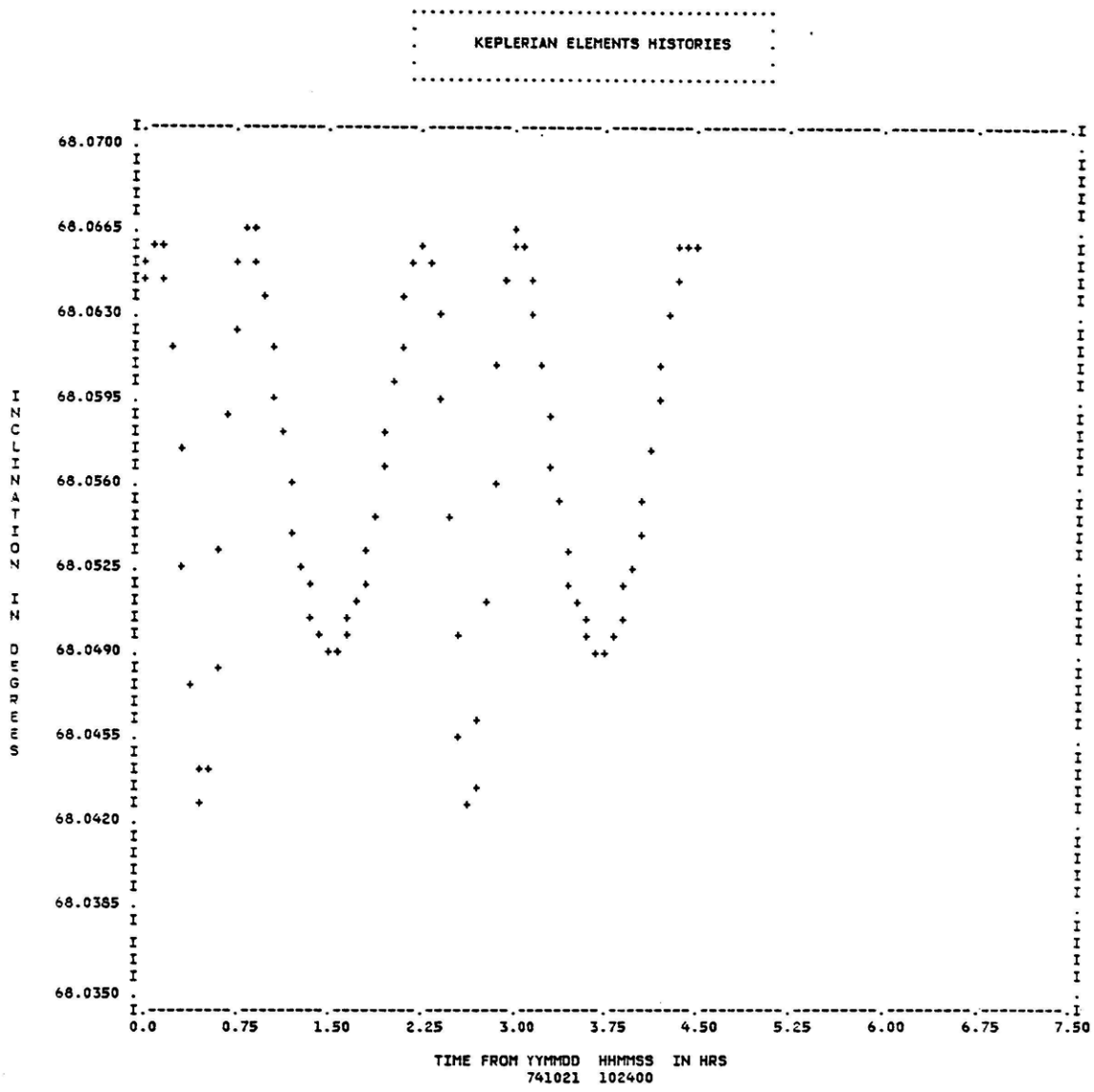
Figure 2.53. Eccentricity History for Two Revolutions of the Low Altitude Elliptical Test Case



Epoch Conditions: Table 2.4

Force Model: J_2-J_8, no drag

Figure 2.54. Inclination History for Two Revolutions of the Low Altitude Elliptical Test Case



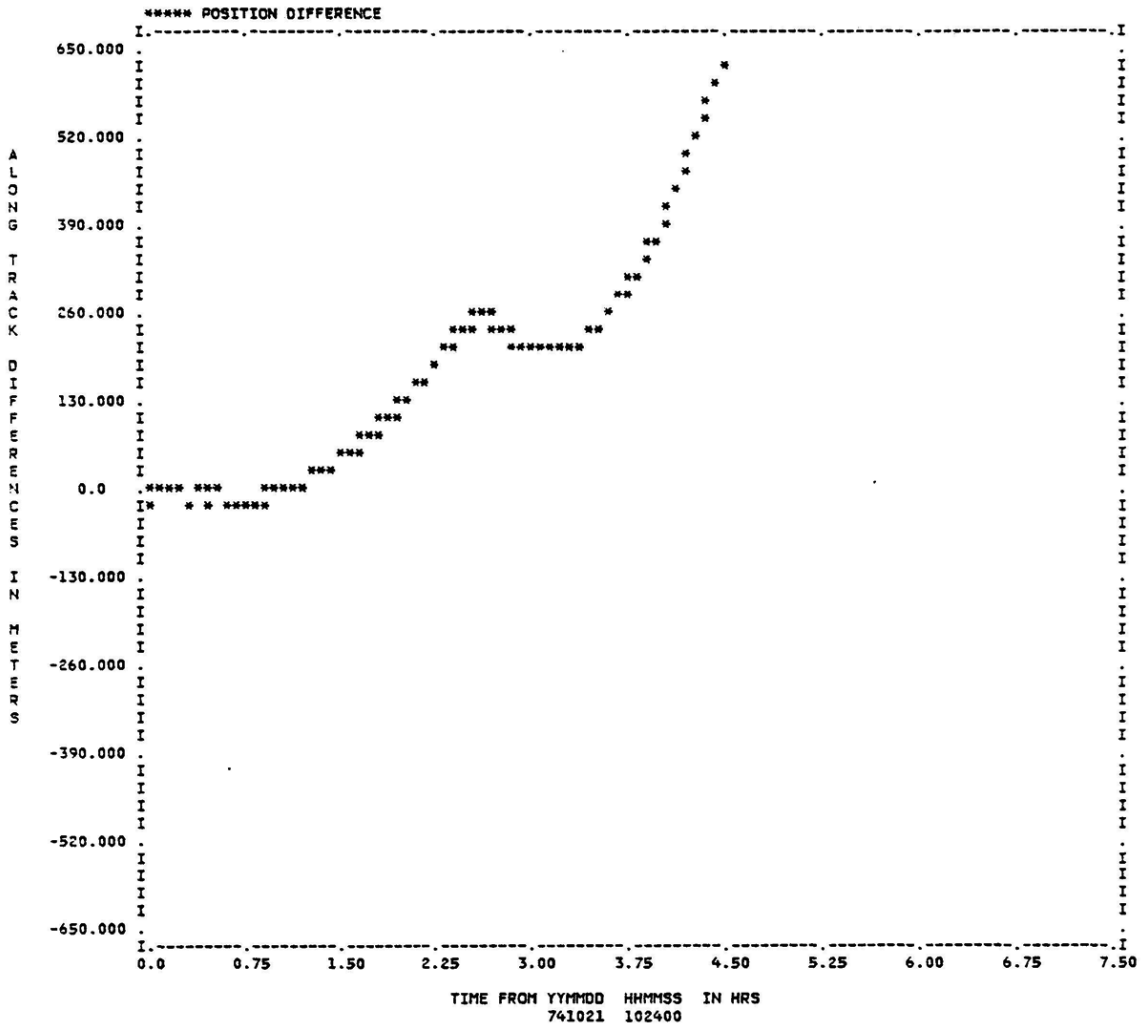
Epoch Conditions: Table 2.4

Force Model: J_2-J_8, no drag

It must now be recalled that the discrete Chebyshev orthogonality points are weighted such that more points lie towards the ends of the interval of representation (see Figure 2.1), with fewer points in the middle of the interval. This is useful when discontinuities are encountered at the ends of the interval of representation. However, in the case described above it may be possible to encounter a nearly discontinuous element rate in the middle of the interval of representation where the Chebyshev orthogonality points are least dense. This could occur due to an unfortunate combination of epoch mean anomaly and integration stepsize, such that the perigee passage would occur near the middle of the integration step. It is the Chebyshev points and their role in the evaluation of the Chebyshev series coefficients that determine the quality of the series representations for the integrands of the equations of motion. If nearly discontinuous changes in these integrands were to occur near the middle of the interval of representation, it is likely that there would be an insufficient number of orthogonality points in that region to accurately reflect those changes. Therefore, the resulting integrated solutions would be in error. This error would be compounded at each integration step, causing the Picard-Chebyshev integration to diverge rapidly from the Cowell reference trajectory.

While the above difficulty does not appear to cause significant error growth in the case of zonal perturbations, the inclusion of drag in the force model changes the results dramatically. The results for test run 2 in Table 2.8 indicate that the inclusion of drag in the perturbation model, keeping all other factors the same as test run 1, causes an increase of nearly 6 orders of magnitude in the maximum along track difference over the 4.5 hour compare span. Figure 2.55 gives the actual time history of the along track difference over the compare span. It is easily seen that the difference remains relatively small until the first perigee passage and then grows quite rapidly. This is due to the Chebyshev series not accurately representing the drag and zonal perturbations in the perigee region. The resulting errors in the integrated series representations of the position and velocity components (Cowell formulation) or of the orbital elements (VOP formulation) is reflected in several ways. A predominant error is the resulting mismatch in mean motions between the reference Cowell integration and the Picard-Chebyshev integration. This is a prime cause of the rapid secular growth in the along track error indicated during the first 2.25 hours of the compare span. The second perigee passage occurred near the middle of the second 2.25 hour integration step.

Figure 2.55. Along Track Difference Over 4.5 Hour Compare Span: Picard-Chebyshev (1 Rev Stepsize) versus Time Regularized Cowell Integration



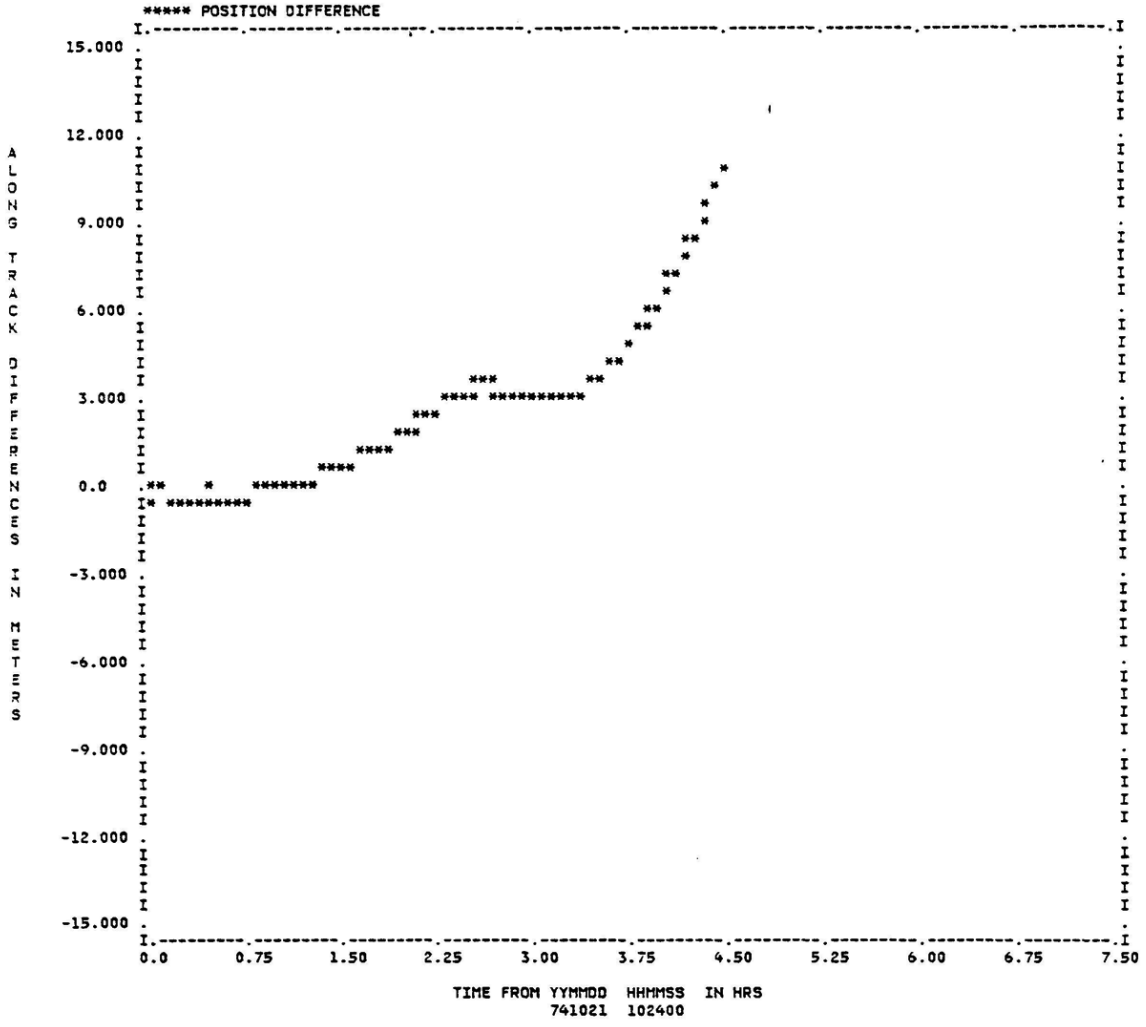
Epoch Conditions: Table 2.4

	Picard-Chebyshev (M=64, N=69)	Cowell
Stepsize:	8100 sec	200 steps/rev
Field:	8x0	8x0
Drag:	yes	yes
Formulation:	Cowell	Cowell

Figure 2.55 indicates that the mean motion error was compounded in the second integration step and that the along track error is growing at an even faster rate.

The error growth described above can be significantly reduced by reducing the Picard-Chebyshev integration stepsize. Test runs 3 and 4 reduced the original 8100 second stepsize (one satellite orbital period) by factors of 2 and 4 respectively. The results in Table 2.8 for these test runs indicate the maximum along track differences were reduced about one order of magnitude for each stepsize reduction. The primary effect of reducing the stepsize is to include more orthogonality points in the computation of series coefficients for any given time span. For example, when an integration stepsize of one satellite period is used, seventy orthogonality points must represent the integrands over that whole orbital revolution. On the other hand, using an integration stepsize of one-fourth the satellite period places 280 orthogonality points around the orbit and allows a much more accurate Chebyshev series representation of the right hand sides of the equations of motion and of their resulting integrals. Figure 2.56 presents the along track error difference for test run 4, which uses a one-fourth period stepsize for the Picard-Chebyshev integrator. This figure graphically demonstrates

Figure 2.56. Along Track Difference Over 4.5 Hour Compare Span: Picard-Chebyshev (1/4 Rev Stepsize) versus Time Regularized Cowell Integration



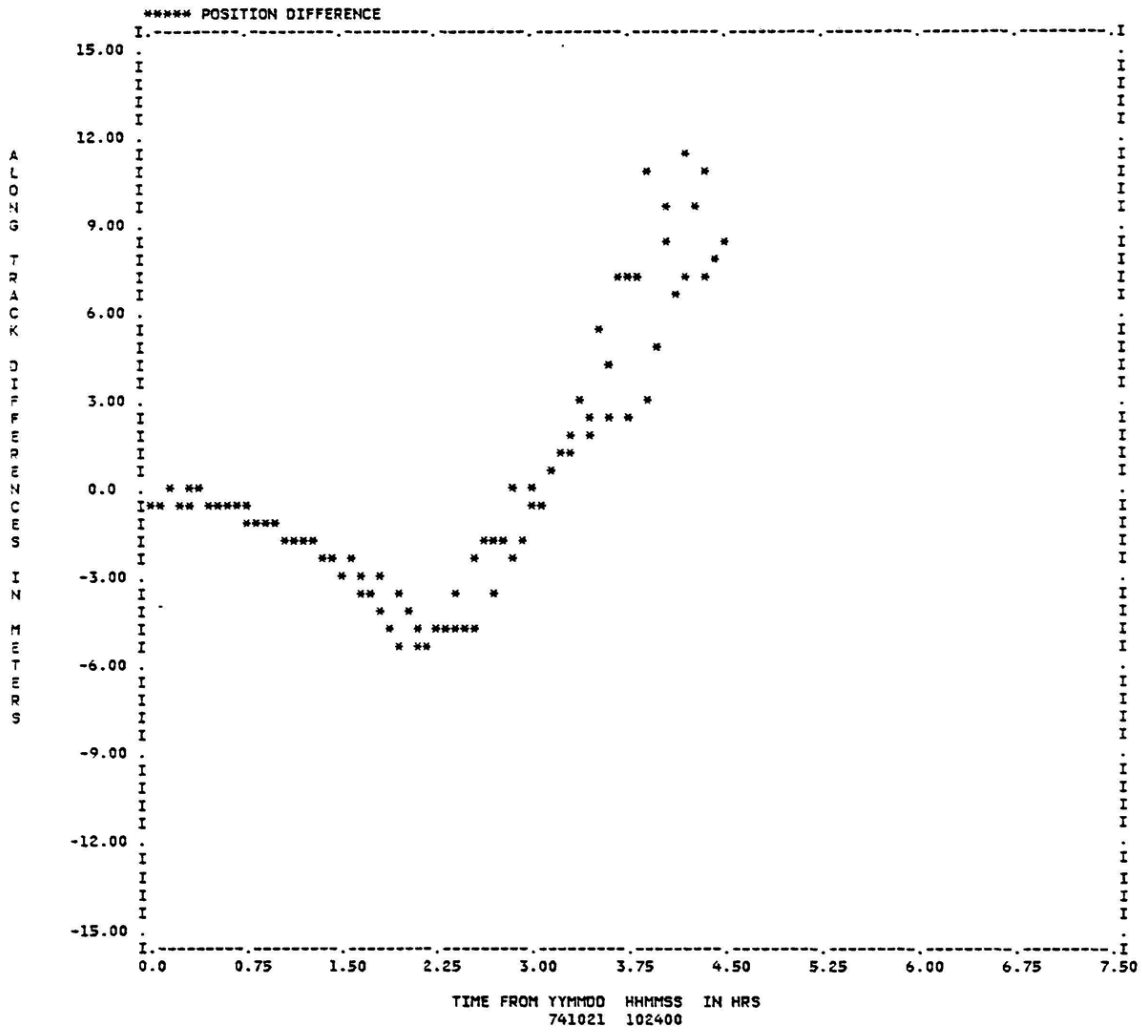
Epoch Conditions: Table 2.4

	Picard-Chebyshev (M=64, N=69)	Cowell
Stepsize:	2025 sec	200 steps/rev
Field:	8x0	8x0
Drag:	yes	yes
Formulation:	Cowell	Cowell

the above conjecture, in that the along track difference is significantly smaller than that of test run 2 (Figure 2.55) and its rate of growth is much less.

Test run 5 uses the same conditions as test run 2, but the VOP equations of motion are being integrated. The error behavior is essentially the same as of test run 2, with a maximum along track error of 1 km over the 4.5 hour compare span. In order to clearly demonstrate the effect that the distribution of the Chebyshev orthogonality points can have on the accuracy of the integration results a simple test was made. In test run 6, the same conditions as test run 5 were used, with the exception that the epoch mean anomaly was set to zero degrees. This has the effect of clustering the majority of the Chebyshev orthogonality points within the perigee region where the perturbing accelerations are strongest. This simple change in epoch conditions reduces the maximum along track error by two orders of magnitude, as shown by Table 2.8 and Figure 2.57 and indicates that the problem of the distribution of orthogonality points within the interval of representation can certainly influence the accuracy of the Picard-Chebyshev algorithm.

Figure 2.57. Along Track Difference Over 4.5 Hour Compare Span: Picard Chebyshev (1 Rev Stepsize; Epoch Mean Anomaly = 0°) versus Time Regularized Cowell Integration



Epoch Conditions: Table 2.4, but epoch mean anomaly M = 0°

	Picard-Chebyshev (M=48, N=69)	Cowell
Stepsize:	8100 sec	200 steps/rev
Field:	8x0	8x0
Drag:	yes	yes
Formulation:	Equinoctial VOP (λ)	Cowell

Finally, it was of interest to examine briefly the ability of the Picard-Chebyshev algorithm to handle tesseral harmonics in the perturbation force model for this low altitude elliptical test case. Therefore, test run 8 used a 6x6 harmonic field and no drag in an integration of the equinoctial VOP equations. As shown in Table 2.8, the resulting Picard-Chebyshev integration agreed extremely well with the time regularized Cowell reference trajectory, having a maximum position error of only 10 cm after 4.5 hours of integration.

In general, the above test runs seem to indicate that the Picard-Chebyshev parallel orbit propagation algorithm produces ephemeris data of comparable accuracy to current accepted high precision integration techniques. It is felt that this has been demonstrated over a broad range of orbital geometries, integration stepsizes and formulations of the equations of motion. Some care must be taken in selecting the proper stepsize for cases of eccentric, drag-perturbed orbits, but this should present little problem to a potential algorithm user. The results presented in this chapter show little difference in accuracy between integrating the Cowell formulation and integrating the VOP equations of motion. However, this study has indicated that the VOP equations have a significantly faster rate

of convergence in the Picard method than do the Cowell equations. This characteristic of the Picard-Chebyshev algorithm will be discussed further in Chapter 4.

CHAPTER 3

A PARALLEL ORBIT ESTIMATION ALGORITHM

The previous chapter developed and demonstrated a highly parallel numerical integration algorithm. The algorithm was applied to the problem of propagating a set of orbital initial conditions forward in time. The numerical results showed that the Picard-Chebyshev method provided comparable accuracy to current high-precision Cowell numerical integrations. Thus, it has been shown that this algorithm could provide a parallel processing orbit determination system with a viable orbit propagation capability.

It is well known that over time the true satellite trajectory will deviate from that trajectory predicted by any orbit propagator. This is mainly due to incomplete or incorrect modelling of the perturbation dynamics and to numerical error accumulation in the integration process. Therefore, observations of the satellite motion are taken periodically in an attempt to improve the knowledge of the true satellite trajectory. The problem is complicated by the fact that the observations are not perfect, but are

corrupted by a noise component. The problem of orbit determination then is to find accurate values for certain parameters considered representative of the true satellite motion, using as input an initial guess of the parameter values, a set of noisy measurements, a deterministic model of the satellite motion and any appropriate statistical information.

All major orbit determination systems perform the task described above. Section 1.1.2 lists the primary methods of accomplishing this task for several major OD systems. To be a viable system, a parallel processing orbit determination system must also be able to perform the estimation task, but the algorithms used are again required to exhibit a high degree of parallelism to provide the expected performance improvement and to allow the system to operate efficiently. Hence, the second task required of a parallel processing orbit determination system is to incorporate noisy measurements of the satellite motion to improve the estimate of certain parameters representative of the true satellite motion.

This chapter will discuss a highly parallel algorithm for performing the orbit estimation task. The algorithm is a nonlinear function minimization technique and is gradient

dependent. Section 3.1 discusses the steps in the algorithm and Section 3.2 describes the application of the algorithm to the orbit estimation problem. In this implementation, the deterministic model for the satellite motion is a Special Perturbations formulation of the satellite equations of motion, using the Picard-Chebyshev method as the numerical integration technique. This required a special treatment for the evaluation of the cost functional and its gradient vector. This special treatment is described in Section 3.3. The fact that the gradient is computed as an inherent part of the method allows the covariance matrix of the estimation errors to be simply computed, as shown in Section 3.4. Section 3.5 discusses the architecture of the estimation algorithm, with emphasis on the parallelism displayed by this formulation. Finally, the numerical reulsts of the orbit estimation algorithm are presented in Section 3.6.

3.1 Algorithm Flow

The algorithm used for the parallel processing orbit estimation task is a nonlinear function minimization technique due to Straeter. The algorithm is a variable metric algorithm which Straeter developed specifically for parallel computing or vector streaming (pipeline) computer architectures. As previously stated, the algorithm requires the computation of the gradient of the cost functional with

respect to those parameters being estimated. The development of the algorithm and proofs of certain relevant properties are documented in Reference 139.

Straeter's method was chosen for use in the parallel processing orbit estimation algorithm for several reasons. First and foremost, the algorithm was expressly designed for parallel computation. Its major architectural structure is therefore compatible with the candidate parallel processor hardware architecture, a rarity in most current numerical algorithms. Second, the method has been successfully applied in a different context than the orbit estimation problem. Travassos and Kaufman (87, 146, 147) have applied the Parallel Variable Metric (PVM) method of Straeter to the solution of nonlinear two-point boundary value problems arising in the design of optimal nonlinear flight controllers. Their results indicate that the PVM method is very effective for determining the optimal solve-for parameters in their formulation and that the associated computation time in a parallel processor environment can be siginficantly less than that required by highly efficient serial minimization methods. Third, recent results by Adimurthy and Joy (4, 5) have indicated that nonlinear function minimization techniques may have convergence characteristics far superior to those exhibited by classical differential correction methods. That is,

some function minimization techniques will converge for certain initial estimates of the solve-for parameters for which other methods will not converge. This algorithm "robustness" enhances the probability of achieving a converged estimate of the solve-for parameters for cases where the *a priori* solve-for vector is not well known, or where the observation span is short. Finally, the PVM method requires the computation of the gradient of the cost functional with respect to the solve-for parameters. This fact allows simple computation of the covariance matrix of the estimation errors. While the covariance matrix computation is not required by the PVM method, it represents data that is essential to orbital analysts in "real world" orbit determination systems and therefore should be provided by the parallel processing orbit determination system.

3.1.1 Parallel Variable Metric (Straeter's) Method

The Parallel Variable Metric (PVM) method is a gradient method for the solution of nonlinear parameter estimation problems. The characteristics of gradient methods in optimization are well documented. Bard (26) presents a comparison of gradient methods for the solution of nonlinear parameter optimization problems. His research indicated that modifications of the Gauss method performed better than variable metric methods. However, it must be remembered that the

formulation of the Gauss method disregards certain terms containing the measurement residuals by assuming that these residuals will be small. If the initial guess, or *a priori* estimate, is far away from the true solution, the measurement residuals will not be small and the Gauss method may not converge. This conjecture is born out by the results of Adimurthy and Joy, which indicate that the Davidon-Fletcher-Powell algorithm (with modification by Stewart) and Fletcher's method, both variable metric-type algorithms, had significantly wider radii of convergence than the other methods tested.

The basic approach in gradient methods is to form a search direction of the form (26)

$$\vec{s} = R\vec{g} \tag{3.1}$$

where R is a matrix and $\vec{g}$ is the gradient of the function being minimized. These methods then generate successive approximations to the optimal solution by adjusting the previous iterate in the following manner

$$\vec{x}^{i+1} = \vec{x}^{i} + \alpha_i \vec{s}_i \tag{3.2}$$

where α_i is the scalar step length along the search direction which extremizes the function in question. If, for example, R is taken to be the identity matrix, then the method becomes Cauchy's method of steepest descent. If R is taken to be the

negative of the inverse of the Hessian matrix of the function to be minimized, then the Newton-Raphson method results. When R is an approximation to the Hessian formed by the outer product of the gradient vector with itself, the formulation is called Gauss' method. Variable metric algorithms successively approximate the negative of the inverse of the Hessian by successive additions of low-rank matrices. Dyer and McReynolds (67) present the fundamental properties and characteristics of gradient algorithms in the context of the theory of optimal control. Dahlquist, Björck and Anderson (59) also contains an excellent discussion of these types of algorithms, with special emphasis on the potential numerical difficulties in implementing these methods.

3.1.1.1 Steps in the PVM Method

The problem to be solved by the PVM method is to find the vector $\vec{x}^*$ which minimizes a differentiable scalar functional $J(\vec{x})$. The cost functional $J(\vec{x})$ is assumed to have a gradient $\vec{g}(\vec{x})$, such that

$$\vec{g}^T(\vec{x}) = \frac{\partial J(\vec{x})}{\partial \vec{x}} \tag{3.3}$$

Following the development by Straeter in Reference 139, the *a priori* estimate of the solve-for vector $\vec{x}$ is

$$\vec{x}_{ap} = \vec{x}^1 \tag{3.4}$$

The solve-for vector $\vec{x}$ is assumed to be an element of a finite separable Hilbert space H on which the inner product is given functionally by $(\cdot, \cdot)$. In this development, the inner product on this finite-dimensional vector space will be given by

$$(\vec{x},\vec{y}) = \vec{x}^T\vec{y}$$

When the solve-for vector, and therefore the Hilbert space, is of finite-dimension n, it is assumed that a set of basis vectors $\Sigma = \{\hat{\sigma}_j\}$, (j = 1,2, ... ,n) exist for H. For the set of vectors Σ to be basis vectors, it is required that the vectors $\hat{\sigma}_j$, (j = 1,2, ... ,n) be linearly independent and span the vector space H. The vector, or Hilbert, space H may be thought of as the space of n solve-for parameters defined by the basis vectors $\hat{\sigma}_j$, (j = 1,2, ... ,n). The scalar functional $J(\vec{x})$ represents an n+1-dimensional hypersurface in the parameter space. The element $\vec{x}^*$ which locates the minimum of the hypersurface is sought. The direction of search for the minimizing value of $\vec{x}$ is found by Equation 3.1, where R is the current iterate of the metric matrix and $\vec{g}(\vec{x})$ is the current iterate of the gradient of $J(\vec{x})$.

The initialization of the method involves the evaluation of the cost and gradient at the *a priori* solve-for vector given by Equation (3.4). Thus

$$J(\vec{x}_{ap}) = J(\vec{x}^1)$$

$$\vec{g}(\vec{x}_{ap}) = \vec{g}(\vec{x}^1)$$

If the magnitude of $\vec{g}(\vec{x}^1)$ is sufficiently near zero, then the *a priori* estimate is in fact the minimizing value of $\vec{x}$ and no further computation is required. If $(\vec{g}(\vec{x}),\vec{g}(\vec{x})) \neq 0$, then the PVM method is entered. In the PVM method, Straeter assumes that his candidate parallel processor has a degree of parallelism p equivalent to the dimension of the solve-for parameter space n. Therefore, the parallel paths in this algorithm are keyed to the dimension of the solve-for vector $\vec{x}$. A single iteration of the PVM method is described below.

STEP 1: The first step in the PVM method involves parallel computations of the cost and gradient vector at p distinct values of the solve-for vector $\vec{x}$. The p values of the solve-for vector are given by

$$\vec{x}_j = \vec{x}^i + \hat{\sigma}_j \qquad (3.5)$$

for j = 1,2,...,p

and $\vec{x}^i$ = i^{th} iterate of the solve-for vector

Note: For i = 1, $\vec{x}^1$ equals the *a priori* estimate of the solve-for vector.

It is important to notice that the j^{th} value of the solve-for vector in Equation 3.5 is a variation only of the j^{th} element in $\vec{x}^i$. This is because the $\hat{\sigma}_j$ basis vector can only influence the coefficient of $\hat{\sigma}_j$ in

$$\vec{x}^i = x_1{}^i\hat{\sigma}_1 + x_2{}^i\hat{\sigma}_2 + \ldots + x_j{}^i\hat{\sigma}_j + \ldots + x_n{}^i\hat{\sigma}_n$$

Thus, the p = n values of the solve-for parameters $\vec{x}_j$, (j = 1,2, ... ,p) are independent and the cost and gradient evaluations at these points are also independent and therefore perfect candidates for parallel processing. Hence, the p evaluations of the cost functional and its gradient vector are computed in parallel as

$$J_j = J(\vec{x}_j) \qquad (3.6.a)$$

$$\vec{g}_j = \vec{g}(\vec{x}_j) \qquad (3.6.b)$$

$$\text{for } j = 1,2,\ldots,p$$

If any of the p gradient vectors computed by Equation 3.6.b satisfies the condition

$$(\vec{g}(\vec{x}_j),\vec{g}(\vec{x}_j)) \simeq 0$$

$$\text{for any } j \text{ in } 1 \leq j \leq p$$

and its corresponding cost functional value $J(\vec{x}_j)$ is less than all other computed costs, then $\vec{x}_j$ is the value of the solve-for vector that minimizes the cost functional $J(\vec{x})$ and the algorithm is terminated. Otherwise, the algorithm proceeds to Step 2.

STEP 2: The second step of the PVM method involves the modification of the metric matrix by p rank-one corrections. The ith iterate of the metric matrix is defined to be $V^{(i)}$ and is assumed to be a self-adjoint linear operator in the space H. The zeroth iterate $V^{(0)}$ is assumed to be the identity matrix, such that

$$V^{(0)} = I \tag{3.7}$$

The metric matrix at the ith iteration determines the search direction vector at that iteration from Equation (3.1), that is $R = -V^{(i)}$. As the number of iterations increases, the metric matrix approaches the inverse of the Hessian matrix of the function being minimized. Straeter shows that, if the cost functional is quadratic, the convergence occurs in one parallel iteration. The resulting solve-for vector is the minimizing value of $\vec{x}$ such that

$$\vec{x}^{1} = \vec{x}^{*}$$

and $$V^{(1)} = G^{-1}$$

where G is the Hessian matrix.

It is first necessary to compute in parallel the p vectors $\vec{y}$ defined by

$$\vec{y}_j = \vec{g}(\vec{x}_j) - \vec{g}(\vec{x}^i) \tag{3.8}$$

for $j = 1,2,\ldots,p$

where i = iteration number = $1,2,\ldots$

Next, the p vectors called residual vectors $\vec{r}$ must be computed. The residual vectors are computed by

$$\vec{r}_1 = V^{(i-1)}\,\vec{y}_1 - \hat{\sigma}_1 \tag{3.9a}$$

and

$$\vec{r}_j = V^{(i-1)}\,\vec{y}_j - \hat{\sigma}_j + \sum_{k=1}^{j-1} \tau_k(\vec{r}_k,\vec{y}_j)\vec{r}_k \tag{3.9b}$$

for $j = 2,3,\ldots,p$

and $i = 1,2,\ldots$

where the $\vec{y}_j$'s are computed by Equation (3.8). The p scalars τ_k ($k = 1,2,\ldots,p$) are computed by the formula

$$\tau_k = \begin{cases} -(\vec{r}_k,\vec{y}_k)^{-1} & \text{for } (\vec{r}_k,\vec{y}_k) \neq 0 \\ 0 & \text{for } (\vec{r}_k,\vec{y}_k) = 0 \end{cases} \tag{3.10}$$

It must be noted here that the computation of each residual vector cannot be done in parallel. The summation term for the ℓth residual vector requires the use of the ℓ-1 previously computed residual vectors. While not a fully parallel computation, this computation is suited for a vector-streaming or pipeline computer. For simplicity, however, this computation will be considered to be a serial computation in the performance analysis of Chapter 4.

Once the residual vectors $\vec{r}_j$, (j = 1,2,...,p) and the scalars τ_j, (j = 1,2,...,p) are computed, the p-dimensional metric matrix $V^{(i-1)}$ can be updated by p correction matrices. The new iterate of the metric matrix is given by

$$V^{(i)} = V^{(i-1)} + \sum_{j=1}^{p} \tau_j B^{(j)} \qquad (3.11)$$

for $i = 1,2,...$

Straeter assumes that the $B^{(j)}$'s are self-adjoint linear operators that map the Hilbert space H into itself. The $B^{(j)}$'s are defined such that

$$B^{(j)} \vec{x} = (\vec{x},\vec{r}_j)\ \vec{r}_j \qquad (3.12)$$

for all $\vec{x} \in H$

For the finite dimensional real vector spaces assumed here, the assumption of self-adjointness can be translated into the condition that the pxp matrices $B^{(j)}$, (j = 1,2,...,p) be symmetric. A simple formula for computing these rank-one correction matrices may be found by using the definition of the inner product on the vector space H, which is $(\vec{x},\vec{y}) = \vec{x}^T\vec{y} = \vec{y}^T\vec{x}$. Taking the inner product of both sides of Equation (3.12) with an arbitrary vector element $\vec{x}$ of the space H gives

$$(B^{(j)}\,\vec{x},\vec{x}) \;=\; ((\vec{x},\vec{r}_j)\,\vec{r}_j,\vec{x}) \qquad (3.13a)$$

Applying the inner product definition to the left-hand side of Equation (3.13a) yields

$$\begin{aligned}(B^{(j)}\vec{x},\vec{x}) &= (B^{(j)}\vec{x})^T\,\vec{x} \\ &= \vec{x}^T\,B^{T(j)}\,\vec{x}\end{aligned}$$

But, since $B^{(j)}$ is symmetric (self-adjoint) on the space H, $B^{(j)} = B^{T(j)}$. Thus, Equation (3.13a) becomes

$$\vec{x}^T\,B^{(j)}\,\vec{x} \;=\; ((\vec{x},\vec{r}_j)\,\vec{r}_j,\vec{x}) \qquad (3.13b)$$

Looking now at the right-hand side and applying the basic inner product definition and noting that the term $(\vec{x},\vec{r}_j)$ is a scalar which may be taken outside the inner product, the following results

$$((\vec{x},\vec{r}_j)\ \vec{r}_j,\vec{x}) \quad = \quad (\vec{x},\vec{r}_j)(\vec{r}_j,\vec{x})$$

Using the inner product definition, this can be rewritten as

$$((\vec{x},\vec{r}_j)\ \vec{r}_j,\vec{x}) \quad = \quad \vec{x}^T\ \vec{r}_j\ \vec{r}_j^{\,T}\ \vec{x}$$

and Equation (3.13b) becomes

$$\vec{x}^T\ B^{(j)}\ \vec{x} \quad = \quad \vec{x}^T\ \vec{r}_j\ \vec{r}_j^{\,T}\ \vec{x} \qquad (3.13c)$$

But $\vec{x}$ was assumed to be any arbitrary vector element of the vector space H. For Equation (3.13c) to hold for all $\vec{x} \in H$, the desired result follows as

$$B^{(j)} \quad = \quad \vec{r}_j\ \vec{r}_j^{\,T} \qquad (3.14)$$

for all $\vec{x} \in H$

and $j \quad = \quad 1,2,\ldots,p$

Thus the rank-one correction matrices are easily computed from the dyadic products of the residual vectors given by Equations (3.9a) and (3.9b). The updated metric matrix is then given by Equation (3.11) and this updated metric matrix determines the new search direction in the form of Equation (3.1) used in Step 3 of the method.

STEP 3: The final step in the PVM method involves minimizing the cost functional along the Newton-like search direction vector given by

$$\vec{s}_i = -V^{(i)} \vec{g}(\vec{x}^i) \qquad (3.15)$$

for i = iteration number = 1,2,...

The search direction is termed Newton-like because the $V^{(i)}$'s successively approximate the inverse of the Hessian matrix, as i increases. As previously stated, when the R matrix in Equation (3.1) equals the negative of the inverse of the Hessian matrix, the method is called Newton's method. Thus, while the PVM search direction is not exactly the Newton search direction, the PVM search direction approaches that of the Newton method as the number of iterations increases, and is hence called Newton-like.

The new iterate of the solve-for vector is formed by

$$\vec{x}^{i+1} = \vec{x}^{i} + \alpha_i \vec{s}_i \qquad (3.16)$$

where $\vec{s}_i$ is found from Equation (3.15). The scalar stepsize α_i represents the distance along the search direction which must be taken to yield the minimum value of the cost functional. In essence, the p-dimensional minimization problem with respect to the p components of the solve-for vector $\vec{x}$ has been reduced to a one-dimensional minimization problem with respect to α_i, where the direction of search is specified by $\vec{s}_i$. The value of α_i may be found by any of a number of one-dimensional search techniques. The simple method chosen for this implementation is called Coggins Algorithm and is taken from Reference 97. The Coggins method combines the method of Davies, Swann and Campey with a method due to Powell. The Davis, Swann and Campey algorithm is used to rapidly find three points that bracket the minimum of the function along the search direction. This algorithm contains an acceleration feature which increases the search stepsize until the minimum is located. Once the minimum is bracketed by three points, the Powell algorithm then fits a quadratic to those points and analytically finds the minimizing value of α_i.

With α_i determined from the one-dimensional search given above, the new iterate of the solve-for vector is then given by Equation (3.16). The gradient vector at the new iterate of the solve-for vector is computed. If the gradient vector is sufficiently near zero, the algorithm is terminated. That is, if

$$(\vec{g}(\vec{x}^{i+1}), \vec{g}(\vec{x}^{i+1}) < \varepsilon^2$$

where ε represents some input convergence criterion

then the element $\vec{x}^{i+1} = \vec{x}^*$, the element of Π which minimizes $J(\vec{x})$ as desired. If not, the algorithm returns to Step 1 with the new $\vec{x}^{i+1}$ as the new iterate of the solve-for vector.

As in the parallel integration algorithm, the primary emphasis has been to induce a high degree of parallelism in the algorithm. Step 1 and part of Step 2 show this high degree of parallelism. Additionally, the parallel operations include p parallel evaluations of the cost and gradient vector. These computations are usually considered to be the most time-consuming computations in these types of algorithms. As such, the application of parallelism

in their computation yields a significant performance advantage.

3.1.2 Example of the PVM Method

For clarity, a simple illustration of the PVM method has been included. The problem is to minimize a quadratic function of two variables. Thus, the dimension of the vector space is 2 and n = p = 2. The solve-for vector is $\vec{x} = (x_1, x_2)^T$. The scalar functional to be minimized is given by

$$J(x_1,x_2) = J(\vec{x}) = x_1^2 + 2.1157x_2^2 - 6x_1 - 12.6492x_2 + 28.04 \qquad (3.17)$$

From a property of the PVM method demonstrated by Straeter, it is expected that the PVM method will converge to the minimum of $J(\vec{x})$ in a single iteration through the algorithm for this purely quadratic functional. It is easily shown that in any plane J = constant, this form of the cost functional results in an ellipse. The surface formed is an elliptic paraboloid centered at (3,3) in the (x_1,x_2) plane. Figure 3.1 depicts graphically the form of the cost functional projected on the (x_1,x_2) plane, with the lines of equal cost shown. Figure 3.2 shows another projection of

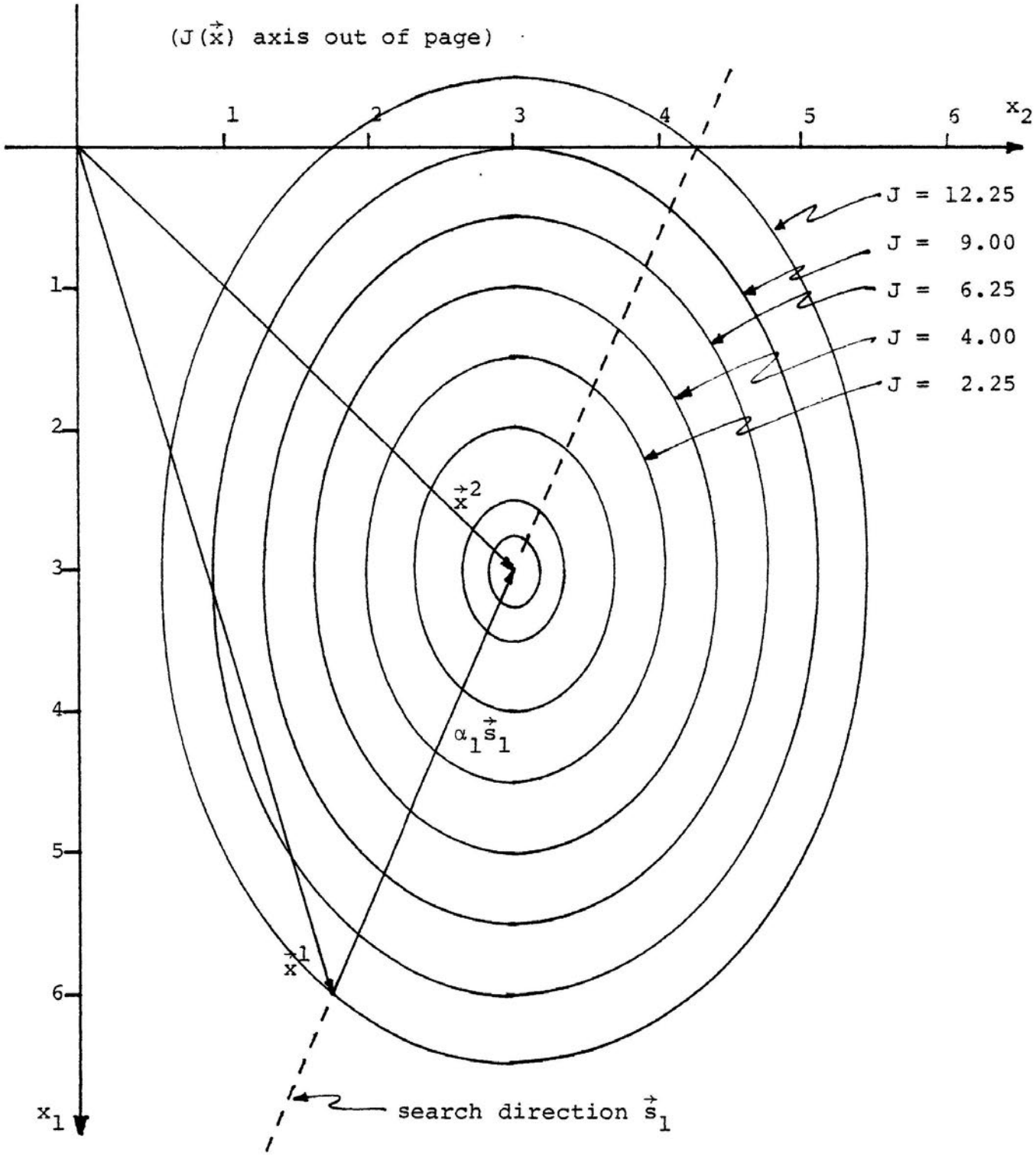


Figure 3.1. Example Problem Cost Functional Projected on (x_1, x_2) Plane

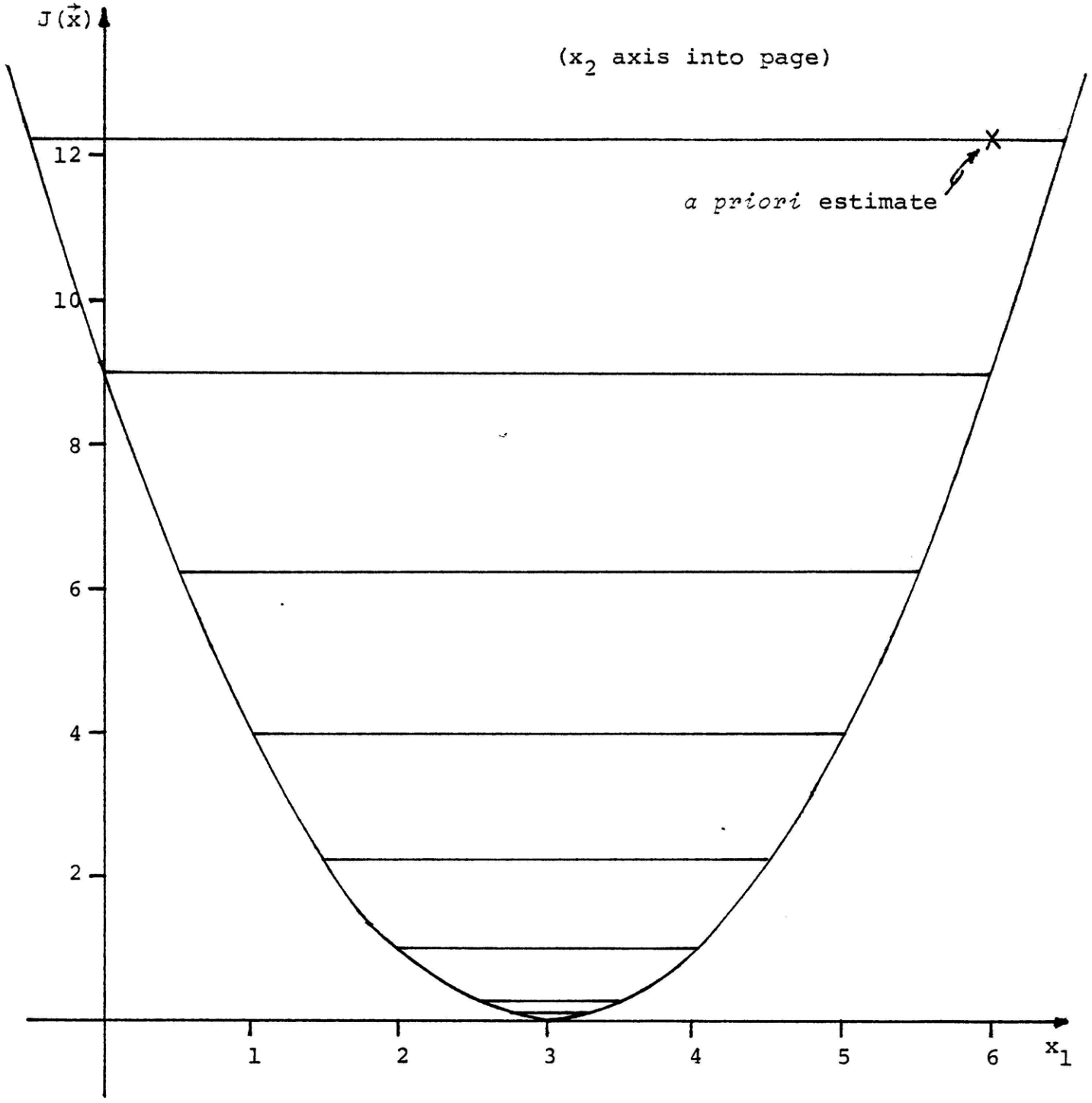


Figure 3.2. Example Problem Cost Functional Projected on $(x_1, J(\vec{x}))$ Plane

the surface onto the $(x_1, J(\vec{x}))$ plane.

With the cost functional defined by Equation (3.17), its gradient is then computed by

$$\vec{g}(\vec{x}) = (2x_1 - 6)\hat{\sigma}_1 + (4.2314x_2 - 12.6492)\hat{\sigma}_2 \tag{3.18}$$

where $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the basis vectors of the 2-dimensional parameter space. Following Straeter, these will not be taken as unit vectors but will be scaled to be about 10^{-4} of the order of magnitude of their respective parameter values. Since both x_1 and x_2 are of order one, both $\hat{\sigma}_1$ and $\hat{\sigma}_2$ will be of order 10^{-4}. Assume that the *a priori* value of the solve-for vector is specified by

$$\vec{x}^1 = \begin{pmatrix} 6.00 \\ 1.76 \end{pmatrix} \tag{3.19a}$$

and the zeroth iterate of the metric matrix is the identity matrix, such that

$$V^{(0)} = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.19b}$$

The *a priori* value of $\vec{x}$ represents a point of the $J(\vec{x}) = 12.25$ equal-cost curve. This point is shown on Figures 3.1 and 3.2. The resulting gradient at the a priori solve-for vector is

$$\vec{g}(\vec{x}^1) = \begin{pmatrix} 6.00 \\ -5.24 \end{pmatrix} \tag{3.19c}$$

Since $(\vec{g}(\vec{x}^1), \vec{g}(\vec{x}^1))^{1/2}$ is not less than the convergence criterion of 10^{-7}, the iterative portion of the PVM method is entered.

Figure 3.3 presents the results of the PVM algorithm for this example problem. After initialization, the two vectors $\vec{x}_1$ and $\vec{x}_2$ are formed by Equation (3.5) in parallel. Two parallel evaluations of the cost and gradient are then performed and the two $\vec{y}$ vectors are computed from Equation (3.8) in parallel. The results of these calculations are given in Equations (3.20a), (3.20b) and (3.20c) in Figure 3.3. The calculation of the two residual vectors $\vec{r}_1$ and $\vec{r}_2$ defined by Equations (3.9a) and (3.9b) are done serially, including the computation of the two scalars τ_1 and τ_2 defined by Equation (3.10). The rank-one correction matrices can now be computed in parallel from Equation (3.14). These numerical results are given by Equations

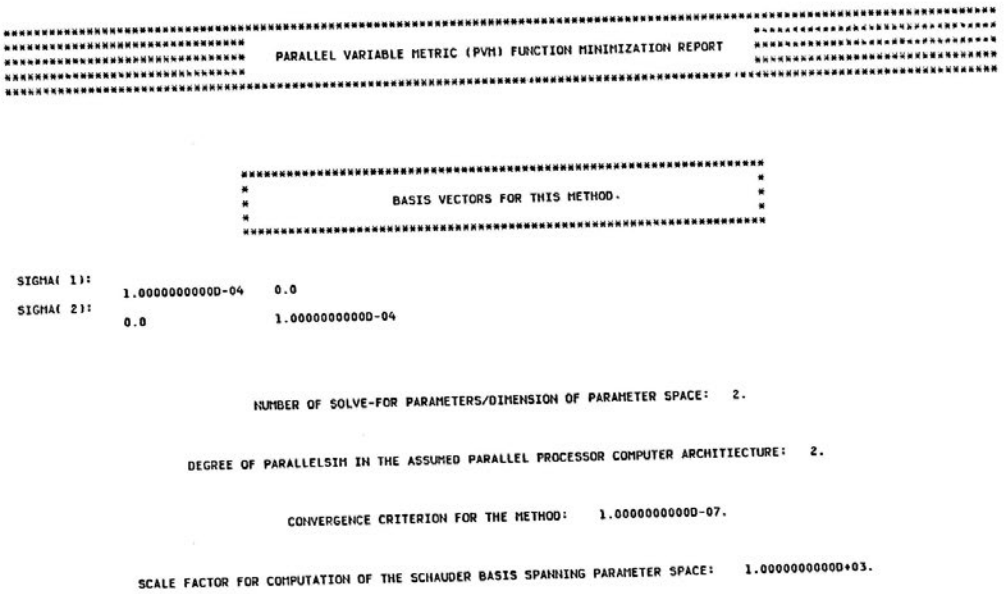


Figure 3.3. PVM Algorithm Results for Example Problem (Page 1 of 8)

```
**********************************************************************************************************
**********                          ITERATION NUMBER    0                          **********
**********************************************************************************************************


 CURRENT SOLVE-FOR VECTOR:
      X:
            6.0000000000D+00    1.7605917490D+00


 COST FUNCTIONAL EVALUATED AT THE CURRENT ITERATE OF THE SOLVE-FOR VECTOR:
   J(X):    1.2250000230D+01


 GRADIENT OF THE COST FUNCTIONAL EVALUATED AT THE CURRENT ITERATE OF THE SOLVE-FOR VECTOR:
   G(X):
            6.0000000000D+00   -5.2444382183D+00


 SQUARE OF THE MAGNITUDE OF THE CURRENT GRADIENT VECTOR:
 G DOT G:   6.3504132225D+01


 CURRENT ITERATE OF THE METRIC MATRIX V(N):
 ROW( 1):
            1.0000000000D+00    0.0
 ROW( 2):
            0.0                 1.0000000000D+00


      NUMBER OF FUNCTION EVALUATIONS (SEQUENTIAL) IN THE UNIVARIATE SEARCH METHOD FOR ITERATION  0:   0.


           NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EXCLUDING UNIVARIATE SEARCH)
           IN SEQUENTIAL EXECUTION OF PVM METHOD FOR ITERATION  0:   2.


           NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (INCLUDING UNIVARIATE SEARCH)
           IN SEQUENTIAL EXECUTION OF PVM METHOD FOR ITERATION  0:   2.


          EFFECTIVE NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EXCLUDING UNIVARIATE
          SEARCH) IN PARALLEL EXECUTION OF PVM METHOD FOR ITERATION  0:   2.


          EFFECTIVE NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (INCLUDING UNIVARIATE
          SEARCH) IN PARALLEL EXECUTION OF PVM METHOD FOR ITERATION  0:   2.
```

Figure 3.3. PVM Algorithm Results for Example Problem (Page 2 of 8)

```
*****************************************************************
*                                                               *
*          CONVERGENCE NOT ACHIEVED -- CONTINUE SEARCH          *
*                                                               *
*****************************************************************

 THE  2 VECTORS AT WHICH THE COST AND GRADIENT MUST BE COMPUTED (IN PARALLEL):
  X( 1):
            6.0001000000D+00    1.7605917490D+00
  X( 2):
            6.0000000000D+00    1.7606917490D+00                    (3.20a)

 THE  2 EVALUATIONS OF THE COST FUNCTIONAL AND THE GRADIENT VECTOR:

 J(X 1):    1.2250600240D+01
 G(X 1):
            6.0002000000D+00   -5.2444382183D+00
 J(X 2):    1.2249475808D+01                                        (3.20b)
 G(X 2):
            6.0000000000D+00   -5.2440150778D+00

 THE DOT PRODUCTS OF EACH OF THE  2 GRADIENT VECTORS WITH ITSELF:
GDOTG( 1):    6.3506532265D+01
GDOTG( 2):    6.3499694136D+01


 THE  2 Y VECTORS:
  Y( 1):
            2.0000000000D-04    0.0
  Y( 2):                                                           (3.20c)
            0.0                 4.2314049580D-04

 THE  2 RESIDUAL VECTORS:
  R( 1):
            1.0000000000D-04    0.0
  R( 2):                                                           (3.20d)
            0.0                 3.2314049580D-04

 THE  2 VALUES OF THE SCALARS TAU(J) AND MATRICES B(J):

TAU( 1):   -5.0000000000D+07

  B( 1):  ROW( 1)
            1.0000000000D-03    0.0

  B( 1):  ROW( 2)
            0.0                 0.0                                (3.20e)
TAU( 2):   -7.3134790628D+06

  B( 2):  ROW( 1)
            0.0                 0.0

  B( 2):  ROW( 2)
            0.0                 1.0441978003D-07
```

Figure 3.3. PVM Algorithm Results for Example Problem (Page 3 of 8)

```
 UPDATED ITERATE OF THE METRIC MATRIX V(N+1):
ROW( 1):
          5.0000000000D-01    0.0                                    } (3.20f)
ROW( 2):
          0.0                 2.3632812504D-01

 NEWTON-LIKE SEARCH DIRECTION VECTOR:
    SN:
         -3.0000000000D+00    1.2394082510D+00                       } (3.20g)

                  ****************************************************************
                  *                                                              *
                  *          SINGLE VARIABLE SEARCH REPORT -- COGGINS METHOD     *
                  *                                                              *
                  ****************************************************************

                                    INITIALIZATION VALUES

      THE INITIAL VALUE OF THE INDEPENDENT VARIABLE ALPHA:          ALPHA(1) =     0.0
      THE INITIAL VALUE OF THE STEPSIZE                :          DELTA ALPHA =  2.0000000000D+00
      THE CONVERGENCE LIMIT FOR THE INDEPENDENT VARIABLE :         EPSILON =      1.0000000000D-04
      THE NUMBER OF PARAMETERS IN THE SOLVE-FOR VECTOR X : NO. OF PARAMETERS =    2

**********          BRACKETING THE FUNCTION MINIMUM -- DAVIES, SWANN AND CAMPEY ALGORITHM          **********

      ALPHA              F(XN + ALPHA * SN)      DELTA ALPHA
      0.0                1.2250000230D+01        2.0000000000D+00

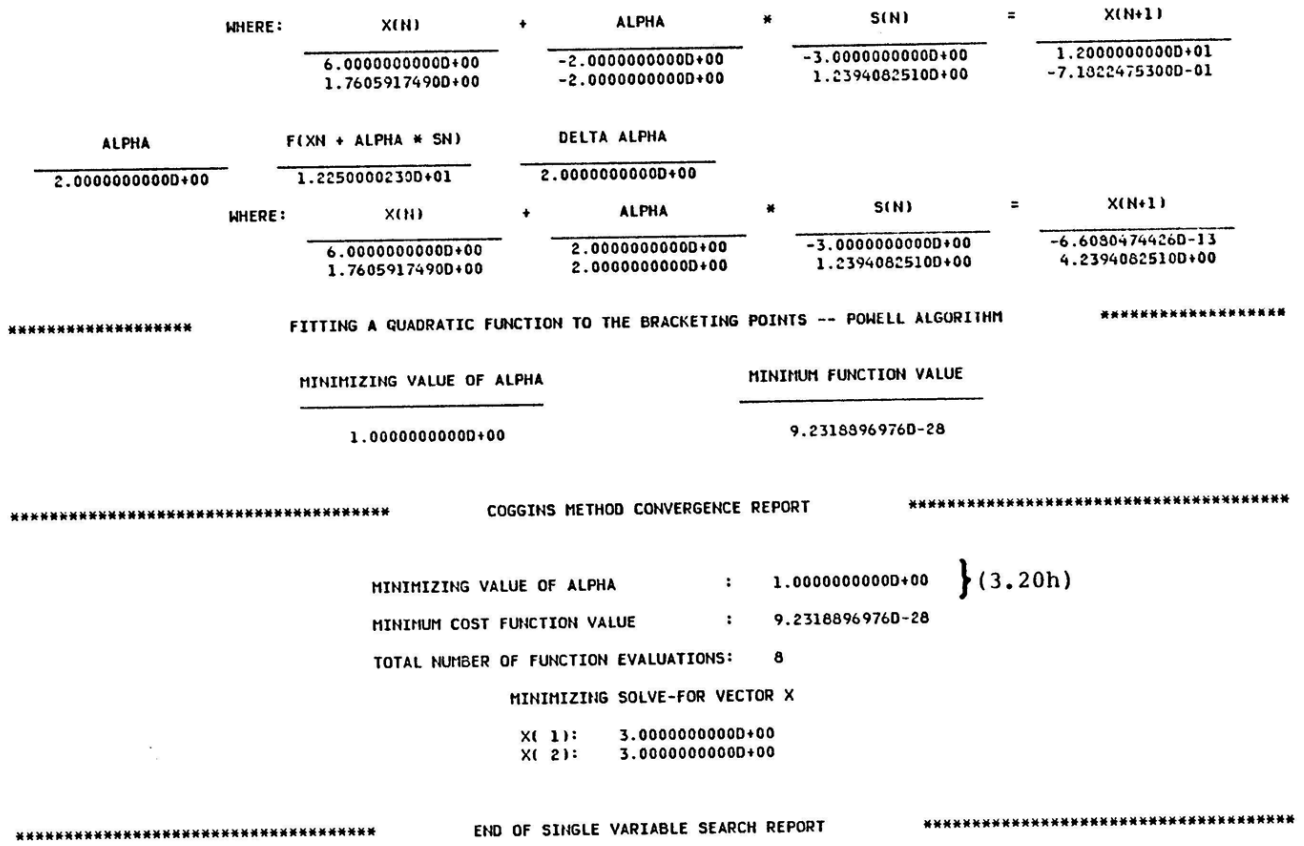
      WHERE:    X(N)                  +  ALPHA          *  S(N)                  =  X(N+1)
                6.0000000000D+00         0.0               -3.0000000000D+00      6.0000000000D+00
                1.7605917490D+00         0.0               1.2394082510D+00       1.7605917490D+00

      ALPHA              F(XN + ALPHA * SN)      DELTA ALPHA
      2.0000000000D+00   1.2250000230D+01        2.0000000000D+00

      WHERE:    X(N)                  +  ALPHA          *  S(N)                  =  X(N+1)
                6.0000000000D+00         2.0000000000D+00  -3.0000000000D+00      -6.6080474426D-13
                1.7605917490D+00         2.0000000000D+00  1.2394082510D+00       4.2394082510D+00

      ALPHA              F(XN + ALPHA * SN)      DELTA ALPHA
      -2.0000000000D+00  1.1025000207D+02        -2.0000000000D+00
```

Figure 3.3. PVM Algorithm Results for Example Problem (Page 4 of 8)



(3.20h)

Figure 3.3. PVM Algorithm Results for Example Problem (Page 5 of 8)

```
**********************************************************************************************************
**********                          ITERATION NUMBER    1                          **********
**********************************************************************************************************

CURRENT SOLVE-FOR VECTOR:
     X:
          3.0000000000D+00     3.0000000000D+00

COST FUNCTIONAL EVALUATED AT THE CURRENT ITERATE OF THE SOLVE-FOR VECTOR:
  J(X):     9.2318996976D-28

GRADIENT OF THE COST FUNCTIONAL EVALUATED AT THE CURRENT ITERATE OF THE SOLVE-FOR VECTOR:
  G(X):
          3.1974423109D-14     7.5164851374D-14

SQUARE OF THE MAGNITUDE OF THE CURRENT GRADIENT VECTOR:
G DOT G:    6.6721166153D-27

CURRENT ITERATE OF THE METRIC MATRIX V(N):
ROW( 1):
          5.0000000000D-01     0.0                    } (3.20i)
ROW( 2):
          0.0                  2.3632812504D-01

     NUMBER OF FUNCTION EVALUATIONS (SEQUENTIAL) IN THE UNIVARIATE SEARCH METHOD FOR ITERATION  1:   8.

          NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EXCLUDING UNIVARIATE SEARCH)
          IN SEQUENTIAL EXECUTION OF PVM METHOD FOR ITERATION  1:   5.

          NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (INCLUDING UNIVARIATE SEARCH)
          IN SEQUENTIAL EXECUTION OF PVM METHOD FOR ITERATION  1:  13.

          EFFECTIVE NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EXCLUDING UNIVARIATE
          SEARCH) IN PARALLEL EXECUTION OF PVM METHOD FOR ITERATION  1:   3.

          EFFECTIVE NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (INCLUDING UNIVARIATE
          SEARCH) IN PARALLEL EXECUTION OF PVM METHOD FOR ITERATION  1:  11.
```

Figure 3.3. PVM Algorithm Results for Example Problem (Page 6 of 8)



Figure 3.3. PVM Algorithm Results for Example Problem (Page 7 of 8)

PARALLEL VARIABLE METRIC (PVM) CONVERGENCE REPORT

COMPONENT	OPTIMUM SOLVE-FOR VECTOR	OPTIMUM GRADIENT VECTOR	OPTIMUM SEARCH VECTOR
1	3.0000000000D+00	3.1974423109D-14	-3.0000000000D+00
2	3.0000000000D+00	7.5164851374D-14	1.2394082510D+00

(3.20j)

THE MINIMUM VALUE OF THE ASSOCIATED COST FUNCTIONAL AT CONVERGENCE: 9.2318896976D-28

THE NUMBER OF ITERATIONS TO CONVERGENCE: 1

TOTAL NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EXCLUDING UNIVARIATE SEARCH) IN 1 SEQUENTIAL ITERATIONS OF THE PVM METHOD: 7

TOTAL NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (INCLUDING UNIVARIATE SEARCH) IN 1 SEQUENTIAL ITERATIONS OF THE PVM METHOD: 15

EFFECTIVE TOTAL NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EXCLUDING UNIVARIATE SEARCH) IN 1 PARALLEL ITERATIONS OF THE PVM METHOD: 5

EFFECTIVE TOTAL NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (INCLUDING UNIVARIATE SEARCH) IN 1 PARALLEL ITERATIONS OF THE PVM METHOD: 13

THE SPEED-UP RATIO (SR) IS DEFINED AS THE NUMBER OF EVALUATIONS IN THE SEQUENTIAL MODE DIVIDED BY THE NUMBER OF EVALUATIONS IN THE PARALLEL MODE.

EXCLUDING UNIVARIATE SEARCH: SR = 1.4000000000D+00 TO 1.

INCLUDING UNIVARIATE SEARCH: SR = 1.1538461538D+00 TO 1.

END OF PARALLEL VARIABLE METRIC FUNCTION MINIMIZATION REPORT

Figure 3.3. PVM Algorithm Results for Example Problem (Page 8 of 8)

(3.20d) and (3.20e) in Figure 3.3. Because of the simple form of the cost functional (i.e., no cross-product terms of the form $x_1 x_2$) the characteristics of the correction matrices are clearly indicated by the results of Equation (3.20d). The two B matrices are independent and are also clearly of rank one.

The metric matrix $V^{(0)}$ is now serially updated via Equation (3.11) and the search direction vector $\vec{s}_1$ is computed from Equation (3.15). These values are given in Equations (3.20f) and (3.20g). As shown in Equation (3.20h), the one-dimensional search yields the expected stepsize of $\alpha_1 = 1$ and the converged values of the solve-for vector, the cost functional and its gradient are given in Equation (3.20j).

Finally, it is important to note that the Hessian matrix for Equation (3.17) is given by

$$G = \left(\frac{\partial \vec{g}(\vec{x})}{\partial \vec{x}} \right)$$

$$= \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{pmatrix} \qquad (3.21a)$$

where $\vec{g}(\vec{x})$ is given by Equation (3.18).

$$\therefore \quad G = \begin{pmatrix} 2 & 0 \\ 0 & 4.2314 \end{pmatrix} \tag{3.21b}$$

The inverse of this diagonal Hessian matrix G^{-1} is therefore seen to be equal to the metric matrix given by Equation (3.20i), as is expected from Straeter's development.

The example problem was, of course, greatly simplified, but the basic algorithm functions are amply demonstrated. Straeter proved that for a strictly convex cost functional J that is twice differentiable and has bounded second partial derivatives on the finite-dimensional Hilbert space H, the iterates generated by the PVM method converge to the global minimum of J if the iterates of the metric matrix are uniformly positive definite. It would therefore seem possible to apply this method to the problem of satellite orbit estimation, in a manner similar to the approach of Reference 4. The following section discusses the approach taken to apply the PVM method to the orbit estimation problem, the differences between this approach and that of Adimurthy and Joy, and the problems associated with this particular formulation.

3.2 Application to Orbit Estimation Problem

The results of Straeter coupled with the results of Adimurthy and Joy indicate that the Parallel Variable Metric nonlinear function minimization technique could provide an accurate orbit estimation method with excellent performance characteristics in a parallel processing environment. To apply this method to the problem of satellite orbit determination, the problem will be posed in the manner shown below. Much of the notation here is similar to that in Chapter 8 of Reference 46 and of Reference 4. It is assumed that there exists an m-dimensional vector $\vec{y}$ made up of m scalar observations of the satellite motion. These observations are generally considered to be a vector-valued function of a p-dimensional vector of solve-for parameters $\vec{x}$. The observations are also assumed to be corrupted by a random noise component $\vec{n}$. The observation equation is therefore given by

$$\vec{y} = \vec{f}(\vec{x}) + \vec{n} \tag{3.22}$$

where $\vec{f}$ is the mathematical observation model and $\vec{n}$ is a vector whose components are random noise values.

Given the functional form of the observation models $\vec{f}$, the actual measurements $\vec{y}$ and the assumed statistical properties of $\vec{n}$, it is desired to achieve a "best estimate" of the solve-for parameters $\vec{x}$. In the present development, the term "best estimate" will be taken to be that value of the solve-for vector which minimizes the weighted sum of the squares of the observation residuals $\vec{e} = [\vec{y} - \vec{f}(\vec{x})]$. Therefore, the cost functional to be minimized by the PVM method will be given by

$$J(\vec{x}) = [\vec{y} - \vec{f}(\vec{x})]^T W[\vec{y} - f(\vec{x})] \qquad (3.23)$$

where W is the weighting matrix and is assumed by be symmetric.

The PVM method also requires computation of the gradient of the cost functional with respect to the solve-for parameters. The gradient, expressed as a vector in parameter space, may be found as follows

$$\begin{aligned} \vec{g}^T(\vec{x}) &= \frac{\partial J(\vec{x})}{\partial \vec{x}} \\ &= \frac{\partial}{\partial \vec{x}} \{[\vec{y} - \vec{f}(\vec{x})]^T W [\vec{y} - \vec{f}(\vec{x})]\} \\ &= \frac{\partial}{\partial \vec{x}} \{[\vec{y}^T - \vec{f}^T(\vec{x})] W [\vec{y} - \vec{f}(\vec{x})]\} \end{aligned}$$

$$\vec{g}^T(\vec{x}) = \frac{\partial}{\partial \vec{x}} \{\vec{y}^T W \vec{y} - \vec{y}^T W \vec{f}(\vec{x}) - \vec{f}^T(\vec{x}) W \vec{y} + \vec{f}^T(\vec{x}) W \vec{f}(\vec{x})\}$$

But $\vec{y}^T W \vec{f}(\vec{x})$ is a scalar and therefore equal to its transpose, such that $\vec{y}^T W \vec{f}(\vec{x}) = \vec{f}^T(\vec{x}) W \vec{y}$. Note: $W = W^T$ since the weighting matrix is assumed to be symmetric. Therefore

$$\vec{g}^T(\vec{x}) = \frac{\partial}{\partial \vec{x}} \{\vec{y}^T W\vec{y} - 2\vec{y} W\vec{f}(\vec{x}) + \vec{f}^T(\vec{x}) W\vec{f}(\vec{x})\}$$

$$= -2\vec{y}^T W \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} + 2\vec{f}^T(\vec{x}) W \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$$

$$\vec{g}^T(\vec{x}) = -2[\vec{y} - \vec{f}(\vec{x})]^T W \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$$

Thus, the gradient is given by

$$\vec{g}(\vec{x}) = -2 \left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right)^T W [\vec{y} - \vec{f}(\vec{x})] \qquad (3.24)$$

where $\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$ = mxp matrix of partial derivatives of the observation models with respect to the solve-for parameters. This matrix is often called the matrix of observation partials.

A standard technique for solving the problem posed above involves linearizing the observation equations by a first order Taylor Series expansion, such that

$$\vec{f}(\vec{x}) = \vec{f}(\vec{x}_o) + \left.\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}\right|_{\vec{x} = \vec{x}_o} \Delta\vec{x} + \text{H.O.T.}$$

where $\vec{x}_o \triangleq$ *a priori* value of the solve-for vector

$$\Delta\vec{x} \triangleq \vec{x} - \vec{x}_o$$

Neglecting higher order terms in the Taylor series and defining

$$F = \left.\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}\right|_{\vec{x} = \vec{x}_o} \tag{3.25}$$

The expansion becomes

$$\vec{f}(\vec{x}) = \vec{f}(\vec{x}_o) + F\,\Delta\vec{x} \tag{3.26}$$

Substituting this into the nonlinear regression of Equation (3.22) yields the equation of the linearized observation vector

$$\Delta\vec{y} = F\,\Delta\vec{x} + \vec{n} \tag{3.27}$$

where $\Delta\vec{y} \triangleq \vec{y} - \vec{f}(\vec{x}_o)$

A necessary condition for the cost functional to be minimized is that the right hand side of Equation (3.24) equal the zero vector. Substituting the results of Equations (3.26) and (3.27) into the right hand side of Equation (3.24) and setting it equal to zero yields

$$\vec{0} = -2[\vec{y} - \vec{f}(\vec{x})]^T W \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$$

$$= -2[\vec{y} - \vec{f}(\vec{x}_o) - F\,\Delta\vec{x}]^T W F$$

$$\therefore \quad -2[\Delta\vec{y} - F\,\Delta\vec{x}]^T W F = \vec{0}$$

This equation is easily solved for $\Delta\vec{x}$ as

$$\hat{\Delta\vec{x}} = (F^T W F)^{-1} F^T W \Delta\vec{y} \qquad (3.28)$$

where $\hat{\Delta\vec{x}} \triangleq$ the optimal $\Delta\vec{x}$

and $F^T W F \triangleq$ normal matrix

Then the "best estimate" of the solve-for vector $\vec{x}$ is given by

$$\hat{\vec{x}} = \vec{x}_o + \Delta\hat{\vec{x}} \qquad (3.29)$$

The above method is the classical differential correction method and has certain disadvantages. The primary disadvantage, as has been previously stated, is that the linearization of Equations (3.26) and (3.27) may not remain valid if the *a priori* solve-for vector is not sufficiently close to the true minimizing solution. Then the method may not converge or may converge to improper values.

The PVM application proposed here, as with the other nonlinear techniques discussed in Reference 4, preserve the full nonlinear character of the observation equations. In applying the method, several major assumptions are made. First, it is assumed that the actual observations are equal to a known, nonlinear, vector-valued function $\vec{f}$ of the p solve-for parameters $\vec{x}$ plus an additive random noise component $\vec{n}$. This is the same assumption that yielded Equation (3.22), and so

$$\vec{y} = \vec{f}(\vec{x}) + \vec{n}$$

Hence

$$\vec{n} = \vec{y} - \vec{f}(\vec{x}) = \vec{e} \tag{3.30}$$

where $\vec{e} \triangleq$ vector of observation errors or observation residuals

Second, it is assumed that the observation errors are Gaussian random variables with zero mean and with a known moment matrix N. The assumption of zero mean indicates that the expected value of each component of $\vec{n}$ is zero, that is

$$E(n_i) = \bar{n}_i = 0 \tag{3.31}$$

for $i = 1,2,\ldots,m$

where $E(n_i) \triangleq$ expected value operator

$$= \int_{-\infty}^{\infty} n_i \, p(n_i) \, dn_i$$

with $p(n_i)$ the probability density function for random variable n_i

Therefore

$$E(\vec{n}) = \bar{\vec{n}} = \vec{0} \tag{3.32}$$

where $\vec{n} = \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{pmatrix}$ = random vector of observation errors

From Reference 28, the m-dimensional Gaussian distribution of these m random variables is given by

$$p(n_1,n_2,\ldots,n_m) = \frac{1}{\sqrt{(2\pi)^m |N|}} \exp\left(-\frac{1}{2}\,\vec{n}^T\, N^{-1}\,\vec{n}\right) \tag{3.33}$$

where N represents the moment matrix or correlation matrix of the distribution. This matrix is of the form

$$N = \overline{\vec{n}\,\vec{n}^T} = \begin{pmatrix} \overline{n_1^2} & \overline{n_1 n_2} & \cdots & \overline{n_1 n_m} \\ \overline{n_2 n_1} & \overline{n_2^3} & \cdots & \overline{n_2 n_m} \\ \vdots & \vdots & \cdots & \vdots \\ \overline{n_m n_1} & \overline{n_m n_2} & \cdots & \overline{n_m^2} \end{pmatrix} \tag{3.34}$$

Third, it is assumed that the observation errors are statistically independent. The condition that two random variables are statistically independent means that their

joint probability density function is the product of their respective probability density functions, that is

$$p(n_1,n_2) = p(n_1)\, p(n_2)$$

Hence, the expected value of the product of any two statistically independent components, say n_1 and n_2, of the random vector $\vec{n}$ is given by

$$\overline{n_1 n_2} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} n_1 n_2\, p(n_1,n_2)\, dn_1\, dn_2$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} n_1 n_2\, p(n_1)\, p(n_2)\, dn_1\, dn_2$$

$$= \int_{-\infty}^{\infty} n_1\, p(n_1)\, dn_1 \int_{-\infty}^{\infty} n_2\, p(n_2)\, dn_2$$

$$\therefore \quad \overline{n_1 n_2} = \bar{n}_1\, \bar{n}_2 \qquad (3.35)$$

for statistically independent random variables

But, since the components of the random vector $\vec{n}$ were already assumed to have zero mean, $\bar{n}_1 = \bar{n}_2 = 0 = \overline{n_1 n_2}$. Thus, the moment matrix of the observation errors is diagonal of the form

$$N = \begin{pmatrix} \overline{n_1^2} & & \underline{0} \\ & \overline{n_2^2} & \\ \underline{0} & & \ddots \; \overline{n_m^2} \end{pmatrix} \tag{3.36}$$

The variance of a random variable, or its mean squared value about its mean, is given by

$$\sigma_i^2 = \overline{(n_i - \bar{n}_i)^2}$$

$$= \overline{(n_i^2 - 2n_i\bar{n}_i + \bar{n}_i^2)}$$

$$= \overline{n_i^2} - 2\bar{n}_i\bar{n}_i + \bar{n}_i^2$$

$$\therefore \quad \sigma_i^2 = \overline{n_i^2} - \bar{n}_i^2$$

Again, the assumption that the errors have zero mean implies $\bar{n}_i^2 = 0$, such that

$$\overline{n_i^2} = \sigma_i^2 \quad , \quad (i = 1,2,\ldots,m) \tag{3.37}$$

Thus, the moment matrix becomes

$$N = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \underline{0} \\ & & \ddots & \\ \underline{0} & & & \sigma_m^2 \end{pmatrix} \tag{3.38}$$

which is the covariance matrix of the observation errors. By assuming that the weighting matrix in Equation (3.23) is equal to the inverse of the known covariance matrix of the observation errors, the cost functional can be written as

$$J(\vec{x}) = [\vec{y} - \vec{f}(\vec{x})]^T N^{-1} [\vec{y} - \vec{f}(\vec{x})] \tag{3.39}$$

Similarly, the gradient of Equation (3.24) is given by

$$\vec{g}(\vec{x}) = -2 \left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}\right)^T N^{-1} [\vec{y} - \vec{f}(\vec{x})] \tag{3.40}$$

Figure 3.4 summarizes the definition of the cost functional and its gradient vector used in the appliction of the PVM method to the orbit estimation problem.

The method to this point has been essentially that proposed by Adimurthy and Joy (4). They used the maximum likelihood estimator, which is a special case of the weight-

Figure 3.4. Definition of Cost Functional and Gradient Vector in PVM Orbit Estimation Algorithm

Problem: Find $\vec{x}^*$ which minimizes cost function $J(\vec{x})$

Given: $$J(\vec{x}) = [\vec{y} - \vec{f}(\vec{x})]^T N_{mm}^{-1} [\vec{y} - \vec{f}(\vec{x})]$$

where
- J ∿ scalar functional of the p-dimensional vector $\vec{x}$
- $\vec{y}$ ∿ m-dimensional vector of observed scalar quantities
- $\vec{f}$ ∿ m-dimensional vector function of the p-dimensional vector $\vec{x}$, which provides the predicted observable scalar quantities
- $\vec{x}$ ∿ p-dimensional vector of solve-for parameters to be estimated, given the observations $\vec{y}$ and the requirement to minimize $J(\vec{x})$
- N_{mm}^{-1} ∿ weighting matrix, which is also the information matrix of the measurement errors

N_{mm} = covariance matrix of the measurement errors

$$N_{mm} = \begin{pmatrix} \sigma_1^2 & & \underline{0} \\ & \sigma_2^2 & \\ \underline{0} & & \ddots \\ & & & \sigma_m^2 \end{pmatrix}$$

$$\vec{g}^T(\vec{x}) = -2[\vec{y} - \vec{f}(\vec{x})]^T N_{mm}^{-1} \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$$

where $\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$ ∿ mxp matrix (Jacobian) of observation partials

ed least squares estimator when the measurement errors are Gaussian random variables, but with the inverse of the measurement error covariance matrix used as the weighting matrix. The implementation presented here differs radically from that of Adimurthy and Joy in the choice of the deterministic model for the satellite motion. Adimurthy and Joy used a General Perturbation type of dynamic model to describe the satellite motion. The force models used in their perturbation dynamics are therefore somewhat limited. However, the advantage of the General Perturbation formulation is that explicit formulas exist to directly evaluate the spacecraft inertial position and velocity, used to compute the predicted observable quantities, and to evaluate the observation partials for the gradient vector computation. The implementation presented here uses a Special Perturbation formulation for the satellite dynamical model and uses the Picard-Chebyshev integration technique to propagate the orbit dynamics. This allows the inclusion of more rigorous perturbation models in the satellite theory, but requires a full integration of the trajectory each time the predicted observations are needed and also requires special treatment for the computation of the gradient vector.

3.3 Cost Functional and Gradient Vector Evaluation

In the evaluation of the weighted least squares cost functional of Equation (3.39) and its corresponding gradient vector, given by Equation (3.40), it is required to compute the m-dimensional vector of observation residuals given by Equation (3.30) as

$$\vec{e} = \vec{y} - \vec{f}(\vec{x})$$

The vector $\vec{y}$ in the above equation is the m-dimensional vector of the actual scalar observations of the satellite motion. The vector $\vec{f}$ is the m-dimensional vector of predicted observations computed from the appropriate observation models and based on the solve-for parameter vector $\vec{x}$. When a General Perturbation satellite theory is used, the predicted observations are computed directly from the explicit General Perturbation formulae at the observation times only. A Special Perturbation satellite theory requires a full numerical integration of the equations of motion from the given epoch conditions through the complete observation span of interest. The improved accuracy of the Special Perturbation formulation, due to the more extensive force models involved, is often necessary for the extremely high precision orbit estimation required in many applications.

The gradient computation also requires the computation of the observation partial derivatives. These computations usually involve the evaluation of the state partial derivatives, that is, the partial derivatives of the current position and velocity vectors with respect to the epoch position and velocity vectors. The solution of the variational equations required for these computations can be accomplished by numerical integration or by an analytical solution for a simplified model. In the implementation formulated here, the Picard-Chebyshev orbit propagation algorithm is not currently capable of numerically integrating the variational equations. Therefore, a set of analytical formulae for the computation of the state partial derivatives is proposed. These analytical formulae incorporate the secular rates in the argument of periapsis and the longitude of the ascending node due to the dominant J_2 zonal harmonic in the gravitational potential. The following sections will discuss the assumptions used in the present implementation and will describe the formulae for solution of the analytical state partial derivatives. Additionally, the PVM method requires the parallel evaluation of the cost functional at the p values of the solve-for vector given by Equation (3.5). A direct integration method and a linearized approximation method for this parallel cost functional evaluation are described in the final two sections.

3.3.1 Assumptions in Orbit Estimation Formulation

Several assumptions were made in the application of the PVM method to the orbit estimation problem. First, it was assumed that the solve-for vector would be limited to the following three possible sets of dynamical parameters:

1. Epoch inertial Cartesian position and velocity vectors
2. Epoch equinoctial elements
3. Epoch Keplerian elements.

That, is the solve-for vector in this implementation will be given by

$$\vec{x} = \begin{pmatrix} r_{xo} \\ r_{yo} \\ r_{zo} \\ \dot{r}_{xo} \\ \dot{r}_{yo} \\ \dot{r}_{zo} \end{pmatrix} \text{ or } \begin{pmatrix} a_o \\ h_o \\ k_o \\ p_o \\ q_o \\ \lambda_o \end{pmatrix} \text{ or } \begin{pmatrix} a_o \\ e_o \\ i_o \\ \omega_o \\ \Omega_o \\ M_o \end{pmatrix} \qquad (3.41)$$

This assumption is somewhat restrictive in that many operational OD systems allow much more extensive solve-for vectors in their estimation procedures (46, 122). Additionally, it has been shown by several authors (66, 82, 140) that inclusion of drag-related parameters, such as the ballistic coefficient or the coefficient of drag, in the estimation solve-for vector often produces a more accurate state estimate. However, it is felt that including only the epoch state vector components in the estimation process is adequate to demonstrate the accuracy and performance of the PVM method.

The second assumption is that each observed quantity is one of the scalar components of the inertial Cartesian position or velocity vectors at the observation times. This approach was taken from Battin, et al. (30) and Edwards (70) in their work on the epoch state navigation filter. In this case, the mathematical observation models become the dynamical equations of motion themselves. Therefore, the observation partial derivatives are exactly the state partial derivatives and the necessary computations are considerably simplified. For example, if the components of the solve-for vector are the epoch position and velocity components and at the ith observation time the y-component of position is the observable quantity, then the ith component in the observation residual vector is given by Equation (3.30) as

$$e_i = r_y(t_i) - f_i(\vec{x}, t_i)$$

where f_i = y-component of position computed from the equations of motion at time t_i, based on the current solve-for vector $\vec{x}$

The ith row in the mx6 matrix of observation partial derivatives is then given by

$$\frac{\partial f_i(\vec{x})}{\partial \vec{x}} = \frac{\partial r_y(t_i)}{\partial \vec{r}_o, \dot{\vec{r}}_o}$$

which is the second row of the 6x6 matrix of state partial derivatives evaluated at time t_i. This assumption bypasses the difficult problem of observation modelling, which is an extensive portion of the software in most operational OD systems. However, this assumption is not considered too restrictive in demonstrating the fundamental applicability of the PVM method.

Finally, it is assumed that the observations are corrupted by an additive noise component with certain characteristics. It is assumed that the observation noise is normally distributed, has zero mean and has a known standard

deviation. It is also assumed that different components of the observation noise vector are statistically independent. These assumptions have been previously stated in Section 3.2 and are common to most operational orbit estimation algorithms. This assumption is not considered restrictive at all in the demonstration of the feasibility of the PVM method.

3.3.2 Analytical State Partial Derivatives

The necessary state partial derivatives may be computed from a chain rule differentiation as

$$\left(\frac{\partial \vec{r},\dot{\vec{r}}}{\partial \vec{r}_o,\dot{\vec{r}}_o}\right) = \left(\frac{\partial \vec{r},\dot{\vec{r}}}{\partial \vec{a}}\right)\left(\frac{\partial \vec{a}}{\partial \vec{a}_o}\right)\left(\frac{\partial \vec{a}_o}{\partial \vec{r}_o,\dot{\vec{r}}_o}\right) \qquad (3.42)$$

where $\vec{a}$ is a vector of orbital elements. In this implementation, the primary orbital element set is taken to be the equinoctial elements described by Broucke and Cefola (38). This choice was made because the associated matrices of Poission and Lagrange brackets are free of singularities for zero eccentricity and zero or ninety degree inclination cases. Appendix A provides a brief description of this element set.

3.3.2.1 Partial Derivatives of Position with Respect to Elements

Derivations of $\frac{\partial \vec{r}}{\partial \vec{a}}$ for the equinoctial elements are presented by McClain (110). These expressions include the retrograde factor and are used in this implementation with one exception. McClain computes $\frac{\partial \vec{r}}{\partial \vec{a}}$ for a formulation of six slow variables, that is, the sixth variable is the mean longitude at epoch λ_o. The present implementation uses a formulation of five slow elements and one rapidly varying phase element, which is the mean longitude. This results in a slightly different form for the partial derivative of the position vector with respect to the semi-major axis. Portions of that development will be reproduced here.

As is shown in Appendix A and References 38, 82 and 110, the position and velocity vectors can be expressed in the equinoctial orbit frame as

$$\vec{r} = X \hat{f} + Y \hat{g} \tag{3.43a}$$

and

$$\dot{\vec{r}} = \dot{X} \hat{f} + \dot{Y} \hat{g} \tag{3.43b}$$

In these expressions, the unit vectors $\hat{f}$, $\hat{g}$ and $\hat{w}$ define the equinoctial orbit plane coordinate frame, as described in Appendix A. In terms of the equinoctial elements, they may be expressed as follows, with respect to an inertial reference frame

$$\hat{f} = \frac{1}{1 + p^2 + q^2} \begin{pmatrix} 1 - p^2 + q^2 \\ 2pq \\ -2pI \end{pmatrix} \tag{3.44a}$$

$$\hat{g} = \frac{1}{1 + p^2 + q^2} \begin{pmatrix} 2pqI \\ (1 + p^2 - q^2)I \\ 2q \end{pmatrix} \tag{3.44b}$$

$$\hat{w} = \frac{1}{1 + p^2 + q^2} \begin{pmatrix} 2p \\ -2q \\ (1 - p^2 - q^2)I \end{pmatrix} \tag{3.44c}$$

From the geometry described in Appendix A, the position coordinates X and Y are given by

$$X = r \cos L \tag{3.45a}$$

and

$$Y = r \sin L \tag{3.45b}$$

where L is the true longitude. In these expressions, McClain shows the magnitude of the position vector is given by

$$r = \frac{a(1 - h^2 - k^2)}{1 + k \cos L + h \sin L} \tag{3.45c}$$

Thus, in the expression for the position vector given by Equation (3.43a), the components X and Y depend only on elements a, h, k and λ while the unit vectors $\hat{f}$ and $\hat{g}$ depend only on elements p and q. The partial derivatives of the position vector with respect to the equinoctial elements can therefore be found by algebraic manipulation.

Cefola, et al. (49) have found a rather elegant formulation for the partial derivatives of $\frac{\partial X}{\partial h}$, $\frac{\partial Y}{\partial h}$, $\frac{\partial X}{\partial k}$, and $\frac{\partial Y}{\partial k}$, which arise in the position partial derivatives. These expressions are also given in McClain (110) and are listed below.

$$\frac{\partial X}{\partial h} = -\frac{k\beta \dot{X}}{n} + \frac{a}{G} Y \dot{Y} \tag{3.46a}$$

$$\frac{\partial X}{\partial k} = \frac{h\beta \dot{X}}{n} + \frac{a}{G} (\dot{X} Y - G) \tag{3.46b}$$

$$\frac{\partial Y}{\partial h} = -\frac{k\beta \dot{Y}}{n} - \frac{a}{G} (X \dot{Y} + G) \tag{3.46c}$$

$$\frac{\partial Y}{\partial k} = \frac{h\beta \dot{Y}}{n} - \frac{a}{G} X \dot{X} \tag{3.46d}$$

where

$$\beta = \frac{1}{1 + \sqrt{1 - h^2 - k^2}} \tag{3.47a}$$

$$G = na^2 \sqrt{1 - h^2 - k^2} \tag{3.47b}$$

$$= \text{magnitude of orbital angular momentum}$$

Using the above information, the final form of the position partial derivatives are given in Table 3.1.

3.3.2.2 Partial Derivatives of Velocity with Respect to Elements

The partial derivatives of the velocity vector may be found by differentiating the results of Equation (3.49) with respect to time. That is, assuming that the time derivative and the partial derivative operations may be interchanged

$$\frac{d}{dt}\left(\frac{\partial \vec{r}}{\partial \vec{a}}\right) = \frac{\partial}{\partial \vec{a}}\left(\frac{d\vec{r}}{dt}\right) = \frac{\partial \dot{\vec{r}}}{\partial \vec{a}} \tag{3.48}$$

Table 3.1
Partial Derivatives of Position
with Respect to Equinoctial Elements

$$\frac{\partial \vec{r}}{\partial a} = \frac{\vec{r}}{a} \tag{3.49a}$$

$$\frac{\partial \vec{r}}{\partial h} = \frac{\partial X}{\partial h}\hat{f} + \frac{\partial Y}{\partial h}\hat{g} \tag{3.49b}$$

$$\frac{\partial \vec{r}}{\partial k} = \frac{\partial X}{\partial k}\hat{f} + \frac{\partial Y}{\partial k}\hat{g} \tag{3.49c}$$

$$\frac{\partial \vec{r}}{\partial p} = \frac{2}{1 + p^2 + q^2}\left[q(Y\hat{f} - X\hat{g})I - X\hat{w}\right] \tag{3.49d}$$

$$\frac{\partial \vec{r}}{\partial q} = \frac{2}{1 + p^2 + q^2}\left[p(X\hat{g} - Y\hat{f}) + Y\hat{w}\right]I \tag{3.49e}$$

$$\frac{\partial \vec{r}}{\partial \lambda} = \frac{\vec{r}}{n} \tag{3.49f}$$

with I = retrograde factor

These equations will be evaluated at a single point in time and hence the position and velocity are related to the orbital elements by simple two-body relations. Thus, the elements and the unit vectors are taken to be constants and the equation of motion is taken to be

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r}$$

Direct application of this approach to the results of Equation (3.49) yields the velocity partial derivatives given in Table 3.2.

Equation (3.50a) has a slightly different form because of the element set used. In the six slow element set used by Broucke and Cefola (38), McClain (110) and Long (105), Equation (3.49a) is given by

$$\frac{\partial \vec{r}}{\partial a} = \frac{1}{a}\left[\vec{r} - \frac{3}{2}\dot{\vec{r}}\,(t - t_o)\right]$$

In the five slow elements/one fast element set used here, $t = t_o$ and the final term in the above equation vanishes. Taking the time derivative of the above equation yields

Table 3.2
Partial Derivatives of Velocity
with Respect to Equinoctial Elements

$$\frac{\partial \dot{\vec{r}}}{\partial a} = -\frac{\dot{\vec{r}}}{2a} \tag{3.50a}$$

$$\frac{\partial \dot{\vec{r}}}{\partial h} = \frac{\partial \dot{X}}{\partial h}\hat{f} + \frac{\partial \dot{Y}}{\partial h}\hat{g} \tag{3.50b}$$

$$\frac{\partial \dot{\vec{r}}}{\partial k} = \frac{\partial \dot{X}}{\partial k}\hat{f} + \frac{\partial \dot{Y}}{\partial k}\hat{g} \tag{3.50c}$$

$$\frac{\partial \dot{\vec{r}}}{\partial p} = \frac{2}{1 + p^2 + q^2}\left[q(\dot{\hat{Y}}\hat{f} - \dot{\hat{X}}\hat{g})I - \dot{\hat{X}}\hat{w}\right] \tag{3.50d}$$

$$\frac{\partial \dot{\vec{r}}}{\partial q} = \frac{2}{1 + p^2 + q^2}\left[p(\dot{\hat{X}}\hat{g} - \dot{\hat{Y}}\hat{f}) + \dot{\hat{Y}}\hat{w}\right]I \tag{3.50e}$$

$$\frac{\partial \dot{\vec{r}}}{\partial \lambda} = n\left(\frac{a}{r}\right)^3 \vec{r} \tag{3.50f}$$

where I = retrograde factor

$$\dot{X} = \frac{-na(h + \sin L)}{\sqrt{1 - h^2 - k^2}} \tag{3.51a}$$

$$\dot{Y} = \frac{na(k + \cos L)}{\sqrt{1 - h^2 - k^2}} \tag{3.51b}$$

$$\frac{\partial \dot{\vec{r}}}{\partial a} = \frac{d}{dt}\left(\frac{\partial \vec{r}}{\partial a}\right)$$

$$\frac{\partial \dot{\vec{r}}}{\partial a} = \frac{1}{a}\left[\dot{\vec{r}} - \frac{3}{2}\ddot{\vec{r}}\,(t - t_o) - \frac{3}{2}\dot{\vec{r}}\right]$$

Again, the element set used in this implementation the $(t - t_o)$ term vanishes and the given Equation (3.50a) results.

The equations of Table 3.2 contain factors similar to those of Equation (3.46). By applying a technique similar to that used above, relatively compact forms for $\frac{\partial \dot{X}}{\partial h}$, $\frac{\partial \dot{X}}{\partial k}$, $\frac{\partial \dot{Y}}{\partial h}$ and $\frac{\partial \dot{Y}}{\partial k}$ may be found. Taking the time derivative of both sides of Equation (3.46a) yields

$$\frac{d}{dt}\left(\frac{\partial X}{\partial h}\right) = \frac{d}{dt}\left(-\frac{k\beta \dot{X}}{n} + \frac{a}{G}\,Y\,\dot{Y}\right)$$

$$= -\frac{k\beta}{n}\,\ddot{X} + \frac{a}{G}\,\dot{Y}^2 + \frac{a}{G}\,Y\,\ddot{Y}$$

Since the above equation is a two-body relation evaluated at a specific time t, the time derivative and the partial derivative on the left hand side can be interchanged. This

results in

$$\frac{\partial \dot{X}}{\partial h} = -\frac{k\beta}{n} \ddot{X} + \frac{a}{G} \dot{Y}^2 + \frac{a}{G} Y \ddot{Y} \qquad (3.52)$$

From Equation (3.51a)

$$\ddot{X} = \frac{d}{dt} \dot{X} = \frac{d}{dt} \frac{-na(h + \sin L)}{\sqrt{1 - h^2 - k^2}}$$

$$= \frac{-na \cos L \dot{L}}{\sqrt{1 - h^2 - k^2}}$$

Define $$\gamma = \frac{1}{\sqrt{1 - h^2 - k^2}} \qquad (3.53)$$

$$\ddot{X} = -na \gamma \cos L \dot{L}$$

But $$\dot{L} = \dot{f} = \frac{G}{r^2}$$

where f = true anomaly

G = magnitude of orbital angular momentum

$$= na^2 \sqrt{1 - h^2 - k^2}$$

$$= \frac{na^2}{\gamma}$$

Thus $$\dot{L} = \frac{na^2}{\gamma r^2}$$

and $$\ddot{X} = -\frac{n^2 a^3}{r^3} \cos L$$

But, from Equation (3.45a), $\cos L = \frac{X}{r}$

$$\therefore \quad \ddot{X} = -\frac{n^2 a^3}{2^3} X = -\frac{\mu}{r^3} X$$

Similarly

$$\ddot{Y} = -\frac{\mu}{r^3} Y$$

Substituting these two expressions into Equation (3.52) gives

$$\frac{\partial \dot{X}}{\partial h} = \frac{k\beta\mu}{nr^3} X + \frac{a}{G} \dot{Y}^2 - \frac{\mu a}{Gr^3} Y^2$$

But $$\mu = n^2 a^3$$

$$\therefore \quad \frac{\partial \dot{X}}{\partial h} = \frac{n^2 a^3}{r^3} \left[\frac{k\beta}{n} X - \frac{a}{G} Y^2\right] + \frac{a}{G} \dot{Y}^2$$

After some simplification, the resulting expression is

$$\frac{\partial \dot{X}}{\partial h} = \frac{a}{G} \dot{Y}^2 + \frac{a\gamma}{r^3} (Gk\beta X - naY^2) \qquad (3.54a)$$

In a similar manner, the remaining expressions are given by

$$\frac{\partial \dot{X}}{\partial k} = \frac{a}{G} \ddot{X}Y - \frac{a\gamma}{r^3} (Gh\beta X + naXY) \qquad (3.54b)$$

$$\frac{\partial \dot{Y}}{\partial h} = -\frac{a}{G} \ddot{X}Y + \frac{a\gamma}{r^3} (Gk\beta Y + naXY) \qquad (3.54c)$$

$$\frac{\partial \dot{Y}}{\partial k} = -\frac{a}{G} \dot{X}^2 - \frac{a\gamma}{r^3} (Gh\beta Y - naX^2) \qquad (3.54d)$$

where $\dot{X}$ and $\dot{Y}$ are defined by Equations (3.51a) and (3.51b), β and G are defined by Equations (3.47a) and (3.47b) and γ is defined by Equation (3.53).

3.3.2.3 Partial Derivatives of Elements with Respect to Velocity

Broucke (36) has observed that the partial derivatives of the orbital elements with respect to the velocity vector can be found from the relation

$$\frac{\partial a_i}{\partial \dot{r}_j} = -\sum_{k=1}^{6} (a_i, a_k) \frac{\partial r_j}{\partial a_k} \qquad (3.55)$$

for $i = 1,2,\ldots,6$

and $j = 1,2,3$

The parenthetical terms in the above summation are the Poisson Brackets, given formally by

$$(a_i, a_k) \triangleq \sum_{j=1}^{3} \left[\frac{\partial a_i}{\partial r_j} \frac{\partial a_k}{\partial \dot{r}_j} - \frac{\partial a_i}{\partial \dot{r}_j} \frac{\partial a_k}{\partial r_j} \right] \qquad (3.56)$$

The terms on the right hand side of this expression are those that are being sought. However, it is also possible to determine the matrix of Poisson Brackets by determining the matrix of Lagrange Brackets and directly inverting that matrix. The Lagrange Brackets are given formally by

$$[a_i, a_k] \triangleq \sum_{j=1}^{3} \left[\frac{\partial r_j}{\partial a_i} \frac{\partial \dot{r}_j}{\partial a_k} - \frac{\partial \dot{r}_j}{\partial a_i} \frac{\partial r_j}{\partial a_k} \right] \qquad (3.57)$$

It is seen that the terms on the right hand side of Equation (3.57) have already been found from Equations (3.49) and (3.50). Broucke and Cefola (38) have shown that there are only eleven non-zero Lagrange Brackets and the inverse of the matrix of Lagrange Brackets yields eleven independent non-zero Poisson Brackets. These non-zero Lagrange and Poisson Brackets are included in Tables 3.3 and 3.4. Table 3.3 was taken from Reference 38, but with the retrograde factor included. Table 3.4 may be found in Reference 105 or Reference 110.

Table 3.3

Non-Zero Lagrange Brackets of the Equinoctial Orbital Elements

$$[a,\lambda] = -\frac{na}{2} \qquad [p,q] = -\frac{4na^2 s_3 I}{s_2^2}$$

$$[a,h] = \frac{nak\beta}{2} \qquad [a,p] = \frac{naqs_3 I}{s_2}$$

$$[a,k] = -\frac{nah\beta}{2} \qquad [a,q] = -\frac{naps_3 I}{s_2}$$

$$[h,p] = -\frac{2na^2 hqI}{s_2 s_3} \qquad [k,p] = -\frac{2na^2 kqI}{s_2 s_3}$$

$$[h,q] = \frac{2na^2 hpI}{s_2 s_3} \qquad [k,q] = \frac{2na^2 kpI}{s_2 s_3}$$

$$[h,k] = -\frac{na^2}{s_3}$$

when $s_1 = \frac{1}{na^2}$

$$s_2 = 1 + p^2 + q^2$$

$$s_3 = \sqrt{1 - h^2 - k^2} = \gamma^{-1}$$

$$s_4 = \frac{s_1 s_3}{1 + s_3} = s_1 s_3 \beta = \frac{s_1 \beta}{\gamma}$$

$$s_5 = \frac{s_1 s_2}{2 s_3} = \frac{1}{2} s_1 s_2 \gamma$$

with β defined by Equation (3.47a) and γ defined by Equation (3.53)

Table 3.4

Non-Zero Poisson Brackets

of the Equinoctial Orbital Elements

$$(a,\lambda) = -2as_1 \qquad (h,p) = -kps_5$$

$$(\lambda,h) = -hs_4 \qquad (h,q) = -kqs_5$$

$$(\lambda,k) = -ks_4 \qquad (k,p) = hps_5$$

$$(\lambda,p) = -ps_5 \qquad (k,q) = hqs_5$$

$$(\lambda,q) = -qs_5 \qquad (p,q) = -\frac{1}{2} s_2 s_5 I$$

$$(h,k) = -s_1 s_3$$

where $s_1 = \dfrac{1}{na^2}$

$$s_2 = 1 + p^2 + q^2$$

$$s_3 = \sqrt{1 - h^2 - k^2} = \gamma^{-1}$$

$$s_4 = \frac{s_1 s_3}{1 + s_3} = \frac{s_1 \beta}{\gamma}$$

$$s_5 = \frac{s_1 s_2}{2 s_3} = \frac{1}{2} s_1 s_2 \gamma$$

with β defined by Equation (3.47a) and γ defined by Equation (3.53)

With the Poisson Brackets given in Table 3.4, Equation (3.55) may be used to determine the partial derivatives of the equinoctial elements with respect to the velocity vector. For example, to determine $\partial a/\partial \dot{\vec{r}}$, Equation (3.55) yields

$$\left(\frac{\partial a}{\partial \dot{\vec{r}}}\right)^T = -(a,a)\frac{\partial \vec{r}}{\partial a} - (a,h)\frac{\partial \vec{r}}{\partial h} - (a,k)\frac{\partial \vec{r}}{\partial k} - (a,p)\frac{\partial \vec{r}}{\partial p} - (a,q)\frac{\partial \vec{r}}{\partial q} - (a,\lambda)\frac{\partial \vec{r}}{\partial \lambda}$$

Inspection of Table 3.4 shows that only the Poisson Bracket (a,λ) is non-zero. Thus the above equation becomes

$$\left(\frac{\partial a}{\partial \dot{\vec{r}}}\right)^T = -\left[-2a\ s_1\right]\left[\frac{\dot{\vec{r}}}{n}\right]$$

$$\therefore \quad \frac{\partial a}{\partial \dot{\vec{r}}} = \frac{2}{n^2 a}\dot{\vec{r}}^T$$

Similar computations, although more complex, yield the results summarized in Table 3.5 (105, 110).

Table 3.5

Partial Derivatives of Equinoctial Elements with Respect to Velocity Components

$$\frac{\partial a}{\partial \vec{r}} = \frac{2}{n^2 a} \vec{r}^{\,T} \tag{3.58a}$$

$$\frac{\partial h}{\partial \vec{r}} = -\frac{1}{\mu}\left(G\hat{f}^T + r\dot{X}\hat{y}^T\right) + \frac{k}{G}(qYI - pX)\,\hat{w}^T \tag{3.58b}$$

$$\frac{\partial k}{\partial \vec{r}} = \frac{1}{\mu}\left(G\hat{g}^T + r\dot{Y}\hat{y}^T\right) - \frac{h}{G}(qYI - pX)\,\hat{w}^T \tag{3.58c}$$

$$\frac{\partial p}{\partial \vec{r}} = \frac{(1 + p^2 + q^2)\,Y\,\hat{w}^T}{2G} \tag{3.58d}$$

$$\frac{\partial q}{\partial \vec{r}} = \frac{(1 + p^2 + q^2)\,X\,\hat{w}^T}{2G} \tag{3.58e}$$

$$\frac{\partial \lambda}{\partial \vec{r}} = -\frac{2}{na^2}\vec{r}^{\,T} + \frac{\beta\sqrt{1 - h^2 - k^2}}{na^2}\left[\left(h\frac{\partial X}{\partial h} + k\frac{\partial X}{\partial k}\right)\hat{f}^T + \left(h\frac{\partial Y}{\partial h} + k\frac{\partial Y}{\partial k}\right)\hat{g}^T\right] + \frac{1}{G}(qYI - pX)\,\hat{w}^T \tag{3.58f}$$

where $\hat{y} = \dfrac{\hat{w}\times\vec{r}}{r}$

3.3.2.4 Partial Derivatives of Elements with Respect to Position

Broucke (36) also noted that a relation similar to Equation (3.55) exists to determine the partial derivatives of the equinoctial elements with respect to the components of the velocity vector. The required relation is given by

$$\frac{\partial a_i}{\partial r_j} = \sum_{k=1}^{6} (a_i, a_k) \frac{\partial \dot{r}_j}{\partial a_k} \tag{3.59}$$

for $i = 1,2,\ldots,6$

and $j = 1,2,3$

Applying Equation (3.59) and using the results of Table 3.4 and Equations (3.49a) - (3.49f), the partial derivatives of the equinoctial elements with respect to position are computed. The results are shown in Table 3.6. These results are in a slightly more compact form than those presented in Reference 105.

The matrices $\frac{\partial \vec{r}}{\partial \vec{a}}$ and $\frac{\partial \dot{\vec{r}}}{\partial \vec{a}}$ are both 3x6 matrices. The 6x6 matrix $\frac{\partial \vec{r}, \dot{\vec{r}}}{\partial \vec{a}}$ is formed by combining the above two matrices in the following manner:

Table 3.6
Partial Derivatives of Equinoctial Elements
with Respect to Position Components

$$\frac{\partial a}{\partial \vec{r}} = \frac{2}{a}\left(\frac{a}{r}\right)^3 \vec{r}^T \tag{3.60a}$$

$$\frac{\partial h}{\partial \vec{r}} = \frac{1}{\mu}\ddot{r}X\hat{z}^T - \frac{X}{r^2}\hat{y}^T - \frac{k}{G}[q\dot{Y}I - p\dot{X}]\hat{w}^T \tag{3.60b}$$

$$\frac{\partial k}{\partial \vec{r}} = -\frac{1}{\mu}\ddot{r}Y\hat{z}^T + \frac{Y}{r^2}\hat{y}^T + \frac{h}{G}[q\dot{Y}I - p\dot{X}]\hat{w}^T \tag{3.60c}$$

$$\frac{\partial p}{\partial \vec{r}} = -\frac{(1 + p^2 + q^2)\dot{Y}\hat{w}^T}{2G} \tag{2.60d}$$

$$\frac{\partial q}{\partial \vec{r}} = -\frac{(1 + p^2 + q^2)\dot{X}\hat{w}^T}{2G} \tag{3.60e}$$

$$\begin{aligned}\frac{\partial \lambda}{\partial \vec{r}} = &-\frac{\dot{\vec{r}}^T}{na^2} - \frac{\beta\sqrt{1 - h^2 - k^2}}{na^2}\left[\left(h\frac{\partial \dot{X}}{\partial h} + k\frac{\partial \dot{X}}{\partial k}\right)\hat{f}^T\right. \\ &\left. + \left(h\frac{\partial \dot{Y}}{\partial h} + k\frac{\partial \dot{Y}}{\partial k}\right)\hat{g}^T\right] \\ &- \frac{1}{G}(q\dot{Y}I - p\dot{X})\hat{w}^T\end{aligned} \tag{3.60f}$$

where $\hat{y} = \dfrac{\hat{w}x\vec{r}}{r}$

$\hat{z} = \dfrac{\hat{w}x\dot{\vec{r}}}{r}$

$$\frac{\partial \vec{r}, \dot{\vec{r}}}{\partial \vec{a}} = \begin{pmatrix} \dfrac{\partial \vec{r}}{\partial \vec{a}} \\ \hline \dfrac{\partial \dot{\vec{r}}}{\partial \vec{a}} \end{pmatrix} \tag{3.61}$$

Similarly, the 6x6 matrix $\dfrac{\partial \vec{a}}{\partial \vec{r}, \dot{\vec{r}}}$ can be formed from the 6x3 matrices $\dfrac{\partial \vec{a}}{\partial \vec{r}}$ and $\dfrac{\partial \vec{a}}{\partial \dot{\vec{r}}}$, partitioned as shown below

$$\frac{\partial \vec{a}}{\partial \vec{r}, \dot{\vec{r}}} = \left(\frac{\partial \vec{a}}{\partial \vec{r}} \;\middle|\; \frac{\partial \vec{a}}{\partial \dot{\vec{r}}} \right) \tag{3.62a}$$

When the matrix of Equation (3.62a) is evaluated at the epoch time $t = t_o$, the result is

$$\left. \frac{\partial \vec{a}}{\partial \vec{r}, \dot{\vec{r}}} \right|_{t = t_o} = \frac{\partial \vec{a}_o}{\partial \vec{r}_o, \dot{\vec{r}}_o} \tag{3.62b}$$

These matrices, when evaluated at a specific time, are exact. Thus, the first and third 6x6 matrices of the desired state partial derivatives given by Equation (3.42) have been determined. It now remains to find a suitable approximation to the 6x6 time-varying matrix of partial derivatives

of the current equinoctial elements with respect to the epoch equinoctial elements.

3.3.2.5 Partial Derivatives of Current Elements with Respect to Epoch Elements

The simplest treatment of the partial derivatives of current equinoctial elements with respect to epoch equinoctial elements is to assume a pure two body solution. In this case, the five slowly varying elements are, in fact, constant, such that

$$a = a_o$$
$$h = h_o$$
$$k = k_o$$
$$p = p_o$$
$$q = q_o$$

The rapidly varying mean longitude is given by

$$\lambda = n(t - t_o) + \lambda_o$$

where n = mean motion $= \mu^{1/2} a^{-3/2}$

Under this assumption, the elements of the desired partial derivative matrix become

$$\frac{\partial a_i}{\partial a_{oj}} = \begin{cases} 1 & , \quad i = j;\ i = 1,2,\ldots,6 \\ -\frac{3}{2}\frac{n}{a}(t - t_o) & \quad i = 6,\ j = 1 \\ 0 & \quad \text{all others} \end{cases} \tag{3.63}$$

While the two body solution is computationally easy, it provides the least accurate numerical results. A novel approximation to this matrix of partial derivatives is formulated below. The method includes the secular rates in the argument of periapsis and longitude of the ascending node due to the dominant J_2 zonal harmonic in the gravitational potential.

Kozai (94) and others (41, 60, 113) give the well-known formulae for the first order, secular perturbations due to J_2 in the argument of periapsis (ω) and the longitude of the ascending node (Ω) as

$$\omega = \omega_o + \left[\frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n \left(2 - \frac{5}{2}\sin^2 i\right)\right] (t - t_o) \tag{3.64a}$$

$$\Omega = \Omega_o - \left[\frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n \cos i\right] (t - t_o) \tag{3.64b}$$

with $$M = M_o + n(t - t_o) \tag{3.64c}$$

$$n = n_o + \frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n_o \left(1 - \frac{3}{2} \sin^2 i\right) \sqrt{1 - e^2} \tag{3.64d}$$

where p = parameter = $a(1 - e^2)$

r_e = mean equatorial radius of Earth

e = satellite orbital eccentricity

i = satellite orbital inclination

t_o = epoch time

In the above expressions all quantities are assumed to be mean values (i.e., only contain the secular component of the motion) except n_o, which is the unperturbed mean motion and is related to the unperturbed semimajor axis a by $n_o^2 a_o^3 = \mu$. The J_2 secular rates are therefore given by

$$\dot{a} = \dot{e} = \dot{i} = 0$$

$$\dot{\omega} = \frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n \left(2 - \frac{5}{2} \sin^2 i\right) \tag{3.65a}$$

$$\dot{\Omega} = -\frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n \cos i \tag{3.65b}$$

The term $(r_e/p)^2\, n$ can be rewritten as

$$\left(\frac{r_e}{p}\right)^2 n = \frac{r_e^2}{a^2(1 - e^2)^2}\, \mu^{1/2}\, a^{-3/2}$$

$$= r_e^2\, \mu^{1/2}\, a^{-7/2} (1 - e^2)^{-2}$$

Define $$x = \frac{1}{1 - e^2} \tag{3.66}$$

Then $$\left(\frac{r_e}{p}\right)^2 n = r_e^2\, \mu^{1/2}\, a^{-7/2}\, x^2 \tag{3.67}$$

It is possible to determine the effects of these secular rates on the equinoctial elements by using the basic definitions of the equinoctial elements in terms of the classical Keplerian elements given in Appendix A. For example

$$h = e \sin(\omega + I\Omega)$$

where I = retrograde factor

By substituting Equations (3.64a) and (3.64b) into the above definition, the equinoctial element h is given by

$$h = e \sin[(\omega_o + I\Omega_o) + (\dot{\omega} + I\dot{\Omega})(t - t_o)]$$

By use of the trigonometric identity

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \gamma \sin \beta$$

and defining

$$\delta_1 = \dot{\omega} + I\dot{\Omega} \tag{3.68}$$

the above expression can be rewritten as

$$h = e\{\sin(\omega_o + I\Omega_o) \cos \delta_1(t - t_o)$$

$$+ \cos(\omega_o + I\Omega_o) \sin \delta_1(t - t_o)\}$$

But, since there is no secular perturbation in the eccentricity, $e = e_o$. The above equation is then

$$h = e_o \sin(\omega_o + I\Omega_o) \cos \delta_1(t - t_o)$$

$$+ e_o \cos(\omega_o + I\Omega_o) \sin \delta_1(t - t_o)$$

From the definitions of the equinoctial elements, it is seen that

$$e_o \sin(\omega_o + I\Omega_o) = h_o$$

$$e_o \cos(\omega_o + I\Omega_o) = k_o$$

Hence, the secular perturbation in the equinoctial element h is given by

$$h = h_o \cos \delta_1(t - t_o) + k_o \sin \delta_1(t - t_o) \tag{3.69}$$

In a similar manner, the secular perturbation in the equinoctial element k is given by

$$k = k_o \cos \delta_1(t - t_o) - h_o \sin \delta_1(t - t_o) \tag{3.70}$$

The definition of the equinoctial element p in terms of the Keplerian elements is given by

$$p = \tan^I \left(\frac{i}{2}\right) \sin \Omega$$

By remembering that $i = i_o$, since there is no secular rate in the inclination due to J_2, and by substituting the results of Equation (3.64b), the definition above can be rewritten as

$$p = \tan^{I}\left(\frac{i_o}{2}\right) \sin\left[\Omega_o + \delta_2(t - t_o)\right]$$

when $$\delta_2 = -\frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n \cos i = \dot{\Omega} \qquad (3.71)$$

Use of the previously mentioned identity yields

$$p = p_o \cos \delta_2(t - t_o) + q_o \sin \delta_2(t - t_o) \qquad (3.72)$$

and the analogous development for q yields

$$q = q_o \cos \delta_2(t - t_o) - p_o \sin \delta_2(t - t_o) \qquad (3.73)$$

Finally, the mean longitude is defined by

$$\lambda' = M + \omega + I\Omega$$

By substituting Equation (3.64a), (3.64b) and (3.64c) into the above

$$\lambda = \lambda_o + (n_o + \delta_1 + \delta_3)(t - t_o) \qquad (3.74)$$

where $$\delta_3 = \frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n_o \left(1 - \frac{3}{2}\sin^2 i\right) \sqrt{i - e^2} \tag{3.75}$$

The term δ_1 defined by Equation (3.68) can be expressed as a function of the epoch elements a_o, h_o, k_o, p_o and q_o as follows. From the definition and using Equations (3.65a), (3.65b), and (3.67), δ_1 is given by

$$\delta_1 = \dot{\omega} + I\dot{\Omega}$$

$$= \frac{3}{2} J_2 \left(\frac{r_e}{p}\right)^2 n \left(2 - \frac{5}{2}\sin^2 i - I \cos i\right)$$

$$\delta_1 = \frac{3}{2} J_2 r_e^2 \mu^{1/2} a^{-7/2} x^2 \left(2 - \frac{5}{2}\sin^2 i - I \cos i\right) \tag{3.76}$$

Equation (3.66) yields

$$x = \frac{1}{1 - e^2} = \frac{1}{1 - h^2 - k^2}$$

Substitution of Equations (3.69) and (3.70) into the above relation provides the identity

$$x = \frac{1}{1 - h_o^2 - k_o^2}$$

since $$h^2 + k^2 \equiv h_o^2 + k_o^2 \tag{3.77}$$

This implies that the J_2 secular perturbation does not change the magnitude of the eccentricity vector but causes it to rotate with respect to inertial space. Similarly, if a variable c is defined as

$$c = p^2 + q^2 = \left[\tan^I\left(\frac{i}{2}\right)\right]^2 \tag{3.78a}$$

substitution of the results of Equations (3.72) and (3.73) yields a similar identity

$$c = p_o^2 + q_o^2 \tag{3.78b}$$

This implies that the J_2 secular perturbation also results in pure rotation of the vector of magnitude $\tan^I\left(\frac{i}{2}\right)$ pointing in the direction of the lines of nodes.

Equation (3.78a) relates the auxiliary variable c to trigonometric functions of the inclination. By use of the trigonometric identity

$$\tan \frac{\alpha}{2} = \pm \sqrt{\frac{1 - \cos \alpha}{1 + \cos \alpha}}$$

it is easily shown that

$$\cos i = \frac{I(1 - c)}{1 + c} \tag{3.79a}$$

and

$$\sin^2 i = \frac{4c}{(1 + c)^2} \tag{3.79b}$$

Hence, the trigonometric terms in Equation (3.76) can be expressed as rational polynomial functions of the auxiliary variable c as

$$\left(2 - \frac{5}{2} \sin^2 i - I \cos i\right) = \frac{3c^2 - 6c + 1}{(c + 1)^2} \tag{3.80}$$

Using this result in Equation (3.76), the δ_1 term is given by

$$\delta_1(a_o, h_o, k_o, p_o, q_o) = K a_o^{-7/2} x^2 \left[\frac{3c^2 - 6c + 1}{(c + 1)^2}\right] \tag{3.81}$$

where $$K \triangleq \frac{3}{2} J_2 r_e^2 \mu^{1/2} \tag{3.82}$$

and x is a function only of h_o and k_o, while c is a function only of p_o and q_o.

A similar analysis may be applied to Equations (3.71) and (3.75). The results are

$$\delta_2(a_o,h_o,k_o,p_o,q_o) = -Ka_o^{-7/2} x^2 \left[\frac{I(1 - c)}{c + 1}\right] \tag{3.83}$$

$$\delta_3(a_o,h_o,k_o,p_o,q_o) = Ka_o^{-7/2} x^{3/2} \left[\frac{c^2 - 4c + 1}{(c + 1)^2}\right] \tag{3.84}$$

The J_2 secular equations of motion are summarized in Table 3.7. It should be noted that, if a small eccentricity assumption is made such that e^2 terms may be neglected, the variable x reduces to one. In this case, the equations of Table 3.7 become exactly those described by Equations (3.4a), (3.4b), (3.16a), (3.16b), (3.18a) and (3.18b) of Zeis' thesis (156). The results of Table 3.7 provide the current mean elements as functions of the epoch mean elements. It now remains to partially differentiate these expressions with respect to the epoch mean elements.

Table 3.7

Summary of Secular Equations of Motion Due to J_2 Zonal Harmonic

$$a = a_o$$

$$h = h_o \cos \delta_1 (t - t_o) + k_o \sin \delta_1 (t - t_o)$$

$$k = k_o \cos \delta_1 (t - t_o) - h_o \sin \delta_1 (t - t_o)$$

$$p = p_o \cos \delta_2 (t - t_o) + q_o \sin \delta_2 (t - t_o)$$

$$q = q_o \cos \delta_2 (t - t_o) + p_o \sin \delta_2 (t - t_o)$$

$$\lambda = \lambda_o + (n_o + \delta_1 + \delta_3)(t - t_o)$$

where

$$\delta_1 = K a_o^{-7/2} x^2 \left[\frac{3c^2 - 6c + 1}{(1 + c)^2} \right]$$

$$\delta_2 = K a_o^{-7/2} x^2 \left[\frac{I(c - 1)}{c + 1} \right]$$

$$\delta_3 = K a_o^{-7/2} x^{3/2} \left[\frac{c^2 - 4c + 1}{(1 + c)^2} \right]$$

with

$$K = \frac{3}{2} J_2 r_e^2 \mu^{1/2}$$

$$c = p^2 + q^2 = p_o^2 + q_o^2$$

$$x^2 = \frac{1}{1 - h^2 - k^2} = \frac{1}{1 - h_o^2 - k_o^2}$$

Note: All elements in the above expressions are assumed to be secular (or average) elements and the overbar notation is therefore neglected.

Table 3.8

Useful Partial Derivatives in Computation of J_2-Secular Perturbed Partial Derivatives

$$\frac{\partial \delta_1}{\partial a_o} = -\frac{7}{2}\frac{\delta_1}{a_o} \qquad \frac{\partial x}{\partial h_o} = 2h_o x^2$$

$$\frac{\partial \delta_1}{\partial x} = 2\frac{\delta_1}{x} \qquad \frac{\partial x}{\partial k_o} = 2k_o x^2$$

$$\frac{\partial \delta_1}{\partial c} = 4Ka_o^{-7/2} x^2 \left[\frac{3c - 2}{(c + 1)^3}\right] \qquad \frac{\partial c}{\partial p_o} = 2p_o$$

$$\frac{\partial \delta_2}{\partial a_o} = \frac{7}{2}\frac{\delta_2}{a_o} \qquad \frac{\partial c}{\partial q_o} = 2q_o$$

$$\frac{\partial \delta_2}{\partial x} = 2\frac{\delta_2}{x} \qquad \frac{\partial h}{\partial \delta_1} = k(t - t_o)$$

$$\frac{\partial \delta_2}{\partial c} = 2Ka_o^{-7/2} x^2 \left[\frac{I}{(c + 1)^2}\right] \qquad \frac{\partial k}{\partial \delta_1} = -h(t - t_o)$$

$$\frac{\partial \delta_3}{\partial a_o} = -\frac{7}{2}\frac{\delta_3}{a_o} \qquad \frac{\partial p}{\partial \delta_2} = q(t - t_o)$$

$$\frac{\partial \delta_3}{\partial x} = \frac{3}{2}\frac{\delta_3}{x} \qquad \frac{\partial q}{\partial \delta_2} = -p(t - t_o)$$

$$\frac{\partial \delta_3}{\partial c} = 6Ka_o^{-7/2} x^{3/2} \left[\frac{c - 1}{(c + 1)^3}\right] \qquad \frac{\partial \lambda}{\partial \delta_1} = \frac{\partial \lambda}{\partial \delta_3} = (t - t_o)$$

Chain rule partial differentiation has been applied to the expressions of Table 3.7. In performing these differentiations, certain useful partial derivatives were computed. These partial derivatives are presented in Table 3.8. An example of this partial differentation is given below. Equation (3.69), for example, gives the mean element h at time t in terms of the epoch mean elements a_o, h_o, k_o, p_o and q_o. To determine the partial derivative of h with respect to a_o, the chain rule yields

$$\frac{\partial h}{\partial a_o} = \frac{\partial h}{\partial \delta_1} \frac{\partial \delta_1}{\partial a_o}$$

Using the results in Table 3.8, this expression can be rewritten as

$$\frac{\partial h}{\partial a_o} = \left[k(t - t_o)\right]\left[-\frac{7}{2}\frac{\delta_1}{a_o}\right]$$

$$= -\frac{7}{2}\frac{k}{a_o}\delta_1(t - t_o)$$

where k is given by Equation (3.70) and δ_1 by Equation (3.81). If the partial derivative of h with respect to h_o is desired, there is an explicit partial derivative, from

the explicit appearance of h_o in Equation (3.69), and an implicit derivative through the dependence of δ_1 on h_o. Thus

$$\frac{\partial h}{\partial h_o} = \left(\frac{\partial h}{\partial h_o}\right)_{\text{explicit}} + \frac{\partial h}{\partial \delta_1} \frac{\partial \delta_1}{\partial x} \frac{\partial x}{\partial h_o}$$

$$= \cos \delta_1 (t - t_o) + \left[k(t - t_o)\right]\left[2 \frac{\delta_1}{x}\right]\left[2h_o \, x^2\right]$$

$$\therefore \quad \frac{\partial h}{\partial h_o} = \cos \delta_1 (t - t_o) + 4k \, h_o \, x \, \delta_1 (t - t_o)$$

In a similar manner, the remaining partial derivatives can be found. These derivatives are summarized in Tables 3.9 through 3.14.

The results presented in these tables have the following physical significance. The pure two body partial derivatives are derived from the motion of a satellite in an orbit fixed in shape (a,e), fixed in orientation with respect to inertial space (ω,Ω,i) and moving with a constant frequency (n_o). The partial derivatives just obtained are based on a slightly more complete model of the satellite motion. In this model the orbit shape (a,e) remains

Table 3.9

Partial Derivatives of "a" with Respect to

Epoch Mean Elements Including J_2 Secular Perturbation

$$\frac{\partial a}{\partial a_o} = 1$$

$$\frac{\partial a}{\partial h_o} = 0$$

$$\frac{\partial a}{\partial k_o} = 0$$

$$\frac{\partial a}{\partial p_o} = 0$$

$$\frac{\partial a}{\partial q_o} = 0$$

$$\frac{\partial a}{\partial \lambda_o} = 0$$

Table 3.10

Partial Derivatives of "h" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation

$$\frac{\partial h}{\partial a_o} = -\frac{7}{2}\frac{k}{a_o}\,\delta_1(t - t_o)$$

$$\frac{\partial h}{\partial h_o} = \cos\,\delta_1(t - t_o) + 4k\,h_o \times \delta_1(t - t_o)$$

$$\frac{\partial h}{\partial k_o} = \sin\,\delta_1(t - t_o) + 4k\,k_o \times \delta_1(t - t_o)$$

$$\frac{\partial h}{\partial p_o} = 8k\,p_o\,K\,a_o^{-7/2}\,x^2\left[\frac{3c - 2}{(c + 1)^3}\right](t - t_o)$$

$$\frac{\partial h}{\partial q_o} = 8k\,q_o\,K\,a_o^{-7/2}\,x^2\left[\frac{3c - 2}{(c + 1)^3}\right](t - t_o)$$

$$\frac{\partial h}{\partial \lambda_o} = 0$$

Table 3.11

Partial Derivatives of "k" with Respect to

Epoch Mean Elements Including J_2 Secular Perturbation

$$\frac{\partial k}{\partial a_o} = \frac{7}{2} \frac{h}{a_o} \delta_1 (t - t_o)$$

$$\frac{\partial k}{\partial h_o} = - \sin \delta_1 (t - t_o) - 4h\, h_o \times \delta_1 (t - t_o)$$

$$\frac{\partial k}{\partial k_o} = \cos \delta_1 (t - t_o) - 4h\, k_o \times \delta_1 (t - t_o)$$

$$\frac{\partial k}{\partial p_o} = -8h\, p_o\, K\, a_o^{-7/2}\, x^2 \left[\frac{3c - 2}{(c + 1)^3}\right] (t - t_o)$$

$$\frac{\partial k}{\partial q_o} = -8h\, q_o\, K\, a_o^{-7/2}\, x^2 \left[\frac{3c - 2}{(c + 1)^3}\right] (t - t_o)$$

$$\frac{\partial k}{\partial \lambda_o} = 0$$

Table 3.12

Partial Derivatives of "p" with Respect to Epoch Mean Elements Including J_2 Secular Perturbation

$$\frac{\partial p}{\partial a_o} = \frac{7}{2} \frac{q}{a_o} \delta_2 (t - t_o)$$

$$\frac{\partial p}{\partial h_o} = 4q\, h_o\, x\, \delta_2 (t - t_o)$$

$$\frac{\partial p}{\partial k_o} = 4q\, k_o\, x\, \delta_2 (t - t_o)$$

$$\frac{\partial p}{\partial q_o} = \cos \delta_2 (t - t_o) + 4q\, p_o\, K\, a_o^{-7/2}\, x^2 \left[\frac{I}{(c+1)^2}\right](t-t_o)$$

$$\frac{\partial p}{\partial q_o} = \sin \delta_2 (t - t_o) + 4q\, q_o\, K\, a_o^{-7/2}\, x^2 \left[\frac{I}{(c+1)^2}\right](t-t_o)$$

$$\frac{\partial p}{\partial \lambda_o} = 0$$

Table 3.13

Partial Derivatives of "q" with Respect to
Epoch Mean Elements Including J_2 Secular Perturbation

$$\frac{\partial q}{\partial a_o} = -\frac{7}{2}\frac{p}{a_o}\,\delta_2(t - t_o)$$

$$\frac{\partial q}{\partial h_o} = -4p\,h_o \times \delta_2(t - t_o)$$

$$\frac{\partial q}{\partial k_o} = -4p\,k_o \times \delta_2(t - t_o)$$

$$\frac{\partial q}{\partial p_o} = -\sin\,\delta_2(t - t_o) - 4p\,p_o\,K\,a_o^{-7/2}x^2\left[\frac{I}{(c+1)^2}\right](t-t_o)$$

$$\frac{\partial q}{\partial q_o} = \cos[\delta_2(t - t_o)] - 4p\,q_o\,K\,a_o^{-7/2}x^2\left[\frac{I}{(c+1)^2}\right](t-t_o)$$

$$\frac{\partial q}{\partial \lambda_o} = 0$$

Table 3.14

Partial Derivatives of "λ" with Respect to

Epoch Mean Elements Including J_2 Secular Perturbation

$$\frac{\partial \lambda}{\partial a_o} = -\left[\frac{3}{2}\frac{n_o}{a_o} + \frac{7}{2a_o}(\delta_1 + \delta_3)\right](t - t_o)$$

$$\frac{\partial \lambda}{\partial h_o} = 2h_o \times \left[2\delta_1 + \frac{3}{2}\delta_3\right](t - t_o)$$

$$\frac{\partial \lambda}{\partial k_o} = 2k_o \times \left[2\delta_1 + \frac{3}{2}\delta_3\right](t - t_o)$$

$$\frac{\partial \lambda}{\partial p_o} = 4p_o\, Ka^{-7/2}\left\{2x^2\left[\frac{3c - 2}{(c + 1)^3}\right] + 3x^{3/2}\left[\frac{c - 1}{(c + 1)^3}\right]\right\}(t-t_o)$$

$$\frac{\partial \lambda}{\partial q_o} = 4q_o\, Ka^{-7/2}\left\{2x^2\left[\frac{3c - 2}{(c + 1)^3}\right] + 3x^{3/2}\left[\frac{c - 1}{(c + 1)^3}\right]\right\}(t-t_o)$$

$$\frac{\partial \lambda}{\partial \lambda_o} = 1$$

constant, as does the orbit plane orientation with respect to the inertial z axis (i). However, this model accomodates the secular rotations of the orbit plane about its angular momentum vector (progression or regression of the line of apsides) and about the inertial z axis (precession of the line of nodes). Additionally, the model accounts for the slightly different orbital frequency occasioned by the dominant zonal harmonic. Over short observation spans, these effects may be negligible and the two body partial derivatives should be adequate for use in computing the gradient of the weighted least squares cost functional. For example, Battin (28) derives the average rates of rotation for the argument of periapsis and longitude of the ascending node for a near circular orbit of radius r_o as

$$\overline{\Delta\omega} = -5.0 \left(\frac{r_e}{r_o}\right)^{3.5} (1 - 5 \cos^2 i) \text{ deg/day} \tag{3.85a}$$

$$\overline{\Delta\Omega} = -10.05 \left(\frac{r_e}{r_o}\right)^{3.5} \cos i \text{ deg/day} \tag{3.85b}$$

For an orbit with the characteristics of the low altitude circular test case of Chapter 2, these rates are approximately given by

$$\Delta\bar{\omega} = -1.29 \text{ deg/day}$$

$$\Delta\bar{\Omega} = -3.26 \text{ deg/day}$$

Thus, over a three hour integration little difference would be seen between the two body partial derivatives and the secular J_2 partial derivatives, since the pericenter position has only shifted by approximately -0.15 degrees and the line of nodes by about -0.41 degrees. Over a three day integration, however, the secular J_2 partial derivatives would be clearly superior, since the pericenter has now regressed by 4 degrees and the node by 10 degrees, neither of which would be reflected in the two body partial derivatives. In the orbit estimation test cases presented in Section 3.6, the observation spans were usually three hours long and little improvement in the estimation solutions could be attributed to the more complete J_2 partial derivatives.

3.3.3 Direct Integration Method

In the evaluation of the weighted least squares cost functional and its gradient vector it is required to determine the vector of observation residuals given by Equation (3.30). In that expression the $\vec{y}$ vector contains the

"batch" of actual observational data to be processed. The vector $\vec{f}(\vec{x})$ represents the corresponding set of predicted observational data computed from the deterministic model of the satellite motion and the necessary observation models as functions of the solve-for vector $\vec{x}$.

Under the assumptions discussed in Section 3.1, the observed quantities are taken to be the components of the inertial Cartesian position and velocity. Therefore, the observation models are the equations of motion for the satellite. Because this implementation uses a Special Perturbation deterministic model for the satellite motion, no explicit analytical formulae directly relate the solve-for vector $\vec{x}$ (comprised of epoch position and velocity, epoch Keplerian elements or epoch equinoctial elements) to the observed quantities. Hence, each evaluation of the cost functional for a given solve-for vector requires a full numerical integration of the satellite equations of motion over the entire observation span. At the ith observation time, the position and velocity can be stored. The appropriate scalar component (predicted observation) can then be used to determine the ith observation residual. Additionally, the position and velocity at that time, along with the corresponding orbital elements, can be used to determine the necessary observation partial derivatives

array. The row of that array that corresponds to the scalar component being observed can then be stored as the ith row of the mxp array $\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}$.

Review of the algorithm flow described in Section 3.1 shows that a single iteration through the Parallel Variable Metric algorithm would require numerous integrations over the total observation span. The initialization of each step requires a cost and gradient evaluation based on the current best estimate of the solve-for vector. Step 1 of the method then performs p parallel cost and gradient evaluations at the p values of the solve-for vector given by Equation (3.5). The variable p is equivalent to the number of solve-for parameters and for the implementation described here is limited to six. Hence, seven integrations are required in Step 1, but it is assumed that six of these can be performed in parallel, making the total effectively two integrations in Step 1. Figure 3.5 presents a pictorial generalization of the necessary integrations required in Step 1 of the PVM method. The left hand "plane" represents the plane of initial condition vectors, including the current best estimate $\vec{x}^i$ and the six other values generated by Equation (3.5). The trajectories are integrated through the observation span and the necessary observation residuals

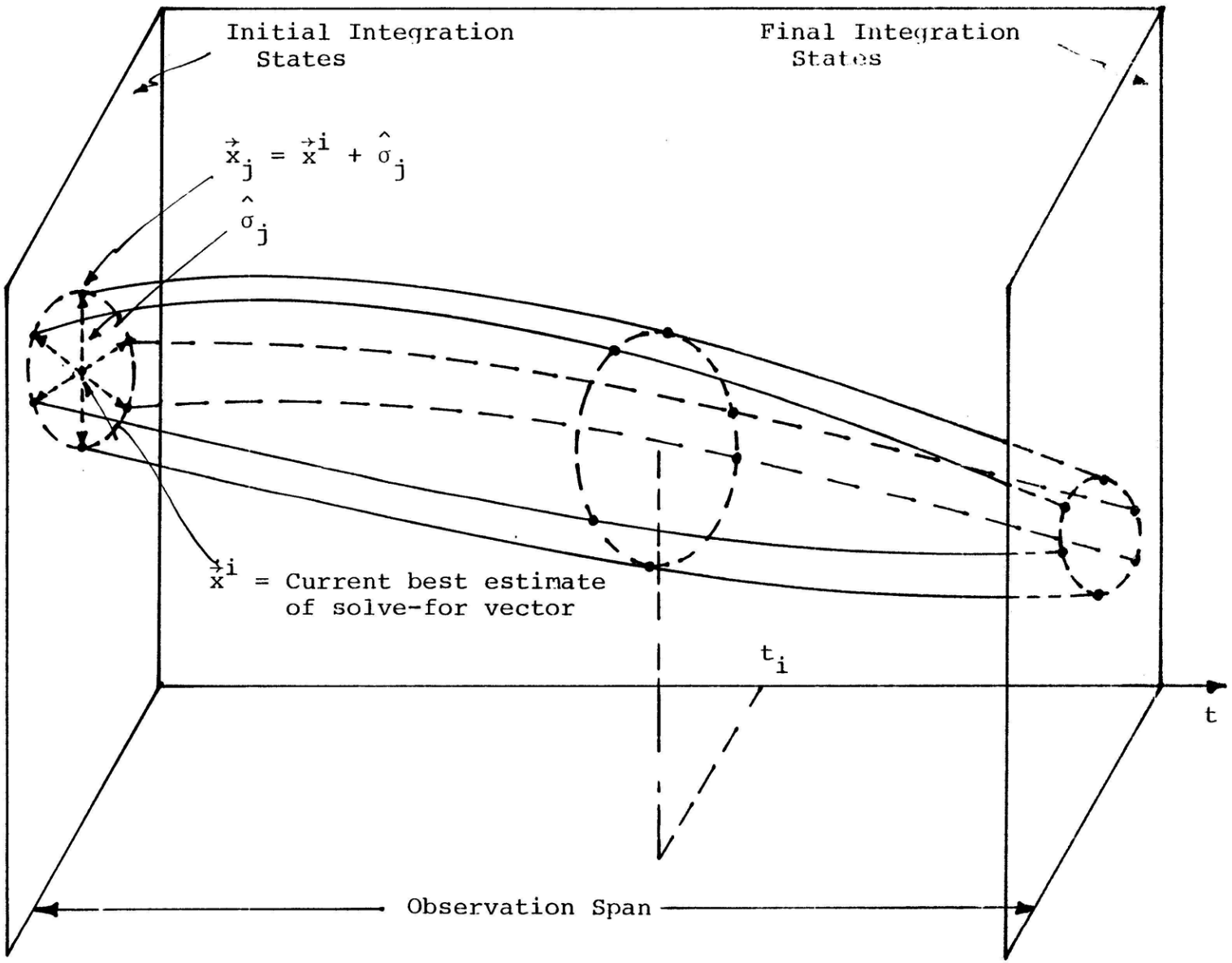


Figure 3.5. Pictorial Representation of the Direct Integration Method for Cost and Gradient Evaluations

and observation partial derivatives computed at each observation time t_i.

The Coggins univariate search technique requires several functional evaluations. The Davis, Swann and Campey portion of the algorithm will require at least three cost functional evaluations to bracket the minimum of the cost functional with respect to the scalar step size α. This assumes a good initial guess for the minimizing value of α. If the initial guess is poor, significantly more cost functional evaluations may be required to bracket the minimum. Also, the Powell portion of the algorithm requires at least two cost functional evaluations to finally locate the minimum of the cost functional with respect to α. If the cost functional is very shallow, such that the minimum is not clearly defined, then more cost functional evaluations may be required by the Powell algorithm. At a minimum, the univariate search technique requires five cost functional evaluations, implying five numerical integrations of the satellite equations of motion. These integrations are not amenable to parallel processing, because the initial conditions in any one integration are dependent on the results of the previous integration. Therefore, any single iteration through the PVM method requires effectively at least seven numerical integrations

of the satellite dynamics over the total observation span in the direct integration method.

3.3.4 Linearization Method

Sequential simulations have indicated that the direct integration method is an extremely costly method for evaluation of the cost functional and its gradient vector. Because of this, an alternate method of evaluating the necessary trajectories was sought. Discussion in Section 3.1.1.1 demonstrated that the application of Equation (3.5) to create one of the p new values of the solve-for vector resulted in a variation in only one of the components of the current "best estimate" of the solve-for vector. This is because the $\hat{\sigma}$'s are basis vectors in parameter space and hence the sum

$$\vec{x}_j = \vec{x}^i + \hat{\sigma}_j$$

can only influence the jth component of the vector $\vec{x}^i$.

If the solve-for parameters are the epoch position and velocity components, then the six values of the solve-for vectors arising from Equation (3.5) are six vectors of epoch position and velocity components, each one with

a single component perturbed from its *a priori* value by the magnitude of its corresponding basis vector. That is, for the first iteration through the method

$$\vec{x}_1 = x^1 + \hat{\sigma}_1 = \begin{pmatrix} r_{xo} + |\hat{\sigma}_1| \\ r_{yo} \\ r_{zo} \\ v_{xo} \\ v_{yo} \\ v_{xo} \end{pmatrix}$$

$$\vec{x}_2 = \quad .$$
$$\quad\quad\quad .$$
$$\vec{x}_3 = \quad .$$

$$\vec{x}_4 = x^1 + \hat{\sigma}_4 = \begin{pmatrix} r_{xo} \\ r_{yo} \\ r_{zo} \\ v_{xo} + |\hat{\sigma}_4| \\ v_{yo} \\ v_{zo} \end{pmatrix}$$

$$\vec{x}_5 = \quad .$$
$$\vec{x}_6 = \quad .$$
$$\quad\quad\quad .$$

It is also interesting to note that the magnitudes of the basis vectors are adjusted by the user and are generally scaled to be about 10^{-4} of the order of magnitude of the particular solve-for parameter. For example, if the epoch position and velocity components are the solve-for parameters, the position components being of the order of 10^3 to 10^4 kilometers and the velocity components being of the order 10^0 to 10^1 kilometers/second, then the basis vectors $\hat{\sigma}_1$, $\hat{\sigma}_2$ and $\hat{\sigma}_3$ should be of the order of 10^{-1} to 10^0 and the basis vectors $\hat{\sigma}_4$, $\hat{\sigma}_5$ and $\hat{\sigma}_6$ should be of the order 10^{-4} to 10^{-3}.

It is desired to determine the effects these perturbations in the epoch state have on the satellite position and velocity at the observation times and ultimately on the values of the cost functional and gradient vectors. In the direct integration method, of course, the perturbed epoch state components are each numerically integrated and the resulting costs and gradients directly computed. From the previous discussion, however, the ith iteration in the PVM method begins with an integration of the trajectory based on the current "best estimate" of the solve-for vector $\vec{x}^i$. This integration can be thought of as providing a reference trajectory about which the other six trajectories can be linearized in the following manner.

The trajectories are considered to be functions of the solve-for vectors $\vec{x}_j$, (j = 1,2,...,6) and it is assumed that each trajectory can be expanded in a first order Taylor Series about the current "best estimate" of the solve-for vector as

$$\begin{pmatrix} \vec{r}(\vec{x}_j) \\ \dot{\vec{r}}(\vec{x}_j) \end{pmatrix} = \begin{pmatrix} \vec{r}(\vec{x}^i) \\ \dot{\vec{r}}(\vec{x}^i) \end{pmatrix} + \frac{\partial}{\partial \vec{x}} \begin{pmatrix} \vec{r}(\vec{x}) \\ \dot{\vec{r}}(\vec{x}) \end{pmatrix} \Bigg|_{\vec{x}=\vec{x}^i} \delta\vec{x}_j \tag{3.86}$$

for j = 1,2,...,6

The first term on the right hand side of the above equation represents results from the numerical integration of the trajectory equations based on the "best estimate" of the solve-for vector at the ith iteration $\vec{x}^i$. The $\delta\vec{x}_j$ term can be expressed as

$$\delta\vec{x}_j = \vec{x}_j - \vec{x}^i = \hat{\sigma}_j \tag{3.87}$$

for j = 1,2,...,6

Finally the matrix of partial derivatives on the right hand side is the matrix of state partial derivatives derived in

Section 3.3.2. Thus, at the ℓth observation time the deltas in the current position and velocity are found from the deltas in the epoch position and velocity by the linear vector equation

$$\begin{pmatrix} \delta\vec{r}(\vec{x}_j, t_\ell) \\ \\ \delta\dot{\vec{r}}(\vec{x}_j, t_\ell) \end{pmatrix} = \left(\begin{array}{c} \dfrac{\partial \vec{r}(\vec{x}, t_\ell)}{\partial \vec{r}_o, \dot{\vec{r}}_o} \\ \hline \dfrac{\partial \dot{\vec{r}}(\vec{x}, t_\ell)}{\partial \vec{r}_o, \dot{\vec{r}}_o} \end{array} \right)_{\vec{x}=\vec{x}^i} \hat{\sigma}_j \qquad (3.88)$$

for $j = 1,2,\ldots,6$

and $\ell = 1,2,\ldots,m$

where

$$\delta\vec{r}(\vec{x}_j, t_\ell) = \vec{r}(\vec{x}_j, t_\ell) - \vec{r}(\vec{x}^i, t_\ell) \qquad (3.89a)$$

$$\delta\dot{\vec{r}}(\vec{x}_j, t_\ell) = \dot{\vec{r}}(\vec{x}_j, t_\ell) - \dot{\vec{r}}(\vec{x}^i, t_\ell) \qquad (3.89b)$$

These deltas in position and velocity at the observation times can then be added to the predetermined reference trajectory position and velocity components to determine the perturbed position and velocity at the observation times. This is done by inversion of Equations (3.89a) and (3.89b), such that

$$\vec{r}(\vec{x}_j, t_\ell) = \vec{r}(\vec{x}^i, t_\ell) + \delta\vec{r}(\vec{x}_j, t_\ell) \qquad (3.90a)$$

$$\dot{\vec{r}}(\vec{x}_j, t_\ell) = \dot{\vec{r}}(\vec{x}^i, t_\ell) + \delta\dot{\vec{r}}(\vec{x}_j, t_\ell) \qquad (3.90b)$$

for $j = 1,2,\ldots,6$

and $\ell = 1,2,\ldots,m$

The resulting values of position and velocity can then be used in the observation models to determine the required values of the cost functionals and their associated gradient vectors.

Figure 3.6 illustrates the linearization method for computing the cost and gradient vectors. It is seen that only one trajectory is numerically integrated over the observation span. That trajectory is based on the "best estimate" of the solve-for vector at the ith iteration $\vec{x}^i$ and serves as a reference trajectory for the linearization process. It is important to note that the linearization assumption always remains valid in this approach because the linearization is done about the current "best estimate" of the solution and not about the true solution. Since the user controls the magnitude of the deltas in the epoch integration state by controlling magnitudes of the basis

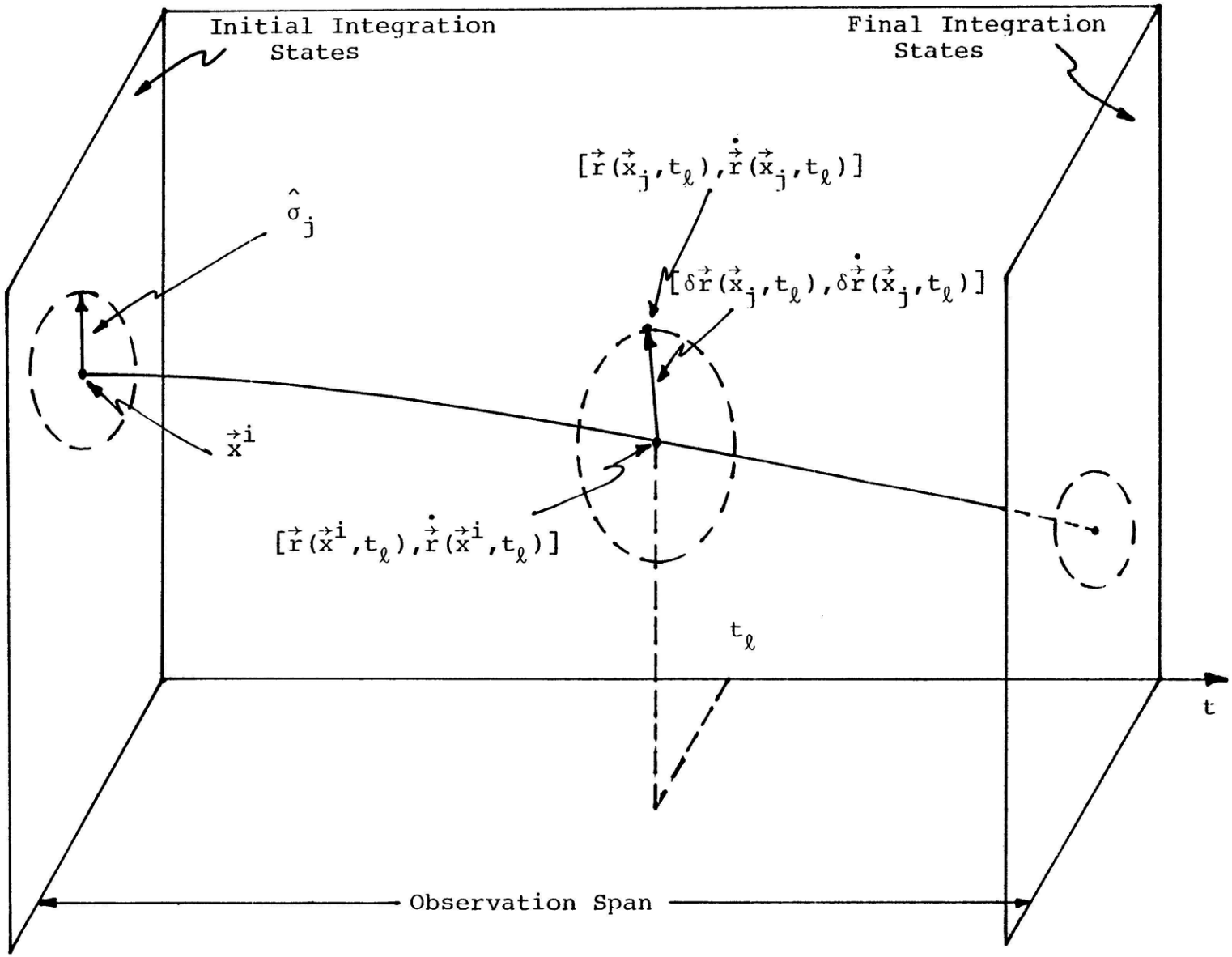


Figure 3.6. Pictorial Representation of the Linearized Method for Cost and Gradient Evaluations

vectors $\hat{c}$, these deltas should never be so large as to violate the linearity assumption. As shown in Figure 3.6, the perturbed trajectories are computed from Equations (3.81), (3.90a) and (3.90b) only at the observation times. The numerical integrations of the perturbed trajectories over the entire observation span have been replaced by a series of matrix multiplies and vector additions, making the computations significantly faster. Comparison of orbit estimation results using both the direct integration method and the linearization method have shown both techniques provide accurate estimation results. However, the linearization method has proved to be much more rapid and efficient in sequential simulations. These points will be further discussed in Chapters 3.6 and 4 of this document.

3.4 Estimation Error Covariance Computation

An additional piece of information which operational orbit analysts find useful is the covariance matrix of the estimation errors. This matrix is an indication of the degree of certainty with which the state estimate is known. In the current implementation, this computation is only performed once in any orbit estimation run after the converged results are obtained. The assumption concerning the statistical nature of the problem has already been presented in Section 3.2.

Because the covariance matrix is only evaluated once after the converged solve-for vector $\vec{x}^*$ has been found, it is assumed that the converged solution and the true solution are close together. Hence, the classical linearized solution of Equation (3.28) is assumed to be valid. Under this assumption, the derivation parallels that presented in Battin (28, 29). Equation (3.28) gives the linearized "best estimate" of the deviations in the solve-for parameters $\Delta\hat{\vec{x}}$ due to the deviations in the observed quantities $\Delta\tilde{\vec{y}}$ as

$$\Delta\hat{\vec{x}} = (F^T WF)^{-1} F^T W \Delta\tilde{\vec{y}}$$

where $$\Delta\vec{x} = \vec{x} - \vec{x}_o$$

and $$\Delta\tilde{\vec{y}} = \tilde{\vec{y}} - \vec{f}(\vec{x}_o)$$

In the above expression, $\vec{x}_o$ is the *a priori* solve-for vector, $\vec{f}$ represents the observation models, which are assumed to exactly model the system dynamics, and the $\sim$ notation implies that the associated quantity is corrupted by Gaussian additive noise. The true position deviations as a function of the true uncorrupted observations are

$$\Delta\vec{x} = (F^T WF)^{-1} F^T W \Delta\vec{y}$$

Subtracting these two equations yields

$$\vec{\varepsilon} = \Delta\hat{\vec{x}} - \Delta\vec{x} = (F^T WF)^{-1} F^T W(\Delta\tilde{\vec{y}} - \Delta y) \tag{3.91a}$$

where $\vec{\varepsilon} \triangleq$ vector of estimation errors

But, combining Equations (3.22) with Equation (3.27b) shows

$$\Delta\tilde{\vec{y}} - \Delta\vec{y} = [(\vec{f}(\vec{x}) + \vec{n}) - \vec{f}(\vec{x}_o)] - [\vec{f}(\vec{x}) - \vec{f}(\vec{x}_o)] \tag{3.91b}$$

such that $\Delta\tilde{\vec{y}} - \Delta\vec{y} = \vec{n}$

where $\vec{n} \triangleq$ vector of observation errors

The properties of the Gaussian random vector of observation errors $\vec{n}$ have been extensively discussed in Section 3.2. Most importantly it should be remembered that, from Equation (3.38), the moment matrix of this random vector $N = \overline{\vec{n}\,\vec{n}^T}$ is also the covariance matrix of the observation errors. Additionally, recall that the weighting matrix in the weighted least squares cost functional is assumed to be the inverse of the covariance matrix of the observation errors, such that

$$W = N^{-1} \tag{3.91c}$$

Using the above information, Equation (3.91a) can be written as

$$\vec{\varepsilon} = (F^T WF)^{-1} F^T W\vec{n} \tag{3.92}$$

By defining the covariance matrix of the estimation errors as

$$\begin{aligned} \Xi &= \overline{\vec{\varepsilon}\,\vec{\varepsilon}^T} \\ &= (F^T WF)^{-1} F^T W \overline{\vec{n}\,\vec{n}^T} W^T F(F^T WF)^{-T} \end{aligned} \tag{3.93}$$

In the above expression W and $F^T WF$ are both symmetric matrices so

$$W = W^T$$

$$(F^T WF)^{-1} = (F^T WF)^{-T}$$

Therefore, the above covariance equation can be given by

$$\Xi = (F^T WF)^{-1} F^T W \overline{\vec{n}\,\vec{n}^T} W F(F^T WF)^{-1}$$

But from Equations (3.38) and (3.91c)

$$\overrightarrow{\vec{n}\,\vec{n}^T} = N = W^{-1}$$

and the above expression becomes

$$\Xi = (F^T WF)^{-1} F^T \underbrace{W W^{-1}}_{I} WF(F^T WF)^{-1}$$

$$= (\underbrace{F^T WF)^{-1} (F^T WF}_{I}) (F^T WF)^{-1}$$

$$\therefore \quad \Xi = (F^T WF)^{-1} \tag{3.94}$$

Thus, as expected in the linearized method, the covariance matrix of the estimation errors is equal to the inverse of the so-called normal matrix. In this case, of course, the matrix F is evaluated at the converged "best estimate" of the solve-for vector $\vec{x}^*$, such that

$$F = \left.\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}\right|_{\vec{x} = \vec{x}^*} \tag{3.95}$$

The matrix W is the inverse of the known covariance matrix of observation errors.

In the case of the solve-for vector being the epoch position and velocity vectors, the estimation error vector $\vec{\varepsilon}$ can be written as

$$\vec{\varepsilon} = \begin{pmatrix} \hat{\vec{r}} - \vec{r} \\ \hline \hat{\dot{\vec{r}}} - \dot{\vec{r}} \end{pmatrix} = \begin{pmatrix} \vec{\varepsilon}_r \\ \hline \vec{\varepsilon}_v \end{pmatrix}$$

Substituting this into Equation (3.93), the covariance matrix is then given by

$$\Xi = \overrightarrow{\varepsilon\,\varepsilon^T} = \overline{\begin{pmatrix} \vec{\varepsilon}_r \\ \hline \vec{\varepsilon}_v \end{pmatrix} \left(\vec{\varepsilon}_r \mid \vec{\varepsilon}_v\right)}$$

$$= \left(\begin{array}{c|c} \overrightarrow{\varepsilon_r\,\varepsilon_r^T} & \overrightarrow{\varepsilon_r\,\varepsilon_v^T} \\ \hline \overrightarrow{\varepsilon_v\,\varepsilon_r^T} & \overrightarrow{\varepsilon_v\,\varepsilon_v^T} \end{array}\right) \qquad (3.96)$$

Using the notation of Battin (28), the trace of the upper left hand 3x3 matrix will be given by

$$\mathrm{tr}\left(\overrightarrow{\varepsilon_r\,\varepsilon_r^T}\right) = \overline{\varepsilon_{rx}^2} + \overline{\varepsilon_{ry}^2} + \overline{\varepsilon_{rz}^2} \qquad (3.97a)$$

and will be referred to as the mean-squared position error. The square root of this trace will be called the root mean square (rms) position error. Similarly, the trace of the lower right hand 3x3 matrix is

$$tr(\overline{\vec{\varepsilon}_v \vec{\varepsilon}_v^T}) = \overline{\varepsilon_{vx}^2} + \overline{\varepsilon_{vy}^2} + \overline{\varepsilon_{vz}^2} \qquad (3.97b)$$

and will be called the mean-squared velocity error, with its square root defined as the rms velocity error.

3.5 Algorithmic Architecture

As in the previous chapter, the algorithm presented here is designed to take full advantage of the parallel processor hardware architecture. This section will outline the software architectural features of the PVM method. The major software modules involved in a single iteration of the method will be described. The parallel flow of the algorithm will be demonstrated for both the direct integration method and the linearization method for the cost and gradient evaluation.

3.5.1 Software Description

The Parallel Variable Metric orbit estimation algorithm has been implemented in a relatively flexible stand-

alone software package. The package will accept input in Cartesian, Keplerian or equinoctial elements, will integrate either the Cowell or the Variation of Parameters equations of motion and will solve for epoch position and velocity components, epoch Keplerian elements or epoch equinoctial elements as the solve-for vector. The implementation includes the Picard-Chebyshev parallel orbit propagation algorithm as the deterministic model of the satellite motion. The following describes the processing tasks and primary data arrays in a single iteration of the method. Table 3.15 provides a brief description of the software modules involved in this implementation.

The basic input parameters for the PVM algorithm are the following:

XAP $\triangleq$ vector of *a priori* solve-for parameters

EPSILN $\triangleq$ convergence criterion for the method

C1DIM $\triangleq$ convergence criterion for the univariate search technique

SCLFAC $\triangleq$ vector of scale factors used to adjust the magnitude of the basis vectors $\hat{\sigma}_i$ ($i = 1,2,\ldots,p$)

NSLV $\triangleq$ number of solve-for parameters (n)

NPAR $\triangleq$ number of parallel processors in the assumed architecture (p). Note: In this development it is always assumed that NSLV $\leq$ NPAR.

Table 3.15. Software Module Descriptions for the Stand-Alone Parallel Variable Metric Orbit Estimation Algorithm (Page 1 of 4)

Name	Description
CGINTG	Compute the value of the cost functional and its gradient vector using the direct integration method.
CGLINE	Compute the value of the cost functional and its gradient vector using the linearization method.
CMPTOB	Compute the predicted observables in the estimation problem, given the i^{th} iterate of the solve-for vector $\vec{x}^{i}$.
COVMTX	Update the *a priori* estimation error covariance matrix of the solve-for parameters.
EPHEX	Control the execution of an ephemeris generation run using the Picard-Chebyshev integration algorithm.
ESTEX	Compute the vector $\vec{x}_{min}$ which minimizes some scalar cost functional $J(\vec{x})$ using Straeter's Parallel Variable Metric algorithm.

Table 3.15. Software Module Descriptions for the Stand-Alone Parallel Variable Metric Orbit Estimation Algorithm (Page 2 of 4)

Name	Description
INPRD1	Compute the inner product of two vectors defined on a set of basis vectors, assuming the magnitudes of the basis vectors are not unity.
INPROD	Compute the inner product of two vectors defined on a set of basis vectors, assuming the magnitudes of the basis vectors are unity.
MAIN	Overall driver routine for the Parallel Processor Orbit Determination System.
MATINV	Invert a matrix (Fortran Scientific Subroutine Package subroutine 'MINV').
M6666	Multiply two 6x6 matrices.
OBSOUT	Output the observation processing report.
OUTPRD	Compute the outer product of two vectors.

Table 3.15. Software Module Descriptions for the Stand-Alone Parallel Variable Metric Orbit Estimation Algorithm (Page 3 of 4)

Name	Description
PAETAK	Compute the partial derivatives of a set of equinoctial orbit elements with respect to their corresponding Keplerian elements.
PATAØ	Compute the matrix of analytical partial derivatives of the current equinoctial orbital elements with respect to the epoch equinoctial orbit elements.
PTLOUT	Output the observation partial derivative report.
PXTEQ	Compute the partial derivatives of the spacecraft position and velocity with respect to the vector of equinoctial orbital elements.
PXTSLV	Compute the partial derivatives of the current position and velocity with respect to the solve-for parameters for use in the estimation process.

Table 3.15. Software Module Descriptions for the Stand-Alone Parallel Variable Metric Orbit Estimation Algorithm (Page 4 of 4)

Name	Description
RUNSET	Process the input card images and initialize the appropriate common blocks for running the Parallel Processor Orbit Determination System.
SERCH1	Compute the unconstrained minimum of a nonlinear function $F(\alpha)$ of a single variable α.

Note: Those subroutines described in Table 2.2 under the stand-alone version of the Picard-Chebyshev integration algorithm must also be included here to give the total stand-alone software system description.

The executive subroutine for the parallel orbit estimation algorithm is subroutine ESTEX. This subroutine reads the input card deck and error-checks the input parameters. It then establishes the set of basis vectors which span the parameter space. Following Straeter (139), these basis vectors are taken to be

$$\hat{\sigma}_j = \text{SCLFAC} * \text{EPSILN} * \hat{e}_j \qquad (3.98)$$

$$\text{for } j = 1,2,\ldots,n$$

where $\hat{e}_j$ is the j^{th} standard unit basis vector. These basis vectors are stored in array [BASIS] as shown in Figure 3.7. Additionally, the current "best estimate" of the solve-for vector $\vec{x}^i$ is stored in array [XAP] as

$$[\text{XAP}] = \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_n^i \end{bmatrix} \qquad (3.99)$$

$$[\text{BASIS}] = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & \cdots & \sigma_{n,n} \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \hat{\sigma}_1 & \hat{\sigma}_2 & \cdots & \hat{\sigma}_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \end{bmatrix}$$

Figure 3.7. Structure of the [BASIS] Array for Storing the Parameter Space Basis Vectors

Subroutine ESTEX now computes the value of the cost functional and its gradient vector based on the current "best estimate" of the solve-for vector stored in array [XAP]. The gradient vector components are stored in array [XGNTH] as

$$[\text{XGNTH}] = \begin{bmatrix} g_1(\vec{x}^i) \\ g_2(\vec{x}^i) \\ \vdots \\ g_n(\vec{x}^i) \end{bmatrix}$$

If the inner product of this vector with itself is less than the input convergence criterion (EPSILN) squared, then the method is considered to have converged and array [XAP]

contains the converged solve-for vector which minimizes the weighted least squares cost functional. If not, computation continues. It should be noted here that this evaluation of the cost and gradient requires a parallel numerical integration of the equations of motion based on the current "best estimate" of the solve-for vector $\vec{x}^{i}$. This integration is accomplished by subroutine CGINTG and must be done in both the direct method and the linearized method of evaluating the cost and gradient. In the linearized method this trajectory becomes the reference trajectory for the subsequent linearizations.

Subroutine ESTEX now computes the n new values of the solve-for vector given by Equation (3.5) and stores them in array [XMAT]. The structure of this array is depicted in Figure 3.8. At this point, Straeter's original description of the PVM method calls for n parallel evaluations of the cost functional and its gradient vector at the n values of the solve-for vector stored in array [XMAT]. When explicit expressions for the cost functional and gradient components exist as functions of the solve-for vector $\vec{x}$, these cost and gradient evaluations can be accomplished by a SIMD parallel processor with n parallel data streams. However, when no explicit functions of this type exist and

$$[XMAT] = \begin{bmatrix} x_1^{i+\sigma_{1,1}} & x_1^{i+\sigma_{1,2}} & \cdots & x_1^{i+\sigma_{1,n}} \\ x_2^{i+\sigma_{2,1}} & x_2^{i+\sigma_{2,2}} & \cdots & x_2^{i+\sigma_{2,n}} \\ \vdots & \vdots & \cdots & \vdots \\ x_n^{i+\sigma_{n,1}} & x_n^{i+\sigma_{n,2}} & \cdots & x_n^{i+\sigma_{n,n}} \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & & \vdots \\ \vec{x}^{i+\hat{\sigma}_1} & \vec{x}^{i+\hat{\sigma}_2} & \cdots & \vec{x}^{i+\hat{\sigma}_n} \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

$$[XMAT] = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \vec{x}_1 & \vec{x}_2 & \cdots & \vec{x}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

Figure 3.8. Structure of [XMAT] Array
for Storing n Parallel Values of the Solve-For Vector

the solve-for parameters and the cost and gradient values are related only by numerical integration and that integration is performed by the parallel Picard-Chebyshev technique, then this computation cannot be performed on a pure SIMD parallel processor. This is because the parallel integration method would be embedded within each of the n parallel streams involved in the cost and gradient computations. Thus it would take at least two instruction streams to control these interacting processes, one to control the estimation process and a second to control the subordinate integration process. The existence of two simultaneous instruction streams in this case would violate the basic assumption of SIMD hardware architecture.

A potential solution to this problem is to assume a second level-of-parallelism in the hardware architecture, making the architecture essentially that of a Multiple Instruction Stream/Multiple Data Stream (MIMD) parallel processor. This approach is taken in the work of Travassos and Kaufman, in which they assume three distinct levels-of-parallelism (146, 147). A second approach to this problem is to compute the integrated trajectories one at a time and process the observations associated with each trajectory in parallel. A third possibility is to use the linearized method and compute the single integrated trajectory via the

parallel Picard-Chebyshev integration method and then process the observation data and cost and gradient values for the solve-for vectors $\vec{x}_j$, (j = 0,1,...,n) in n parallel streams. The first approach abandons the initial assumption of SIMD architecture while the second and third approaches retain the SIMD hardware character. A block diagram description of these approaches will be presented in Section 3.5.2 and their relative performance will be discussed in Chapter 4.

At this point, it will be assumed that the n values of the cost functional and its gradient vector have been computed. These values are stored in arrays [XJATXJ] and [XGATXJ] as shown in Figure 3.9. Subroutine ESTEX then computes in parallel the n values of the vector $\vec{y}_j$ (j = 1,2,...,n) from Equation (3.8). These values are stored in the columns of array [YMAT] as shown in Figure 3.10. Now the algorithm reverts to sequential processing to compute the n residual vectors given by Equations (3.9a) and (3.9b). Additionally, the n scalars τ_j, (j = 1,2,...,n) are computed from Equation (3.10).

$$[\text{XJATXJ}] = \begin{bmatrix} J(\vec{x}_1) \\ J(\vec{x}_2) \\ \vdots \\ J(\vec{x}_n) \end{bmatrix}$$

$$[\text{XGATXJ}] = \begin{bmatrix} g_1(\vec{x}_1) & g_1(\vec{x}_2) & \cdots & g_1(\vec{x}_n) \\ g_2(\vec{x}_1) & g_2(\vec{x}_2) & \cdots & g_2(\vec{x}_n) \\ \vdots & \vdots & \cdots & \vdots \\ g_n(\vec{x}_1) & g_n(\vec{x}_2) & \cdots & g_n(\vec{x}_n) \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & & \vdots \\ \vec{g}_1 & \vec{g}_2 & \cdots & \vec{g}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

Figure 3.9. Structure of [XJATXJ] and [XGATXJ] Arrays for Storage of n Evaluations of the Cost Functional and Gradient Vector

$$[YMAT] = \begin{bmatrix} g_1(\vec{x}_1)-g_1(\vec{x}^i) & g_1(\vec{x}_2)-g_1(\vec{x}^i) & \cdots & g_1(\vec{x}_n)-g_1(\vec{x}^i) \\ g_2(\vec{x}_1)-g_2(\vec{x}^i) & g_2(\vec{x}_2)-g_2(\vec{x}^i) & \cdots & g_2(\vec{x}_n)-g_2(\vec{x}^i) \\ \vdots & \vdots & \cdots & \vdots \\ g_n(\vec{x}_1)-g_n(\vec{x}^i) & g_n(\vec{x}_2)-g_n(\vec{x}^i) & \cdots & g_n(\vec{x}_n)-g_n(\vec{x}^i) \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & & \vdots \\ \vec{g}(\vec{x}_1)-\vec{g}(\vec{x}^i) & \vec{g}(\vec{x}_2)-\vec{g}(\vec{x}^i) & \cdots & \vec{g}(\vec{x}_n)-\vec{g}(\vec{x}^i) \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

$$[YMAT] = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \vec{y}_1 & \vec{y}_2 & \cdots & \vec{y}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

Figure 3.10. Structure of [YMAT] Array for Storage of the n Gradient Vector Differences

Returning to parallel processing, subroutine ESTEX computes in parallel the n matrices $B^{(j)}$, (j = 1,2,...,n) given by Equation (3.14) and the n products of $\tau_j B^{(j)}$ (j = 1,2,...,n). Then in a sequential process, the metric matrix is updated via Equation (3.11) and the new search direction vector is found from Equation (3.15). It now remains to find the minimum of the cost functional along the just determined direction of search $\vec{s}_i$.

3.5.1.1 Univariate Search Technique

The univariate search technique is another area in which the linearization method may be employed to advantage. As described in Section 3.1.1.1, Coggins algorithm is composed of two parts. The first part rapidly brackets the minimum of the cost functional along the search direction with three points. At a minimum this requires three cost functional evaluations and these evaluations must be done in sequence, since the starting conditions for the next evaluation depend on the results of the current evaluation. In the direct integration method this becomes very costly, since each cost function evaluation requires a full numerical integration. In contrast, however, the linearization process is relatively efficient here.

The univariate search is controlled by subroutine SERCH1. Its primary input data are:

XN $\triangleq$ vector containing current iterate of the solve-for vector $\vec{x}^i$

SN $\triangleq$ vector containing new search direction vector $\vec{s}_i$

N $\triangleq$ dimension of the solve-for vector

ALFA1 $\triangleq$ initial guess for the minimizing value of the scalar α_i such that

$$J(\vec{x}^i + \alpha_i \vec{s}_i) = \text{Min}$$

STPSZ $\triangleq$ initial increment in α in the search process

CNVG $\triangleq$ convergence criterion for the method

In completing the first portion of the univariate search by the linearization method, subroutine SERCH1 calls subroutine CGINTG, which computes the value of the cost functional at

$$\vec{x} = \text{XN} + \text{ALFA1} * \text{SN}$$

using the direct integration method. Subsequent evaluations of the cost functional at values of the solve-for vector are given by

$$\vec{x} = XN + [ALFA1 + STPSZ] * SN$$

are computed using the linearization assumptions given in Section 3.3.4. Because the value of STPSZ is dynamically altered as the search progresses, the validity of the linearization process cannot be guaranteed as it could in the previous application of the method. However, in extensive empirical testing no case has been found in which the linearity assumption has failed.

With the minimum bracketed by three points, the second part of Coggins method fits a quadratic through these three points and analytically solves for the minimizing value of alpha. If this new value of alpha differs from the previous three values by less than the input convergence criterion, then the computation is terminated and the value of alpha just determined is considered to be the required minimizing value. If not, the value of alpha resulting in the largest cost value is replaced by the new value of alpha, the cost at this new value is computed, and a new quadratic

function is fit to these points. Here again the linearization method has been applied. A direct integration evaluation of the cost functional is performed for the first value of alpha determined by the above process. Any subsequent values of the cost functional are determined by the linearization method, using the first trajectory as a reference for the linearization.

In the above manner, the maximum number of integrations needed in any one PVM iteration using the linearization method is three. This is less than half the minimum number of integrations required by the direct integration method, even assuming the MIMD hardware architecture discussed above. The results of this are some relatively impressive performance characteristics for the PVM estimation method coupled with the linearization scheme for cost and gradient evaluations. These performance characteristics will be discussed further in Chapter 4.

3.5.2 Algorithm Parallelism

In the above software description, it was pointed out that, when the PVM method was used in conjunction with the direct integration method, the basic assumption of SIMD hardware architecture was violated. Three alternatives

were presented to alleviate this problem. This section uses simple block diagrams to illustrate the parallelism in each approach.

The first approach discussed is that of expanding the assumed hardware to encompass the Multiple Instruction Stream/Multiple Data Stream hardware architectures. Figure 3.11 presents a block diagram of this approach. The second level-of-parallelism allowed by the MIMD hardware architecture allows the "nesting" of the Picard-Chebyshev numerical integrations within the n parallel cost and gradient evaluations as shown. In this manner, the n evaluations of the cost and gradient can be completed using the direct integration method in the effective time required for a single cost and gradient evaluation if the algorithm were executed in a sequential manner. However, the hardware and software involved in managing these extra levels-of-parallelism may be extremely complex and MIMD processors face some significant organizational problems which may significantly impact their effectiveness (78). Additionally, the MIMD architecture provides no particular advantage in the univariate search because the integrations must be done in sequence and hence the second level-of-parallelism does not contribute to performance.

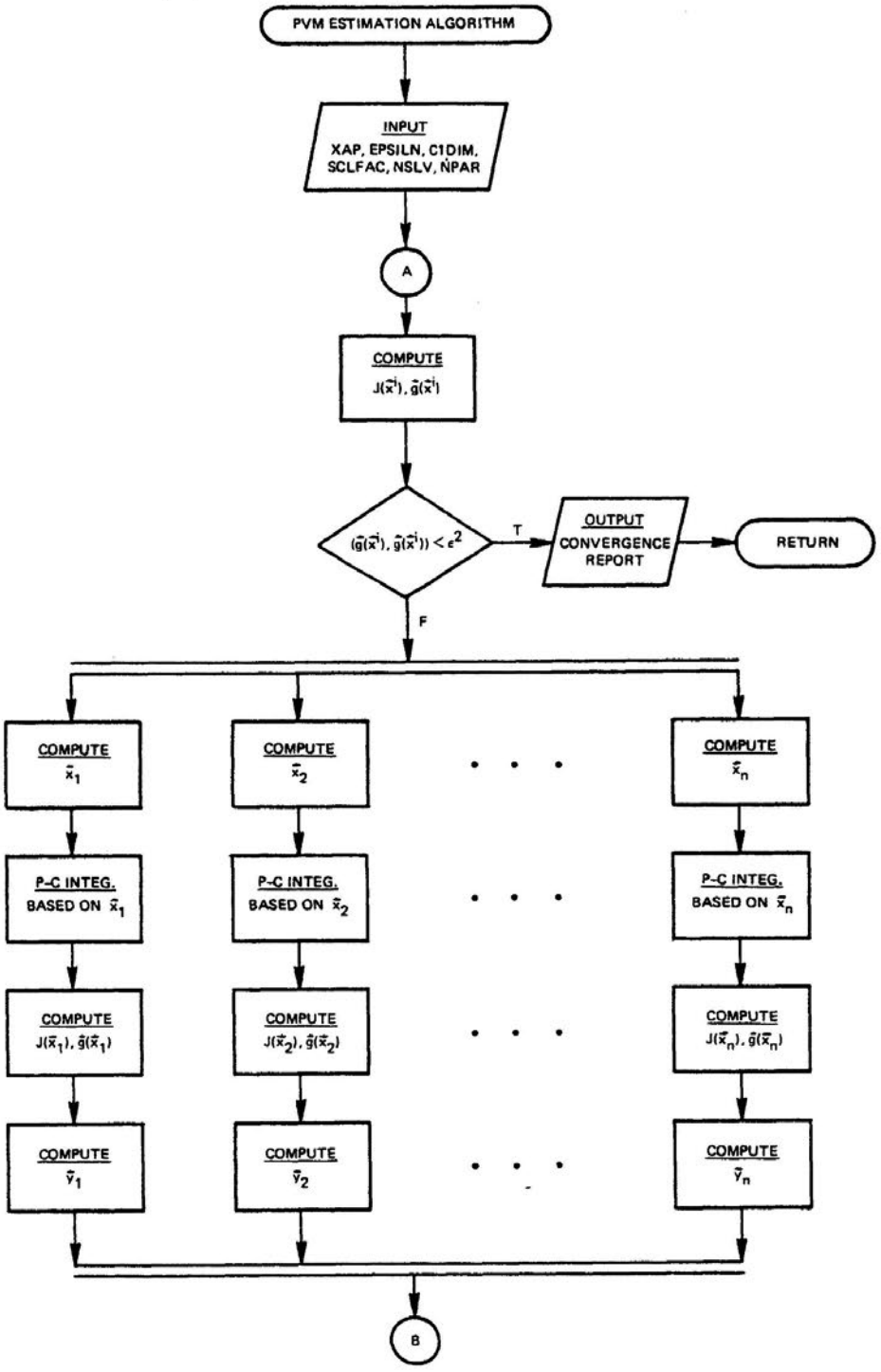


Figure 3.11. Direct Integration Method Implemented on MIMD Hardware Architecture (Page 1 of 2)

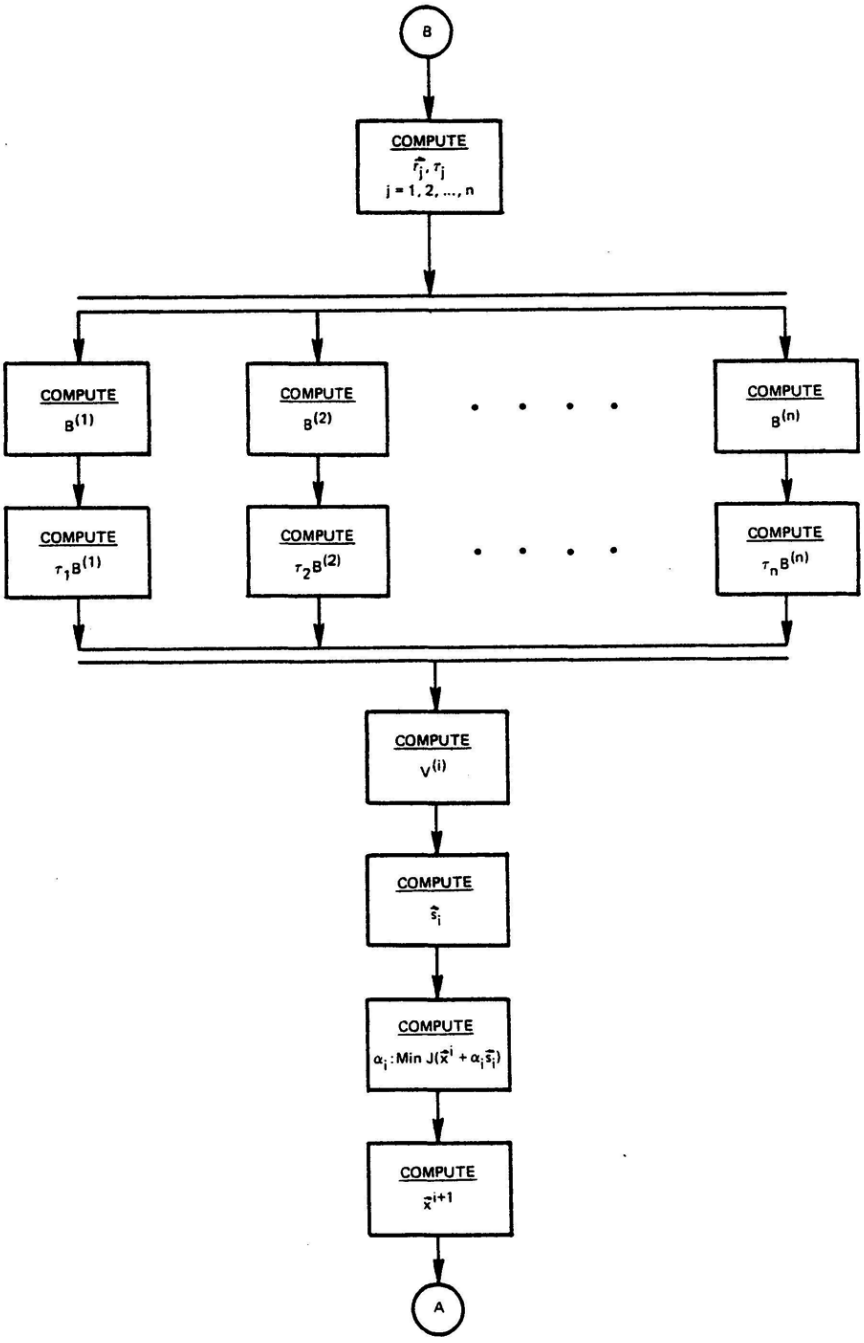


Figure 3.11. Direct Integration Method Implemented on MIMD Hardware Architecture (Page 2 of 2)

The second approach is illustrated in Figure 3.12. In this approach, the numerical integrations required for each of the n cost and gradient evaluations are performed in a sequential manner and then the necessary observation processing is done in parallel. To maintain the SIMD character of the algorithm, the observations processed in any one parallel execution must be of the same observation type. In the test cases presented in Section 3.6, the observations were all of the same type and hence no problems were encountered. However, in an operational implementation of this algorithm, additional software would be required to sort the entire "batch" of observations into sub-batches by observation types and then process the members of each batch in parallel. This approach is compatible with the SIMD hardware architecture originally assumed and provides some advantage in the univariate search due to the parallel observation processing. However, this method must still perform the expensive integrations in sequence.

The third approach suggested in the preceding section includes the linearization method described in Section 3.3.4. In this approach, a single Picard-Chebyshev numerical integration begins each iteration through the PVM method. Using the trajectory as a reference for the subsequent

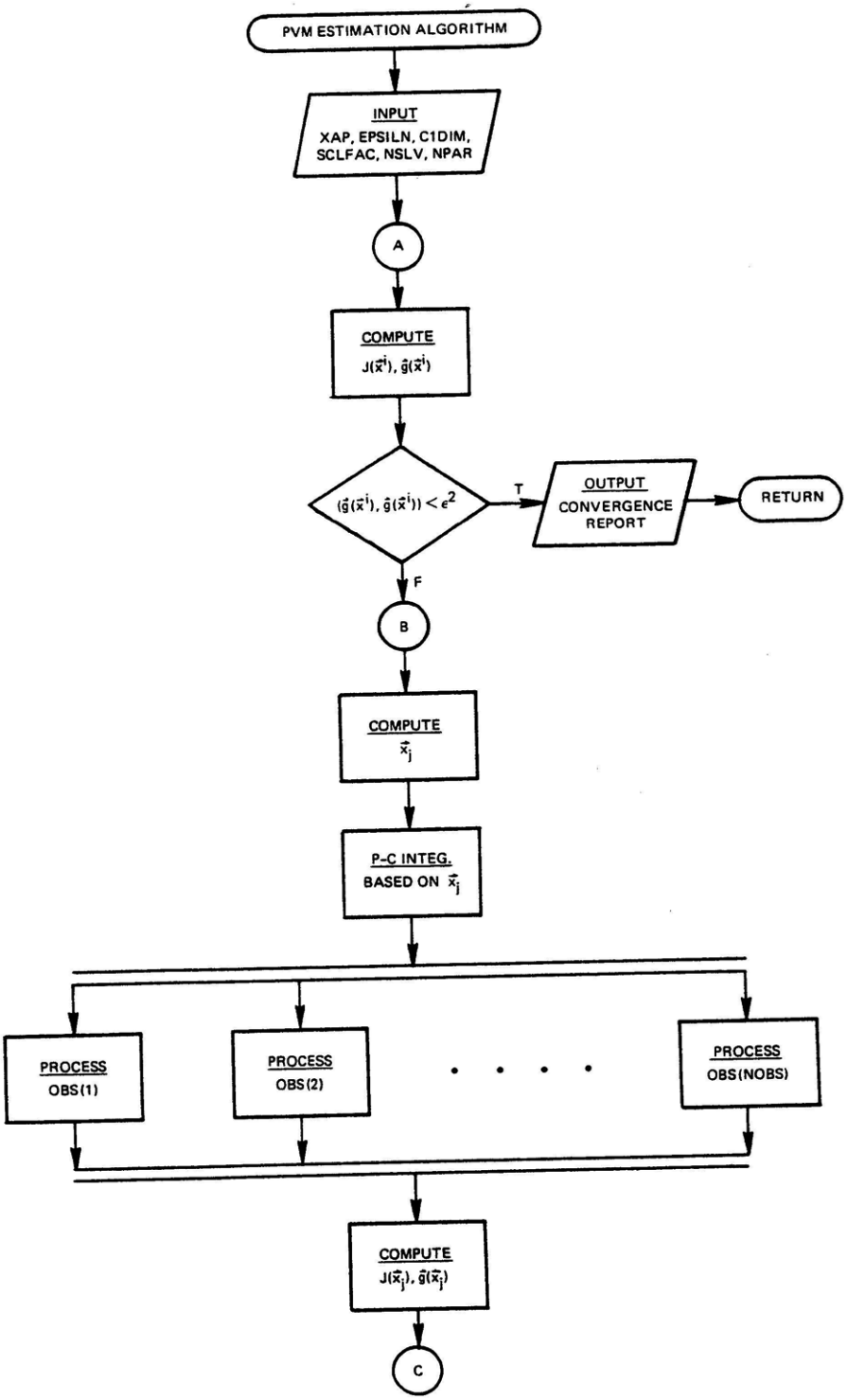


Figure 3.12. Direct Integration Method with Parallel Observation Processing (Page 1 of 3)

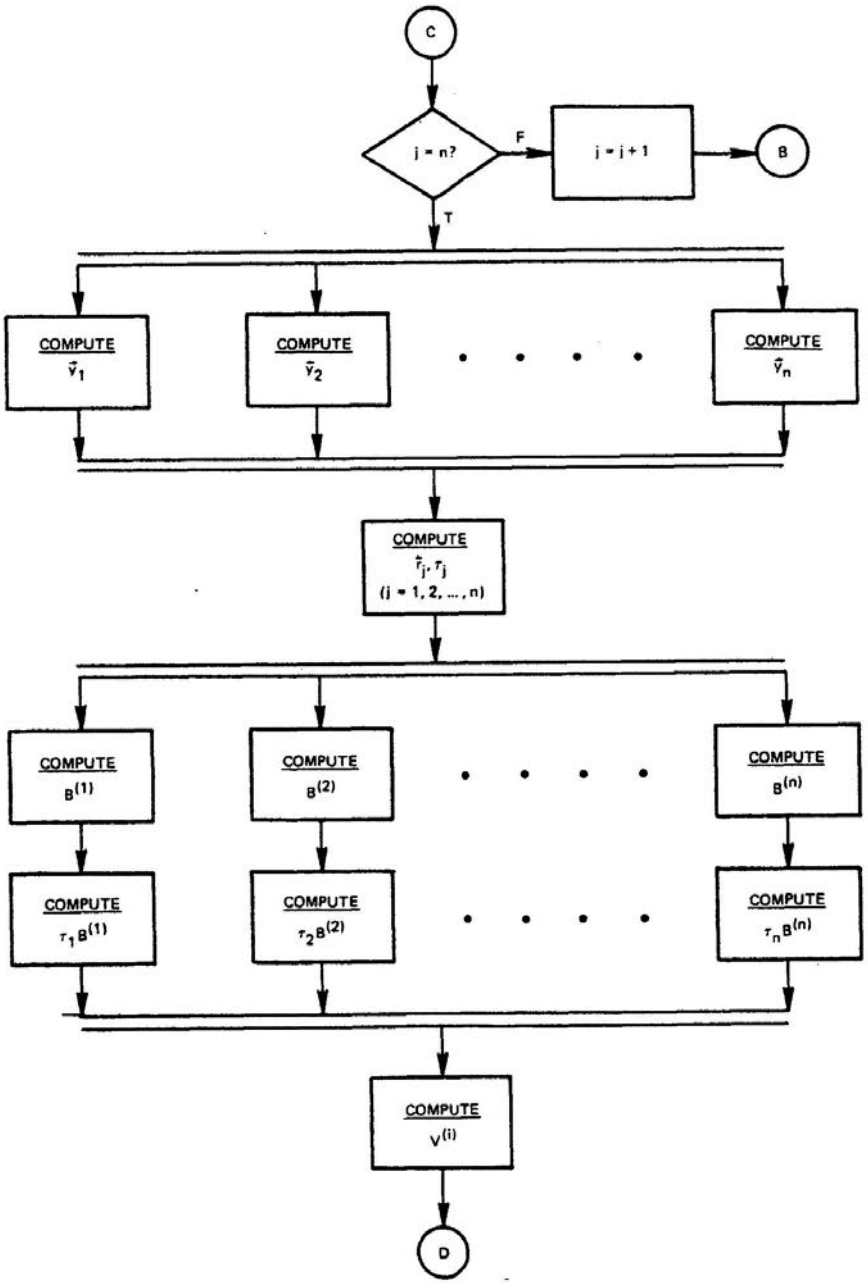


Figure 3.12. Direct Integration Method with Parallel Observation Processing (Page 2 of 3)

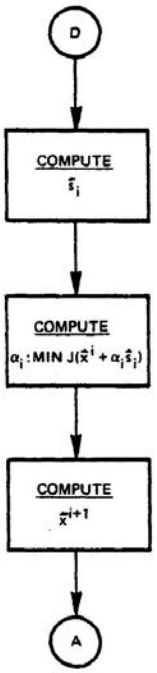


Figure 3.12. Direct Integration Method with Parallel Observation Processing (Page 3 of 3)

linearization, the n cost and gradient evaluations are computed in n parallel streams, as shown in Figure 3.13. Again this method preserves the SIMD nature of the algorithm. Additionally, it follows more closely the original algorithmic structure of Straeter. Finally, this approach provides significant advantage in the univariate search by greatly reducing the number of required integrations in that search. The block diagram of the univariate search is presented in Figure 3.14.

3.6 Numerical Results

The Parallel Variable Metric orbit estimation algorithm described in the previous sections has been exercised on a variety of test cases. This section will present some of the results of these test runs. Also included are descriptions of the major test cases and a brief discussion of the basic philosophy involved in the simulation of the observational data.

3.6.1 Basic Test Case Descriptions

In selecting test cases to demonstrate the quality of the estimation results from the Parallel Variable Metric (PVM) orbit estimation method, it was desired to use cases for which results had appeared in the recent literature.

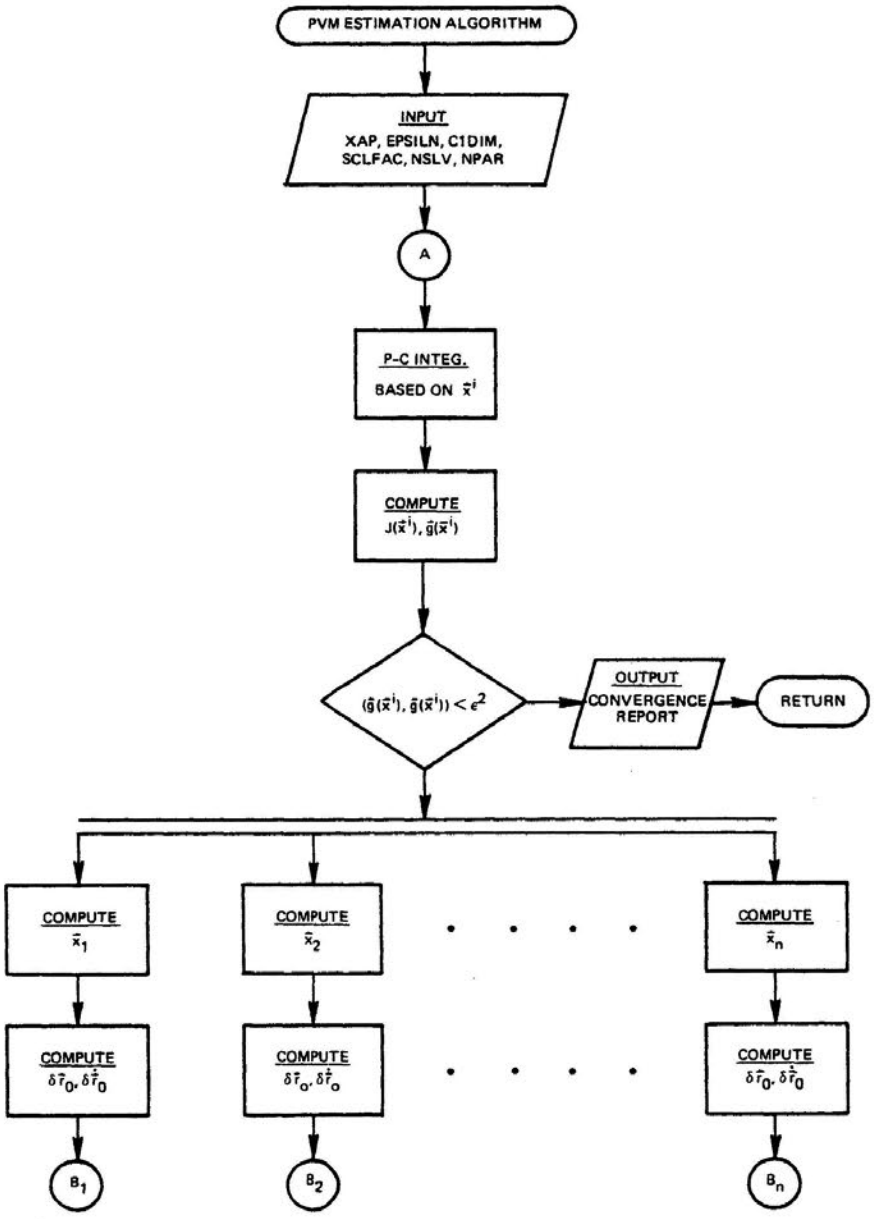


Figure 3.13. Lineariation Method Implemented on SIMD Hardware Architecture (Page 1 of 3)

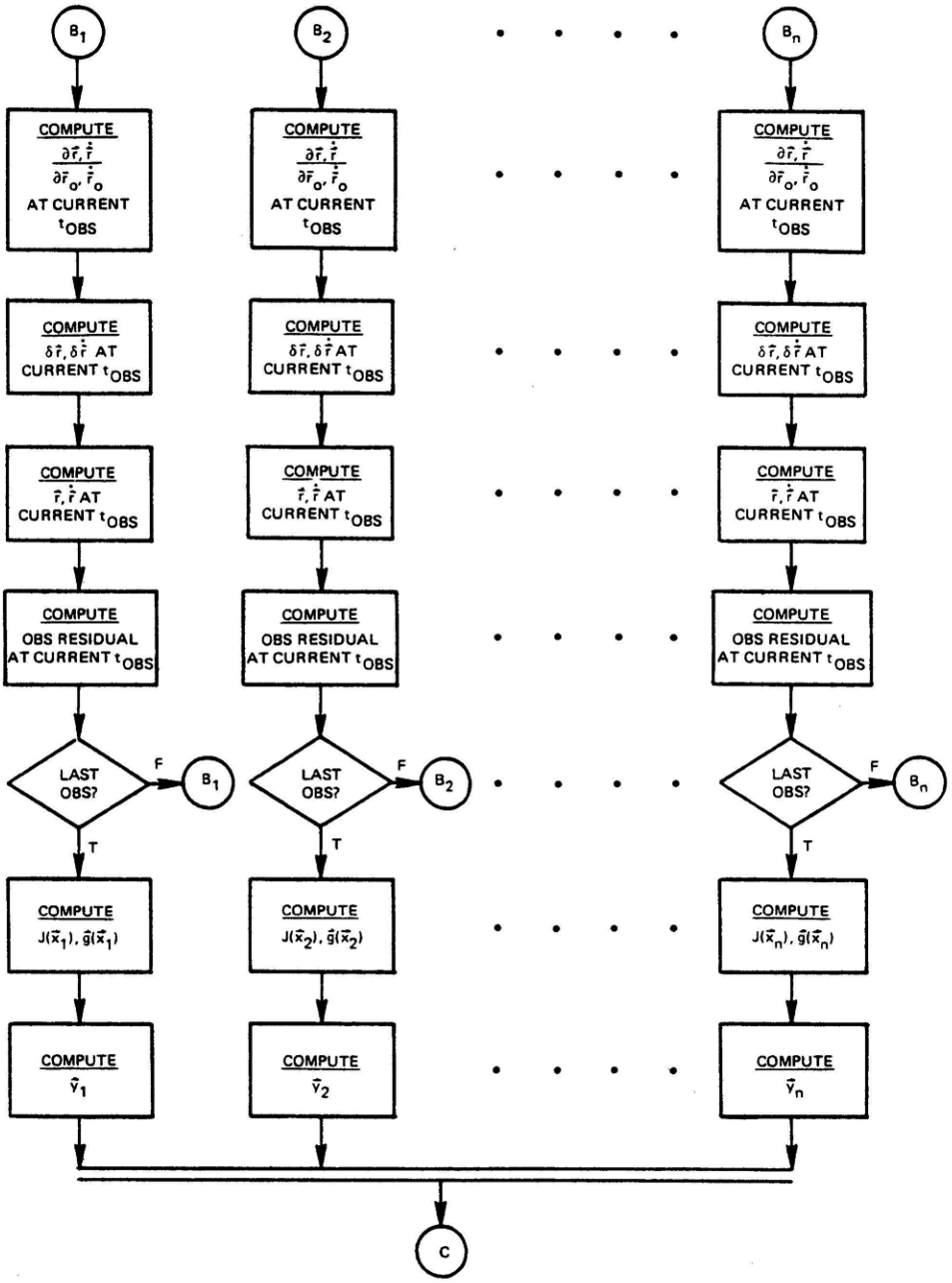


Figure 3.13. Linearization Method Implemented on SIMD Hardware Architecture (Page 2 of 3)

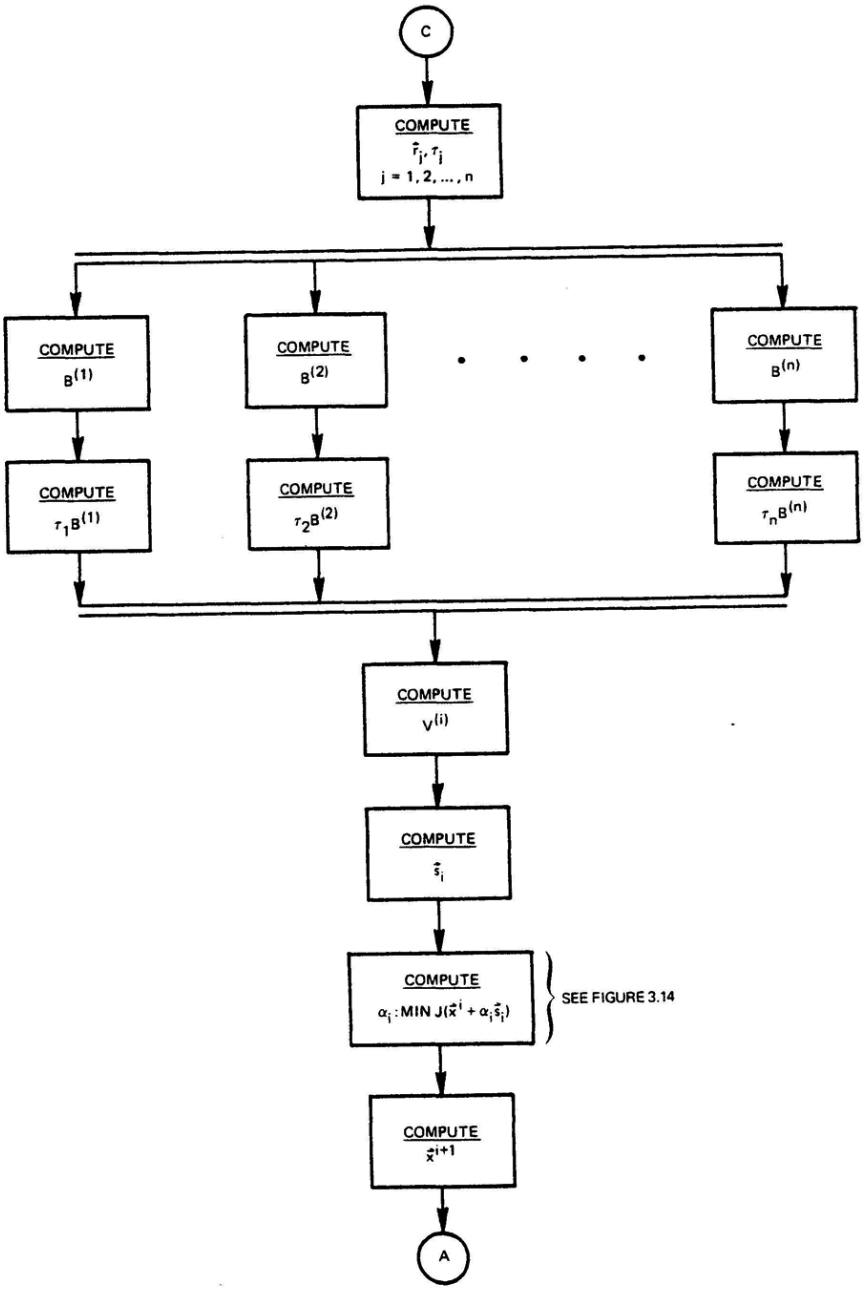


Figure 3.13. Linearization Method Implemented on SIMD Hardware Architecture (Page 3 of 3)

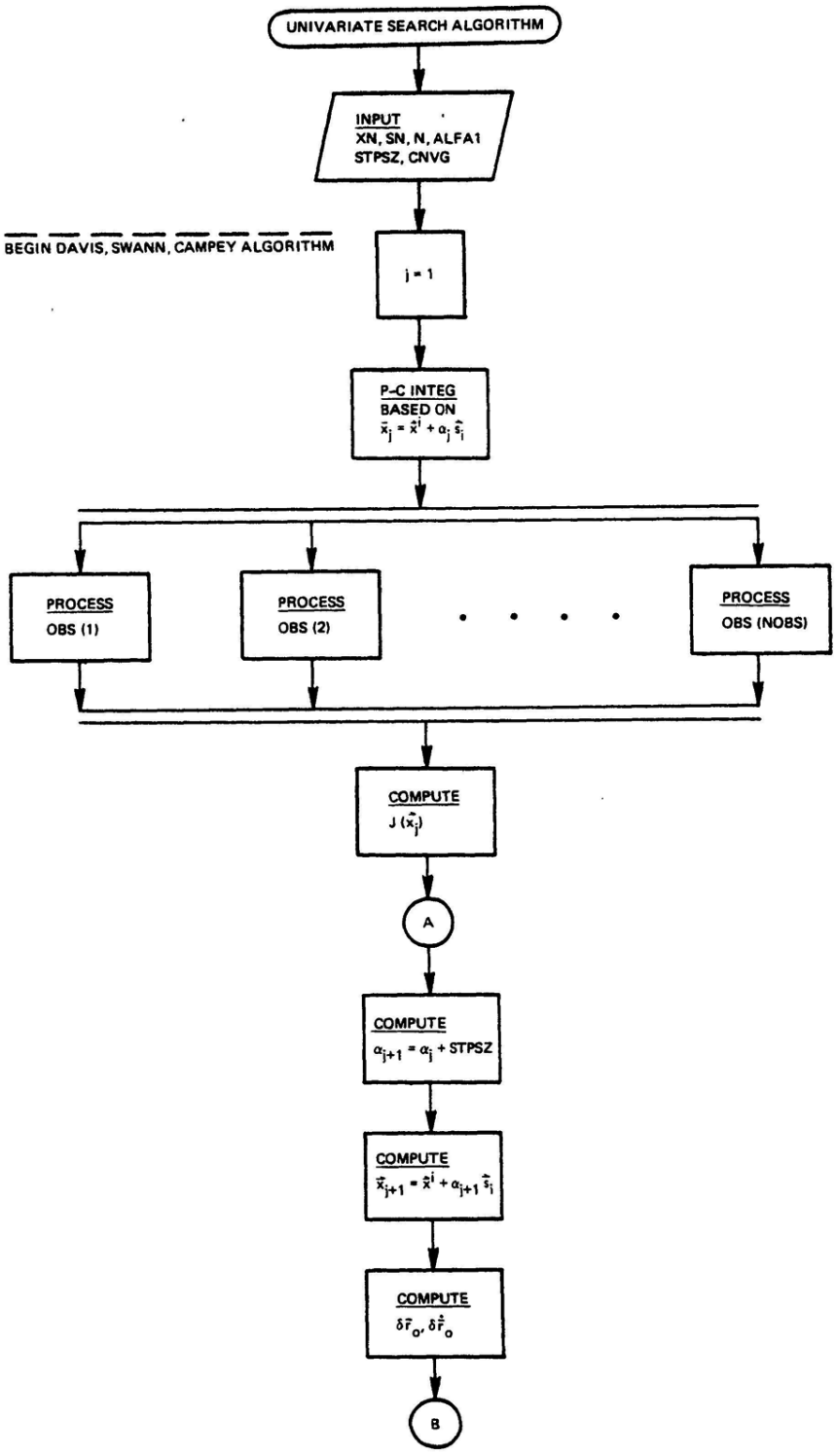


Figure 3.14. Coggins Univariate Search with Linearization and Parallel Observation Processing (Page 1 of 4)

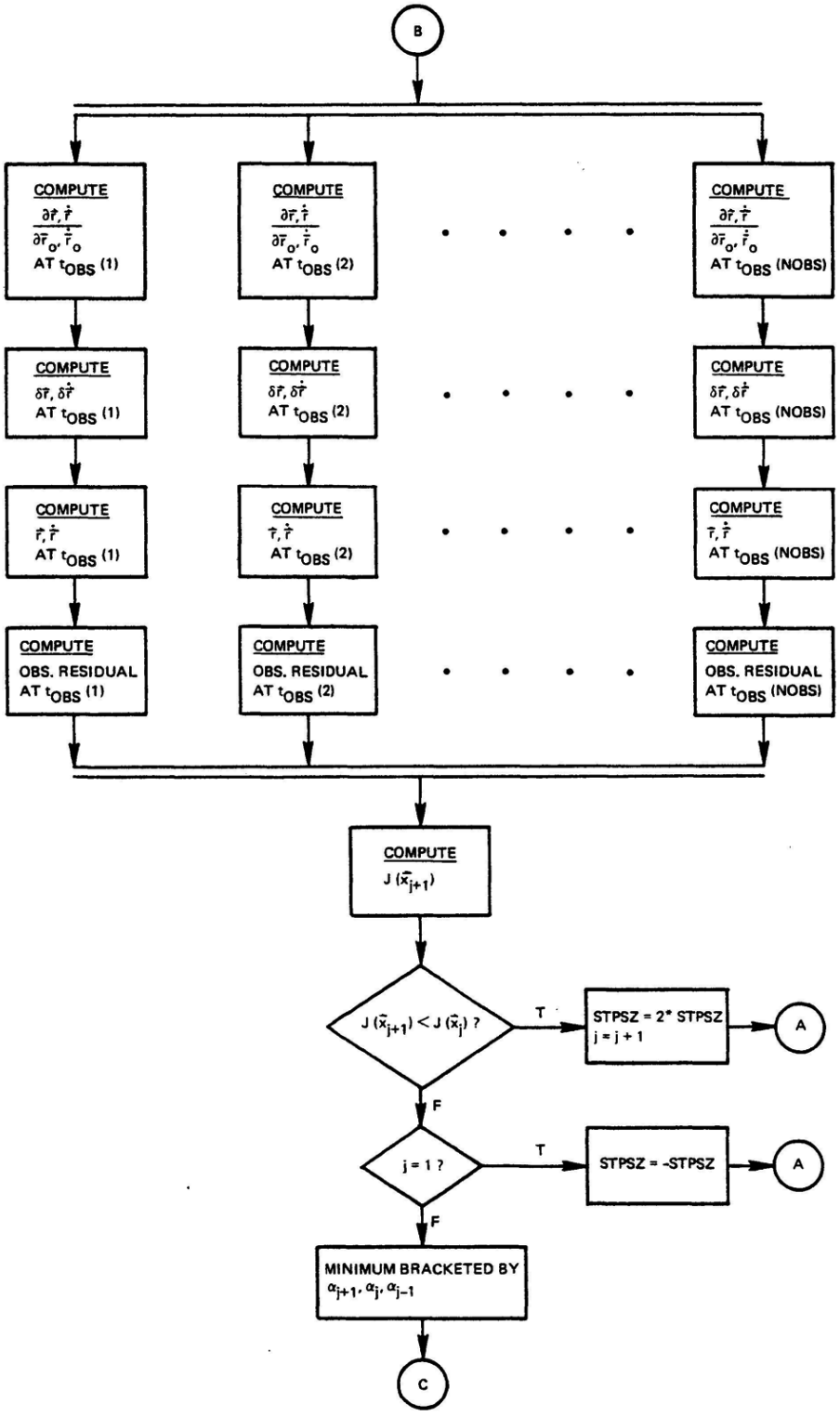


Figure 3.14. Coggins Univariate Search with Linearization and Parallel Observation Processing (Page 2 of 4)

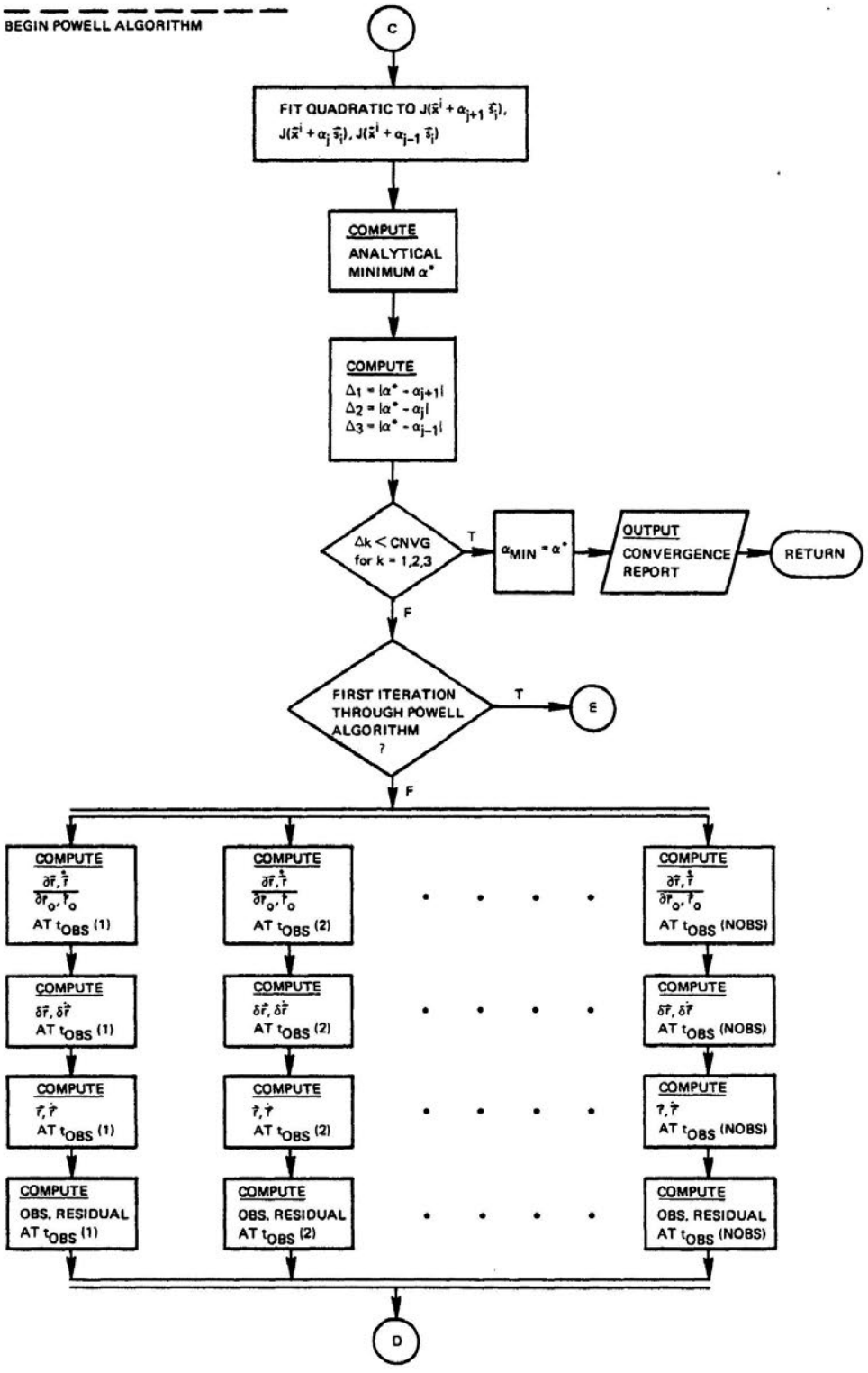


Figure 3.14. Coggins Univariate Search with Linearization and Parallel Observation Processing (Page 3 of 4)

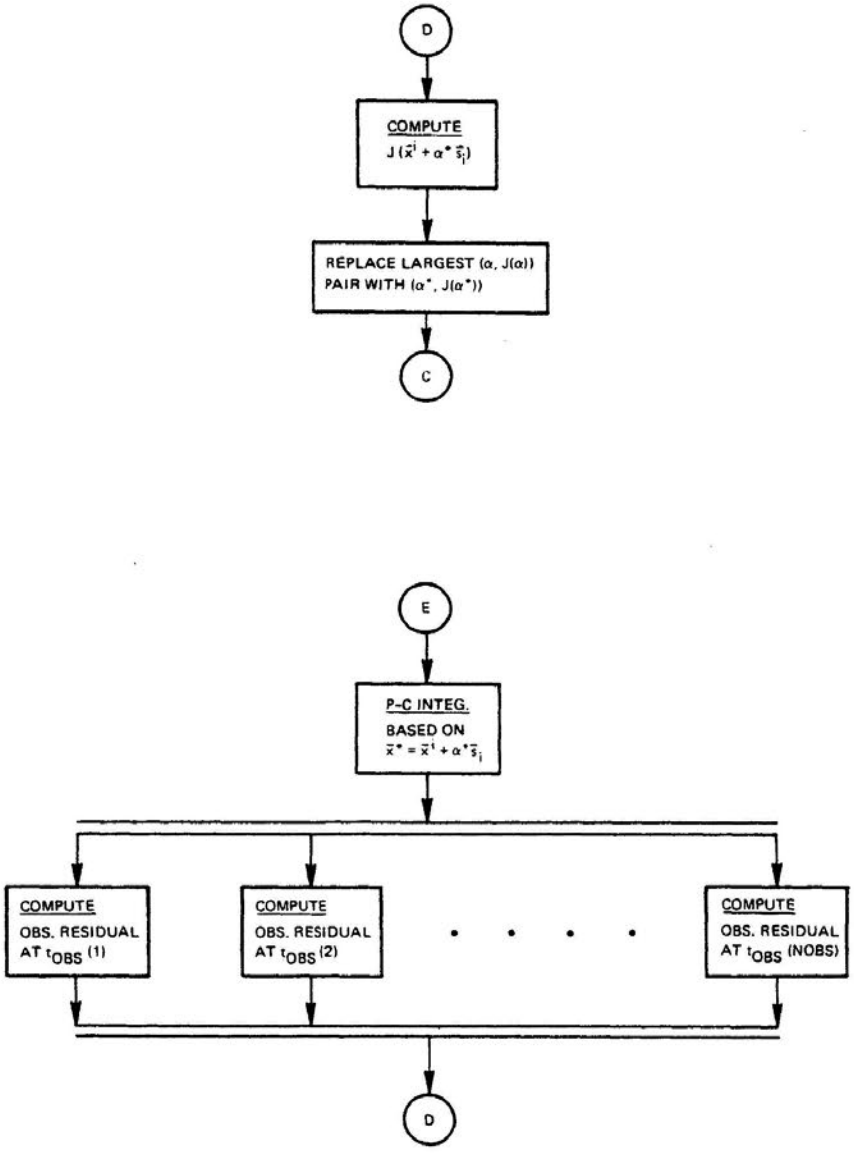


Figure 3.14. Coggins Univariate Search with Linearization and Parallel Observation Processing (Page 4 of 4)

In this light, the two primary test cases for this work were taken from papers by Battin, et al. (30) and Adimurthy and Joy (4). These papers provided good baseline data to verify the PVM results and allowed the PVM method to be compared against other estimation methods. Battin's paper derived the Epoch State Navigation Filter and compared it to the more conventional recursive Kalman Filter. Adimurthy and Joy compared the classical differential corrections method with several nonlinear optimization algorithms.

3.6.1.1 Battin Test Case

The primary test case used in the Epoch State Navigation Filter description was a circular equatorial Earth satellite at 100 nautical mile altitude. The perturbation force model for this test case was limited to the J_2 zonal harmonic. Drag was not included in the truth ephemeris. The epoch conditions for the truth ephemeris in Cartesian, Keplerian and equinoctial elements are presented in Table 3.16. These three sets of epoch elements represent the three different possible sets of solve-for parameters in the current implementation of the parallel orbit estimation algorithm.

Table 3.16

Truth Ephemeris Epoch Conditions for the Battin Test Case

True Epoch Position and Velocity Components

r_x = 6563.34 km	v_x = 0.0 km/sec
r_y = 0.0 km	v_y = 7.7930350894 km/sec
r_z = 0.0 km	v_z = 0.0 km/sec

True Epoch Keplerian Elements

a = 6563.34 km	ω = 0.0 deg
e = 0.0	Ω = 0.0 deg
i = 0.0 deg	M = 0.0 deg

True Epoch Equinoctial Elements

a = 6563.34 km	p = 0.0
h = 0.0	q = 0.0
k = 0.0	λ = 0.0 deg

References 30 and 70 point out that the components of the solve-for vector in those analyses are the epoch position and velocity components. The above references assume the initial covariance matrix to be diagonal with three equal position elements of (500 meters)^2 and three equal velocity elements of (5 meters/second)^2. Hence, in generating a set of *a priori* solve-for parameters, the differences between the true epoch position components and the *a priori* values were considered to be Gaussian random variables with zero mean and a standard deviation of 500 meters. Similarly, the differences between the true epoch velocity components and the *a priori* values were also considered to be Gaussian random values with zero mean and a standard deviation of 5 meters/second. A random number generator was used to determine values for these different quantitites, which were then added to the true epoch quantities to provide the *a priori* values of the epoch position and velocity components. These *a priori* epoch position and velocity components, and their associated Keplerian and equinoctial elements, are presented in Table 3.17.

It is interesting to note that the *a priori* values of the argument of periapsis (ω), the longitude of the ascending node (Ω) and the mean anomaly (M) are significantly

Table 3.17

A Priori Solve-for Vectors for the Battin Test Case

A Priori Epoch Position and Velocity Compoments

r_x = 6562.6199330 km	v_x = 0.0138754810 km/sec
r_y = 0.9349028 km	v_y = 7.7973261140 km/sec
r_z = 0.0309880 km	v_z = -0.0107589055 km/sec

A Priori Epoch Keplerian Elements

a = 6569.1681709 km	ω = 117.1067247 deg
e = 0.0021651	Ω = 180.2042328 deg
i = 0.0790588 deg	M = 62.4769087 deg

A Priori Epoch Equinoctial Elements

a = 6569.1681709 km	p = $-2.4592326 \times 10^{-6}$
h = $-1.9237360 \times 10^{-3}$	q = $-6.8991398 \times 10^{-4}$
k = 9.9338014×10^{-4}	λ = 359.7878622 deg

different from their true values of zero given in Table 3.16. This is due to the fact that these parameters are very poorly defined for an orbit with vanishing eccentricity and inclination such as the one defined by the *a priori* position and velocity vectors. In contrast, the *a priori* equinoctial elements remain close to their true values, as would be expected for the relatively small errors induced in the true epoch position and velocity vectors to form the *a priori* position and velocity. This is a reflection of the nonsingular nature of these elements and indicates that they may be a better element set to use in these types of estimation problems.

3.6.1.2 Adimurthy and Joy Test Case

Adimurthy and Joy examined the application of several nonlinear parameter optimization techniques to the determination of the orbital elements of a satellite from noisy ground observations. Their results indicated that some of these optimization methods have significantly wider radii of convergence than the classical differential correction method of orbit determination. The truth ephemeris for their study was based on the epoch Keplerian elements presented in Table 3.18. Also included in this table are the epoch position and velocity components and epoch equi-

Table 3.18
Truth Ephemeris Epoch Conditions
for the Adimurthy and Joy Test Case

True Epoch Position and Velocity Components

r_x = 27.8719455 km v_x = -6.04436809 km/sec

r_y = 6651.7422931 km v_y = 0.47406058 km/sec

r_z = 590.6414848 km v_z = -5.05359105 km/sec

True Epoch Keplerian Elements

a = 6983.208 km ω = 172.135 deg

e = 0.04371 Ω = 275.777 deg

i = 40.267 deg M = 0.0 deg

True Epoch Equinoctial Elements

a = 6983.208 km p = $-3.6474921 \times 10^{-1}$

h = 4.3680979×10^{-2} q = 3.6901950×10^{-2}

k = 1.5925480×10^{-3} λ = 87.912 deg

noctial elements associated with these epoch Keplerian elements.

Adimurthy and Joy made numerous estimation runs with a wide variety of initial guesses for the solve-for Keplerian elements. Their most stringent test case is summarized in Table 3.19. These values were used as the *a priori* solve-for vectors in the Parallel Variable metric estimation runs to be described later in this chapter.

3.6.2 Observation Simulation

The simulation of observational data for the test cases described above follows that method used by Battin, et al. (30) and Edwards (70) in their evaluation of the Epoch State Navigation Filter. In those efforts, the measurement geometry vector was chosen to be a unit vector along one of the inertial Cartesian coordinate axes. Thus, the observed quantities were the scalar components of the inertial position vector. The choice of scalar components was permuted cyclically to avoid any bias being introduced in the observations. The true observed quantities were found by propagation of the truth ephemeris from the known epoch conditions. These observations were then corrupted by observation errors generated by a random

Table 3.19

A Priori Solve-for Vectors

for the Adimurthy and Joy Test Case

A Priori Epoch Position and Velocity Components

r_x = 1117.6909353 km	v_x = -6.64966776 km/sec
r_y = 6636.7298905 km	v_y = 1.13749177 km/sec
r_z = -29.8751932 km	v_z = -3.89490176 km/sec

A Priori Epoch Keplerian Elements

a = 6900.00 km	ω = 170.00 deg
e = 0.025	Ω = 260.00 deg
i = 30.00 deg	M = 9.99565880 deg

A Priori Epoch Equinoctial Elements

a = 6900.00	p = $-2.6387844 \times 10^{-1}$
h = 2.3492316×10^{-2}	q = $-4.6528889 \times 10^{-2}$
k = 8.5505036×10^{-3}	λ = 79.99565880 deg

number generator with zero mean and a standard deviation of 1 km. For the circular test case used in the above references, eighty such measurements were generated at equal intervals of ten degrees of central angle.

Because of the simplicity of the above simulation technique, it has been adopted for this study. Two slight differences exist between this approach and that described above. First, the observation span is assumed to be three hours long and observations are taken every three minutes for a total of sixty observations instead of the eighty used above. Second, provision is included in this simulation to use components of the inertial velocity vector as observational data. It is assumed that observations of the position components provide simulated range-like observations. Analogously, it is assumed that use of a velocity component would provide a kind of simulated range-rate observation. It was felt that the influence of this type of observation on the estimation process might be of interest.

Truth ephemerides were generated for each of the test cases described in Section 3.6.1 using the Picard-Chebyshev integration method. The epoch conditions for these truth ephemerides are given in Tables 3.16 and 3.18. For the Battin test case, the perturbation force model was limited

to the J_2 zonal harmonic. Zonal harmonics J_2 through J_8 were included in the perturbation force model for the Adimurthy and Joy test case. The effects of drag on this problem were studied in a final set of test runs.

Once the truth ephemeris data had been generated, the true observed quantities were then selected as previously described. These true values were then corrupted by additive observation errors generated by a random number generator. These error values are Gaussian random variables with zero mean and selectable standard deviations. The different observation options for each primary test case are summarized in Table 3.20.

Test case TC-1.1 essentially replicates the primary test case of References 30 and 70, with the exception that TC-1.1 uses sixty observations instead of eighty.

None of the test cases in Table 3.20 exactly replicate the test data of the Adimurthy and Joy paper (4). This is because their force model includes drag effects and their observation model is for a radar giving range and angles information. Although the observational data are not equivalent, it is felt that the Adimurthy and Joy results for their case of 45 observations from a single pass

Table 3.20

Summary of Observational Data for Estimation Test Cases

Test Case	Force Model	Position Observation Standard Deviation	Velocity Observation Standard Deviation
TC-1 (Battin)			
TC-1.1	J_2	1.00 km	---
TC-1.2	J_2	1.00 km	0.010 km/sec
TC-2 (Adimurthy and Joy)			
TC-2.1	$J_2 - J_8$	0.03 km	---
TC-2.2	$J_2 - J_8$	0.03 km	0.002 km/sec

represent a good basis of comparison for the results of the PVM algorithm presented here. Test case TC-2.1 uses a 3-sigma error in range measurments of 0.1 km, as suggested in Reference 4. Similarly, test case TC-2.2 employs a 3-sigma error in range measurements of 0.1 km and a 3-sigma error in range-rate measurements of 6 m/sec.

3.6.3 Battin Test Case Results

Several variations on the basic Battin test case were run using the Parallel Variable Metric (PVM) estimation method and some interesting results were obtained. This section will describe the results, discuss their significance and compare the PVM results to the Epoch State Navigation Filter (ESF) results.

The first PVM option tested was the direct integration method. To remain consistent with References 30 and 70, the solve-for vector consisted of the epoch position and velocity vector components. Table 3.21 presents a comparison of the results of two direct integration runs for the Battin test case. In the first case, the observation data of TC-1.1 (position only) were used, while in the second case, TC-1.2 (position and velocity) observational data were used. Examination of the two columns containing

Table 3.21. Direct Integration Method for Battin Test Case With Cartesian Solve-For Vector (Conv. Criterion = 0.001)

Solve-For Vector Components	True Solve-for Components	Direct Integration Method 60 Pos. OBS (TC-1.1) Estimated Values	Direct Integration Method 60 Pos. OBS (TC-1.1) Est. Error Std. Dev.	Direct Integration Method 30 Pos./30 Vel. OBS (TC-1.2) Estimated Values	Direct Integration Method 30 Pos./30 Vel. OBS (TC-1.2) Est. Error Std. Dev.
r_x (km)	6563.340	6562.992	0.1697267	6562.577	0.2417793
r_y (km)	0.0	0.2281757	0.5636508	0.2171448	0.7383149
r_z (km)	0.0	0.2906238	0.3143994	0.4798692	0.4381003
v_x (km/sec)	0.0	-0.0002726	0.0005929	-0.0001263	0.0007843
v_y (km/sec)	7.7930351	7.7934452	0.0001887	7.7939171	0.0002687
v_z (km/sec)	0.0	-0.0002084	0.0003792	-0.0001640	0.0005366
INITIAL COST		1.660×10^5		8.175×10^4	
CONVERGED COST		4.890×10^1		4.259×10^1	
INITIAL GRADIENT MAG.		1.068×10^4		5.190×10^3	
CONVERGED GRADIENT MAG.		3.101×10^{-2}		7.078×10^{-3}	
NO. OF ITERATIONS		5		5	
EXECUTION TIME (MINS.)		19.305		17.733	
RMS POSITION ERROR		667.350 m		891.907 m	
RMS VELOCITY ERROR		0.729 m/sec		0.988 m/sec	

the estimation error standard deviations indicates that, for the given observational data, the position-only observations provided a better estimate. The tabular values of the rms position and velocity errors verify this. It is felt that this is due to the particular choice of the velocity observation error standard deviation used in the observation simulation of TC-1.2 and is not a characteristic of the PVM estimation method. That is, for the test data of TC-1.2, the velocity measurements represent coarser observational data than do the position measurements. Thus, replacement of thirty position measurements by these coarser velocity measurements tends to degrade the estimation results, as indicated in Table 3.21.

The results summarized in Table 3.21 show that after five iterations the magnitude of the vector difference between the estimated epoch position vector of TC-1.1 and the true epoch position vector of Table 3.16 is approximately

$$|\vec{r}_o(\text{estimate}) - \vec{r}_o(\text{true})|_{PVM} \simeq 508 \text{ m}$$

Similarly, the magnitude of the vector difference between the estimated and true epoch velocity vectors after five iterations is given by

$$|\dot{\vec{r}}_o(\text{estimate}) - \dot{\vec{r}}_o(\text{true})|_{PVM} \simeq 0.5 \text{ m/sec}$$

These values can be compared against the results for the Epoch State Navigation Filter for the nearly identical test case previously discussed. Figures 4.26 and 4.27 of Reference 70 depict the magnitudes of the differences in the epoch position and velocity vectors from their true inertial values for one set of initial conditions as a function of the number of measurements. For that particular set of initial conditions, which unfortunately are not given directly in the reference, the magnitude of the epoch position differences after 60 measurements is

$$|\vec{r}_o(\text{estimate}) - \vec{r}_o(\text{true})|_{ESF} \simeq 150 \text{ m}$$

and the magnitude of the epoch velocity difference after 60 measurements is

$$|\dot{\vec{r}}_o(\text{estimate}) - \dot{\vec{r}}_o(\text{true})|_{ESF} \simeq 0.5 \text{ m/sec}$$

Figures 4.46 and 4.47 of the above reference provide similar data for a different set of true initial conditions. For this test case, the magnitude of the epoch position difference after 60 measurements is

$$|\vec{r}_o(\text{estimate}) - \vec{r}_o(\text{true})|_{ESF} \simeq 375 \text{ m}$$

and the magnitude of the epoch velocity difference after 60 measurements is

$$|\dot{\vec{r}}_o(\text{estimate}) - \dot{\vec{r}}_o(\text{true})|_{ESF} \simeq 0.3 \text{ m/sec}$$

It is not possible to directly compare the PVM results to the ESF results of Reference 70 because the truth ephemeris epoch conditions are not the same. However, the above data indicates that the PVM method produces results of very comparable accuracy to the results produced by the Epoch State Navigation Filter.

The character of the gradient-dependent PVM algorithm for this test case can be seen from Figures 3.15 through 3.22. Figure 3.15 depicts the differences between the components of the epoch position vector calculated by the PVM estimation method and the components of the true epoch position vector as a function of PVM iteration number. The results depicted here are based on the direct integration method using the TC-1.1 simulated observations. Figure 3.16 presents similar data for the components of the epoch velocity vector. These figures show that early iterations of the method may induce some overshoot or undershoot in

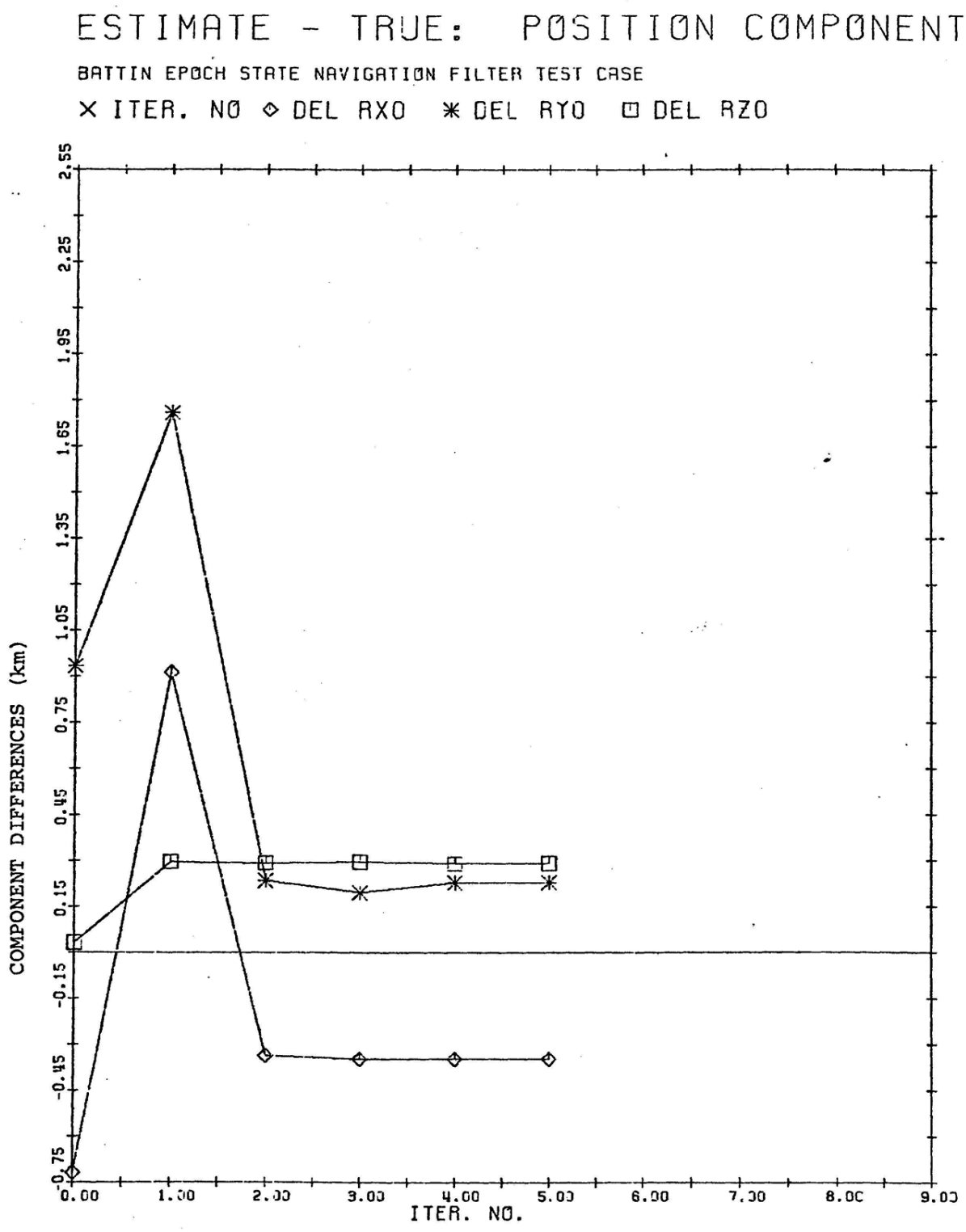


Figure 3.15. Component Differences Between Estimated and True Epoch Position Vectors for Battin Test Case

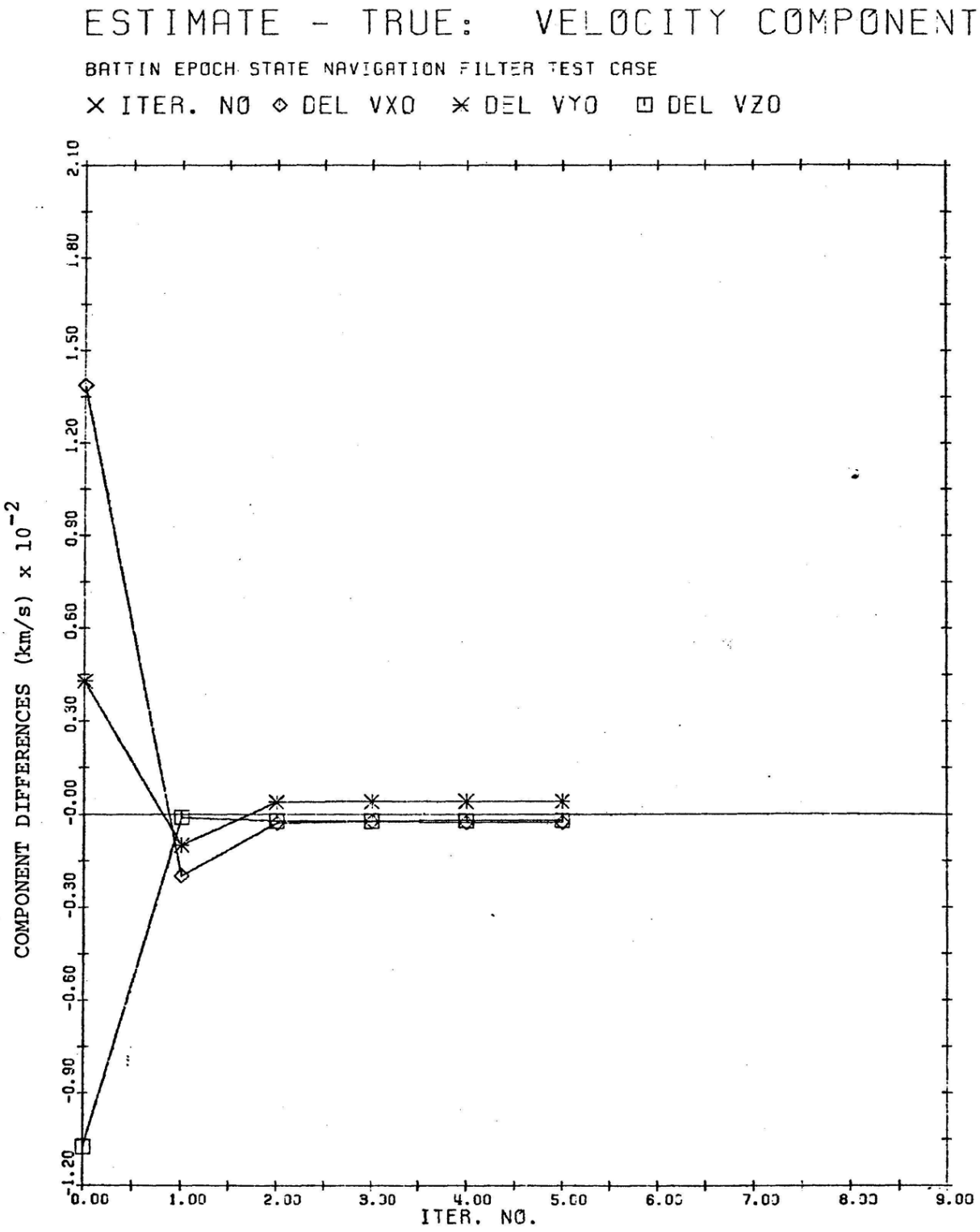


Figure 3.16. Component Differences Between Estimated and True Velocity Vectors for Battin Test Case

the estimated quantities. However, these transients die out within the first few iterations and the estimated quantities rapidly converge to final values consistent with the quality of the observation data. While Figure 3.15 presents the difference between the estimated epoch position vector and the true epoch position vector on a component basis, Figure 3.17 depicts the magnitude of the vector difference between the estimated and true epoch position vectors as a function of iteration number. This plot shows that the position vector difference displays a similar type of early transient nature as the individual components. Again the algorithm rapidly damps out the transient in the estimated quantities. Figure 3.18 shows a similar, though more uniform, convergence plot for the magnitude of the vector difference between the estimated and true epoch velocity vectors.

When the current "best estimate" of the solve-for vector in a gradient-dependent search algorithm is far from the true solution, the gradient components are usually large. The resulting search direction, given in general by Equation (3.1), rapidly moves the algorithm in the direction of the minimum. However, as the current "best estimate" approaches the minimizing value of the solve-for vector, the gradient components approach zero, the gradient

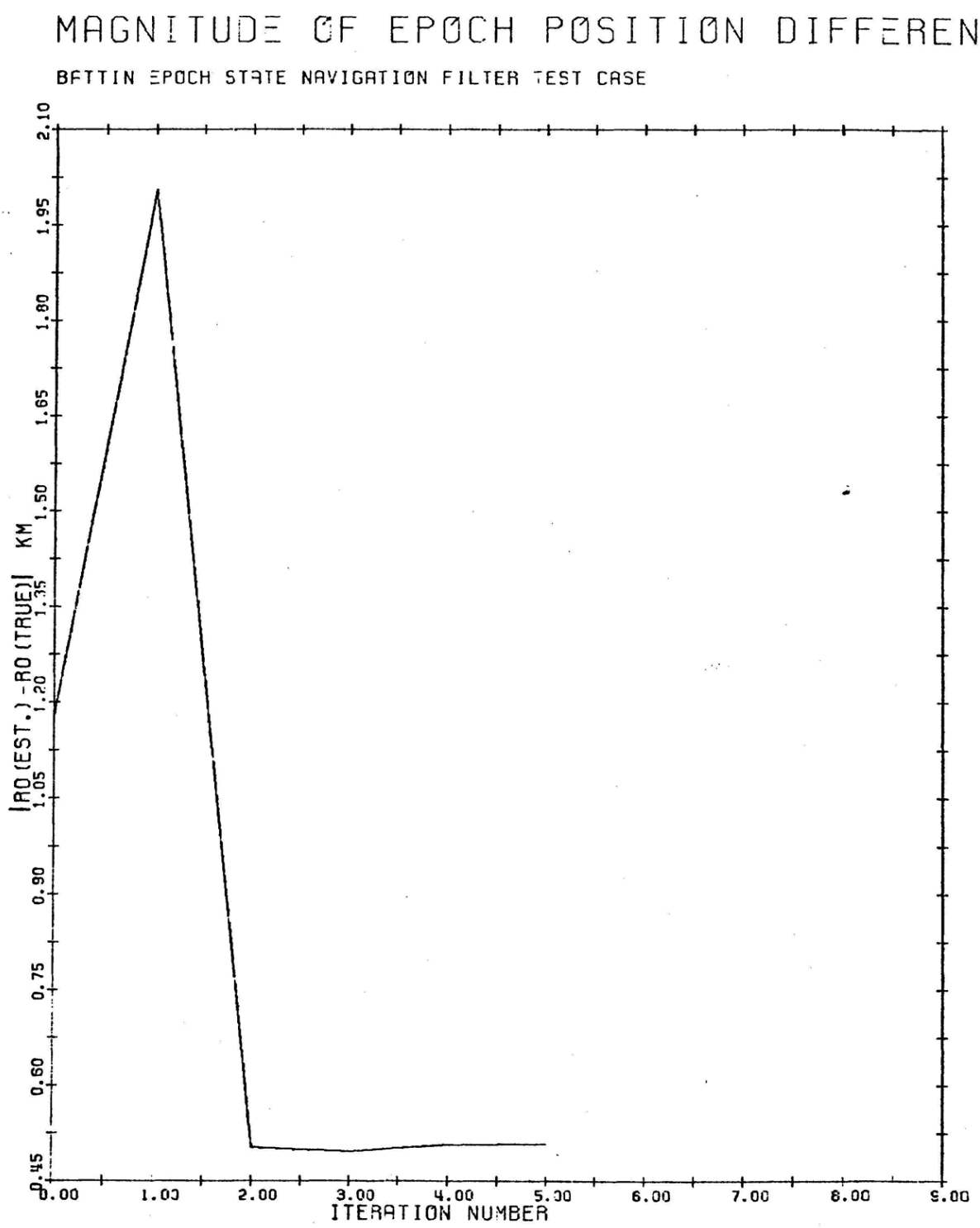


Figure 3.17. Magnitude of Vector Difference Between Estimated and True Epoch Position Vectors for Battin Test Case

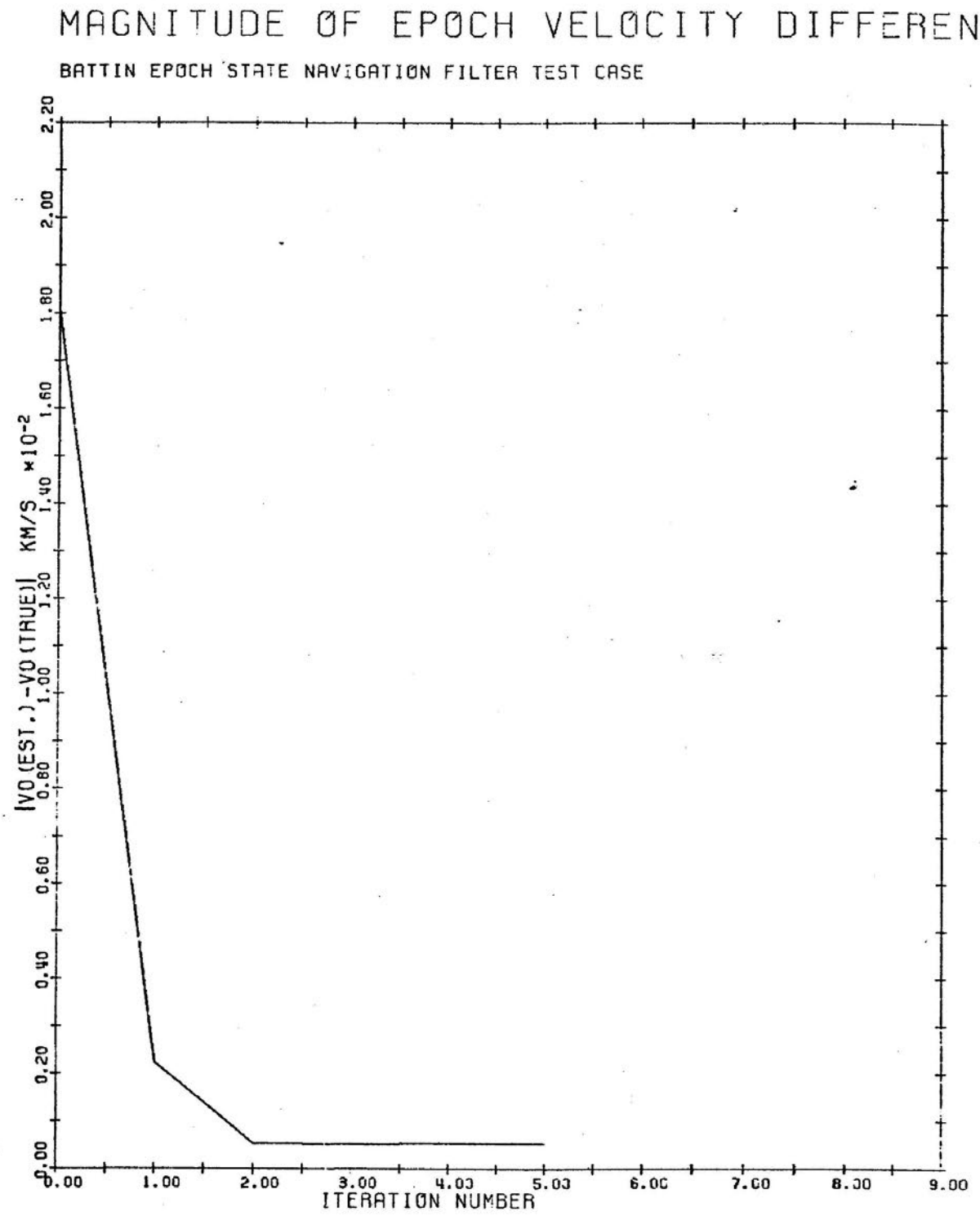


Figure 3.18. Magnitude of Vector Difference Between Estimated and True Epoch Velocity Vectors for Battin Test Case

loses much of its directional information and the search direction vector components become small. Thus, the influence of the search direction vector on the current iterate of the solve-for vector becomes less and the algorithm rate of convergence becomes slower.

The PVM algorithm displays just these classic gradient-dependent search characteristics. Figure 3.19 shows the behavior of the weighted least squares cost functional as a function of iteration number. Table 3.21 shows that the initial value of the cost functional is about 2×10^5. However, after a single PVM iteration, the cost functional has been reduced to about 1×10^2. Thus, the first iteration of the PVM method reduces the cost functional by three orders of magnitude. Because the initial value of the cost functional is so large compared to the subsequent values generated by the PVM orbit estimation algorithm, the scale of Figure 3.19 obscures the behavior of the cost functional beyond iteration number one. Therefore, Figure 3.20 is included to show the cost functional values in iterations two through five. The slow convergence rate in the later iterations is clearly shown, since iterations three, four and five only reduce the cost functional from 48.914306 to 48.904281. The magnitude of the gradient

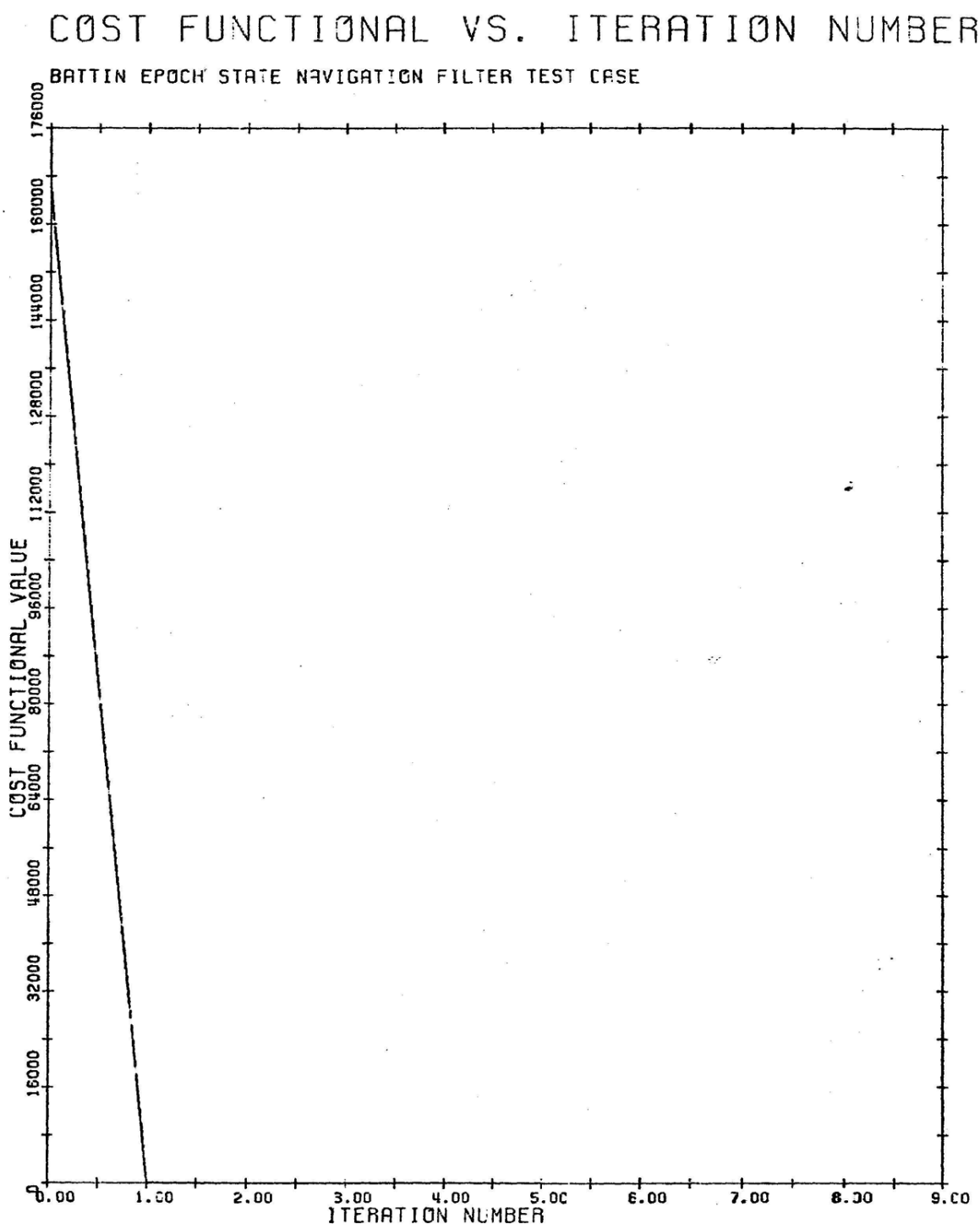


Figure 3.19. Weighted Least Squares Cost Functional Value vs. Iteration Number for Battin Test Case

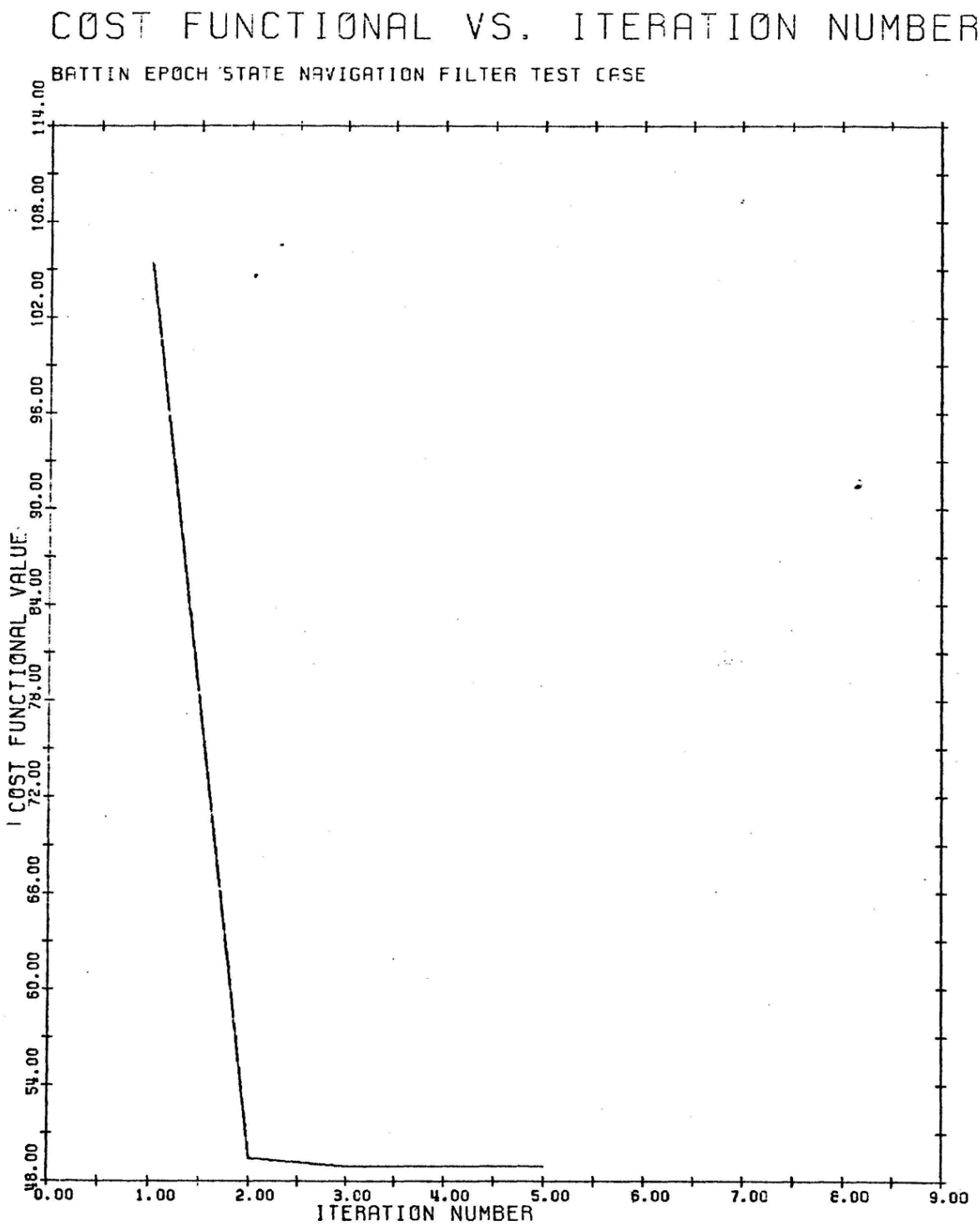


Figure 3.20. Cost Functional vs. Iteration Number for Battin Test Case: Iterations Two Through Five

vector behaves in an analogous manner, as shown in Figures 3.21 and 3.22.

The previous results were for the direct integration approach to the evaluation of the cost functional and its associated gradient vector. As previously stated, this method is the most computationally costly approach to the problem of cost and gradient evaluation. In formulating the linearization method described in Section 3.3.4, it was conjectured that computational advantage could be obtained due to the fact that many of the numerical integrations required in the direct integration method could be replaced by vector-matrix multiplications based on the linearization about the reference trajectory based on the current "best estimate" of the solve-for vector. The PVM estimation results summarized in Table 3.22 dramatically verify these conjectures. Two important points stand out when Tables 3.22 and 3.21 are compared.

First, in comparing the TC-1.1 estimation results for the direct integration method of Table 3.21 against the linearization results of Table 3.22, virtually no difference is seen in the converged solve-for parameters. The same holds true for the TC-1.2 estimation results displayed

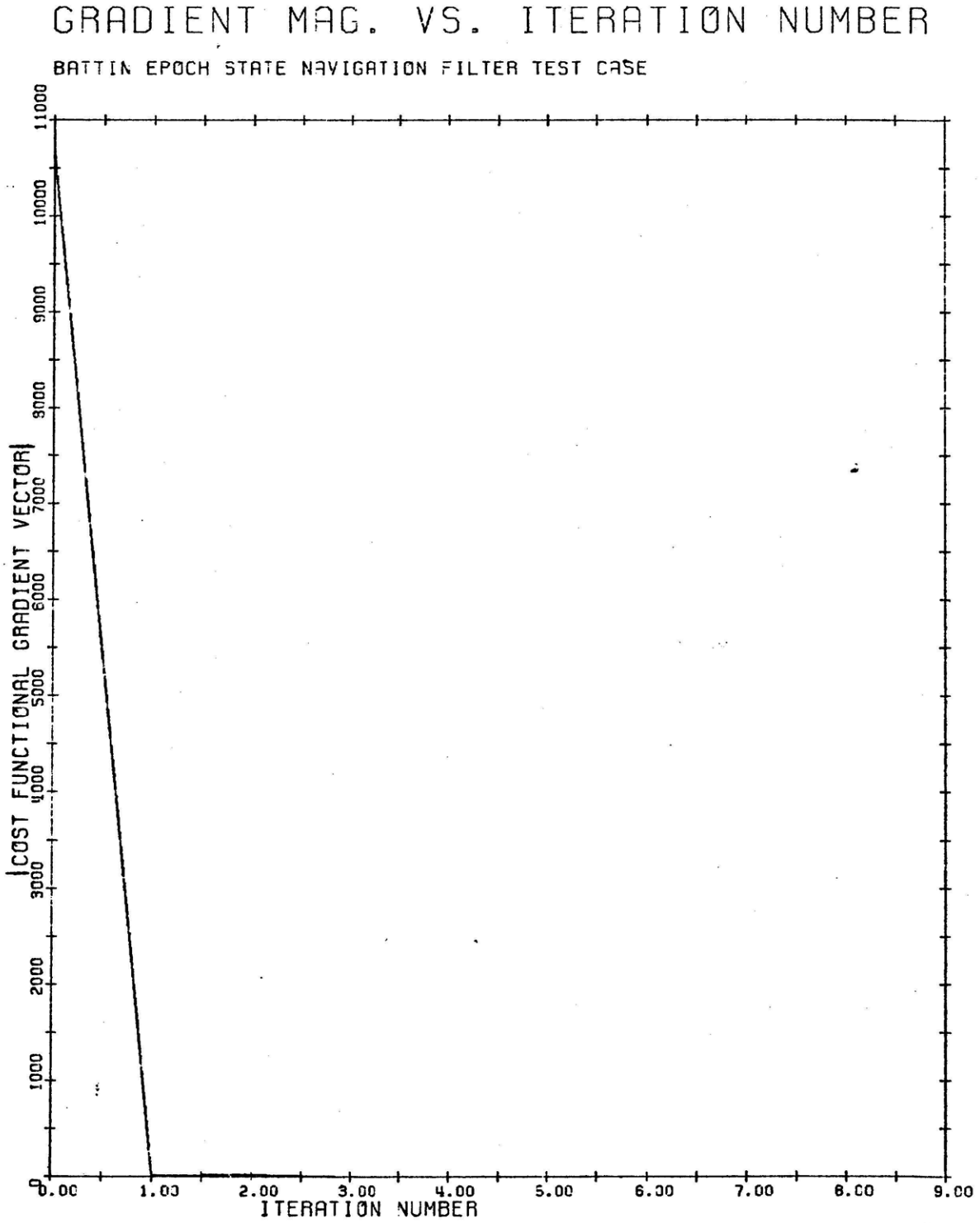


Figure 3.21. Magnitude of Gradient Vector vs. Iteration Number for Battin Test Case

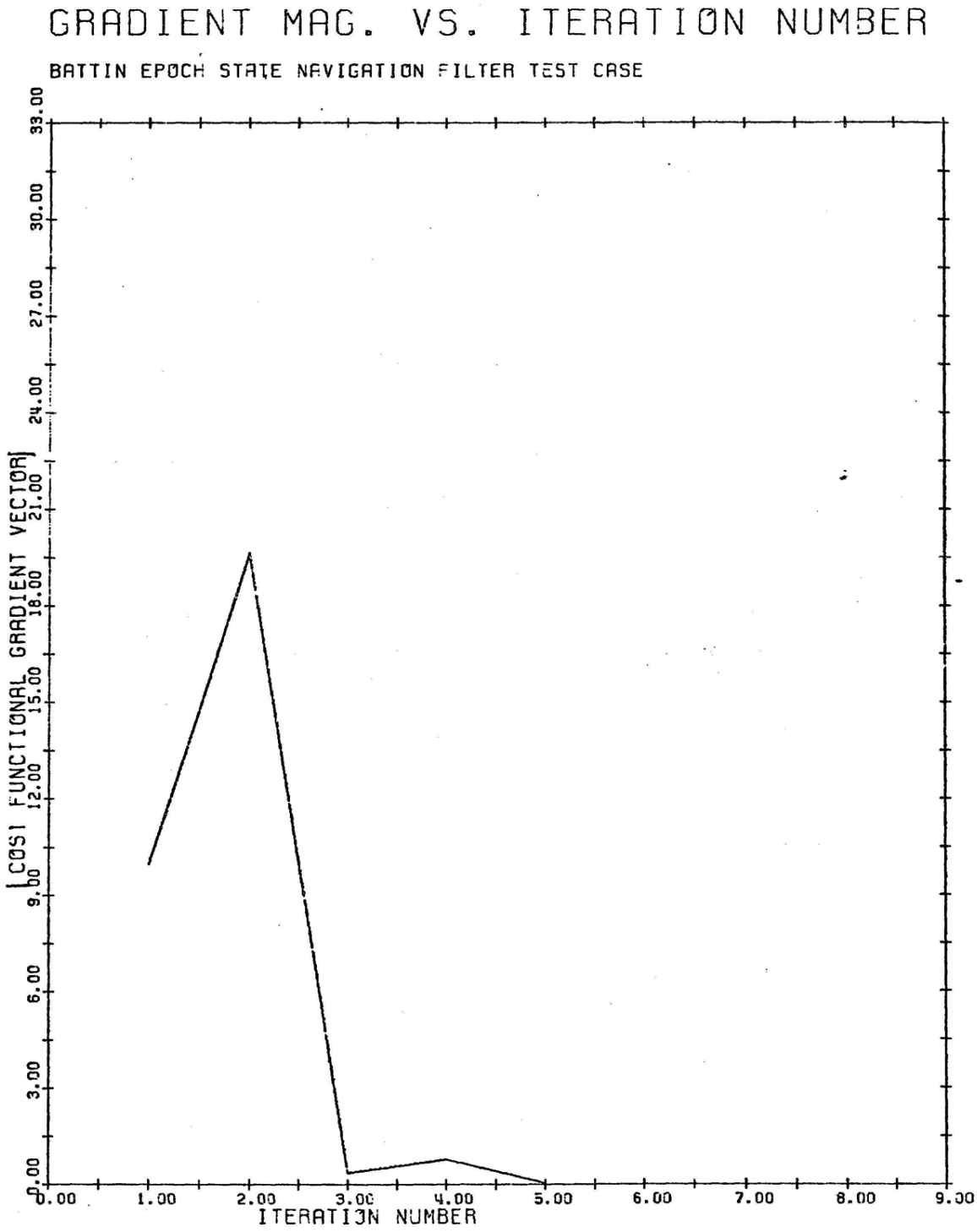


Figure 3.22. Magnitude of Gradient Vector vs. Iteration Number for Battin Test Case: Iterations Two Through Five

Table 3.22. Linearization Method for Battin Test Case with Cartesian Solve-For Vector (Conv. Criterion = 0.001)

Solve-for Vector Components	True Solve-for Components	Linearization Method 60 Pos. OBS (TC-1.1) Estimated Values	Linearization Method 60 Pos. OBS (TC-1.1) Est. Error Std. Dev.	Linearization Method 30 Pos./30 Vel. OBS (TC-1.2) Estimated Values	Linearization Method 30 Pos./30 Vel. OBS (TC-1.2) Est. Error St. Dev.
r_x (km)	6563.340	6562.992	0.1697267	6562.576	0.2417793
r_y (km)	0.0	0.2286400	0.5636508	0.2167734	0.7383149
r_z (km)	0.0	0.2906664	0.3143994	0.4799242	0.4381003
v_x (km/sec)	0.0	-0.0002726	0.0005929	-0.0001261	0.0007843
v_y (km/sec)	7.7930351	7.7934454	0.0001887	7.7939181	0.0002687
v_z (km/sec)	0.0	-0.0002084	0.0003792	-0.0001652	0.0005366
INITIAL COST		1.660×10^5		8.175×10^4	
CONVERGED COST		4.890×10^1		4.259×10^1	
INITIAL GRADIENT MAG.		1.068×10^4		5.190×10^3	
CONVERGED GRADIENT MAG.		3.063×10^{-5}		1.134×10^{-5}	
NO. OF ITERATIONS		4		4	
EXECUTION TIME (MINS.)		2.550		2.504	
RMS POSITION ERROR		667.350 m		891.907 m	
RMS VELOCITY ERROR		0.729 m/sec		0.988 m/sec	

on these tables. The standard deviations of the estimation errors for both TC-1.1 and TC-1.2 are essentially identical for both methods of cost and gradient evaluation, as are the rms position and velocity errors. It can be concluded, therefore, that little or no accuracy is lost in the estimation process by using the linearization technique for evaluation of the cost functional and its associated gradient vector.

The second important point of comparison between Tables 3.21 and 3.22 verifies the conjecture about the potential performance improvement obtained by including the linearization method in the Parallel Variable Metric orbit estimation algorithm. Table 3.21 shows that the direct integration method for the observational data of TC-1.1 and TC-1.2 took 19.305 and 17.733 minutes respectively to execute on the Amdahl 470 V6 computer at the Charles Stark Draper Laboratory. This execution is a sequential simulation of the parallel orbit estimation algorithm and as such the computations are performed in a serial manner. Examination of the execution times given in Table 3.22 for these test cases using the linearization method shows 2.550 and 2.504 minutes respectively. Hence, for these test cases the use of the linearization method in place of the direct integration method results in an execution

speed-up ratio between 7.0 and 7.5. When it is recalled that there is virtually no difference in the accuracy of the estimation results provided by the two methods, the linearization method seems to provide an efficient yet accurate method of cost and gradient evaluation and significantly speeds the PVM estimation process.

In addition to the runs using Cartesian solve-for vectors, another set of test runs using Keplerian and equinoctial solve-for vectors were made. In Table 3.23 the solve-for vector components were the epoch Keplerian elements. Data is provided for both the 60 position measurements of TC-1.1 or the 30 position and 30 velocity measurements of TC-1.2. In both these cases, the poorly defined physical nature of the argument of periapsis (ω), longitude of the ascending node (Ω) and mean anomaly (M) for this near-circular, near-equatorial test case is readily apparent. The estimated values of these three parameters differ markedly from their true values of zero degrees. In addition, the standard deviations of the estimation errors in these quantities reach as high as 108°. This is reflective of the low confidence in the estimate of these three solve-for parameters because of their physical nature in this particular test case.

Table 3.23. Linearization Method for Battin Test Case With Keplerian Solve-For Vector (Conv. Criterion = 0.001)

Solve-for Vector Components	True Solve-for Components	Linearization Method 60 Pos. OBS (TC-1.1) Estimated Values	Linearization Method 60 Pos. OBS (TC-1.1) Est. Error Std. Dev.	Linearization Method 30 Pos./30 Vel. OBS (TC-1.2) Estimated Values	Linearization Method 30 Pos./30 Vel. OBS (TC-1.2) Est. Error Std. Dev.
a (km)	6563.340	6563.335	0.0419613	6563.300	0.0588918
e	0.0	0.0000522	0.0000230	0.0001115	0.0000326
i (deg)	0.0	0.0029376	0.0027823	0.0021011	0.0038457
ω (deg)	0.0	121.677	58.587	278.513	108.084
Ω (deg)	0.0	283.609	53.616	72.931	106.971
M (deg)	0.0	359.717	23.619	8.556	15.471
INITIAL COST		1.660×10^{5}		8.175×10^{4}	
CONVERGED COST		4.890×10^{1}		4.542×10^{1}	
INITIAL GRADIENT MAG.		3.990×10^{3}		1.937×10^{3}	
CONVERGED GRADIENT MAG.		2.160×10^{-2}		7.959×10^{-4}	
NO. OF ITERATIONS		9		8	
EXECUTION TIME (MINS)		5.137		4.597	

Despite the difficulty in estimating these elements, it should be noted that the converged values of the cost functional are very near those cost functional values found using the well-defined Cartesian solve-for vector, as tabulated in Tables 3.21 and 3.22. It is felt, however, that the ill-defined character of these solve-for parameters caused a deleterious effect on the algorithm performance. Because the converged values of ω, Ω, and M were relatively far from their *a priori* values, additional PVM iterations were required to reach convergence. Thus, the test case TC-1.1 took 9 iterations and 5.137 minutes to converge, while TC-1.2 took 8 iterations and 4.597 minutes. Therefore, the linearized method using a Keplerian solve-for vector is only about four times faster than the direct integration method with a Cartesian solve-for vector and is only half as fast as the linearization method with a Cartesian solve-for vector.

In contrast to this, Table 3.24 summarizes the Battin test case results for the PVM orbit estimation algorithm using the linearization method and a solve-for vector of epoch equinoctial elements. The parameters in this solve-for vector are well-defined, even for the vanishing eccentricity and vanishing inclination of this test case. Because of this, the PVM algorithm is more well behaved and

Table 3.24. Linearization Method for Battin Test Case With Equinoctial Solve-For Vector (Conv. Criterion = 0.001)

		Linearization Method 60 Pos. OBS (TC-1.1)		Linearization Method 30 Pos./30 Vel. OBS (TC-1.2)	
Solve-for Vector Components	True Solve-for Components	Estimated Values	Est. Error Std. Dev.	Estimated Values	Est. Error Std. Dev.
a (km)	6563.340	6563.333	0.0419613	6563.300	0.0588918
h	0.0	1.49665×10^{-7}	0.0000215	-1.68485×10^{-5}	0.0000301
k	0.0	5.22487×10^{-5}	0.0000230	1.10222×10^{-4}	0.0000327
p	0.0	-2.21448×10^{-5}	0.0000240	-3.65656×10^{-5}	0.0000334
q	0.0	-1.33681×10^{-5}	0.0000243	-1.05955×10^{-5}	0.0000344
λ (deg)	0.0	0.0020129	0.0040966	-0.0000385	0.0055516
INITIAL COST		1.660×10^{5}		8.175×10^{4}	
CONVERGED COST		4.890×10^{1}		4.259×10^{1}	
INITIAL GRADIENT MAG.		4.509×10^{2}		2.197×10^{2}	
CONVERGED GRADIENT MAG.		5.168×10^{-5}		2.053×10^{-5}	
NO. OF ITERATIONS		3		3	
EXECUTION TIME (MINS.)		1.945		1.903	

the algorithm converges very rapidly. For the TC-1.1 observational data, the method converges in three iterations and 1.945 minutes and for the TC-1.2 observational data, the method converges in three iterations and 1.903 minutes. The speed-up ratio between the direct integration method using a Cartesian solve-for vector and the linearization method using an equinoctial solve-for vector is slightly less than 10. Additionally, the equinoctial solve-for vector formulation provides the most rapid convergence of the possible solve-for vectors implemented in this investigation when used in conjunction with the linearized method of cost and gradient evaluation. This point appears to hold consistently throughout all the test cases run in this effort.

As a final test of the accuracies of the various PVM orbit estimation formulations discussed above, trajectories were generated based on the PVM estimation results of Tables 3.21, 3.22, 3.23 and 3.24. These trajectories were then compared with the truth ephemeris generated from the given epoch conditions of Table 3.16. Comparisons were performed over the initial 3 hour observation span and over a 24 hour period, giving a 21 hour prediction span beyond the 3 hour observation span.

The results of these comparisons are presented in Table 3.25 for the observation span and Table 3.26 for the predict span. The major result indicated in these tables is that there is virtually no difference in prediction performance between the four options presented in these tables for either data case TC-1.1 or TC-1.2. For example, for the case of 60 position observations (TC-1.1), the direct integration method with a Cartesian solve-for vector and the linearization method using the three possible solve-for vectors all yielded maximum position differences that varied by a maximum of only 3 meters over the observation span and 11 meters over the predict span. Similar results are shown for the case of 30 position and 30 velocity measurements (TC-1.2). This implies that the accuracy of the estimation results is unaffected by the choices of solve-for vector and of method for evaluation of the cost functional and its gradient. However, it must be remembered that significant differences in execution performance exist between these different formulations, with the linearization method using an equinoctial solve-for vector exhibiting the best performance. Since no degradation in accuracy is found, this formulation is certainly the method of choice for the orbit estimation problem.

Table 3.25. Comparison of Estimated Trajectory vs. Truth Ephemeris During 3 Hour Observation Span for Battin Test Cases

Test Case Description	Maximum Sampled* Position Error (km)			
	Radial	Cross Track	Along Track	Delta-R
Direct Int. Method (Cartesian Solve-For)				
TC-1.1	0.348	0.339	0.990	1.010
TC-1.2	0.772	0.499	1.949	1.951
Linearization Method (Cartesian Solve-For)				
TC-1.1	0.348	0.339	0.990	1.010
TC-1.2	0.772	0.500	1.951	1.953
Linearization Method (Keplerian Solve-For)				
TC-1.1	0.348	0.336	0.989	1.010
TC-1.2	0.772	0.241	1.951	1.952
Linearization Method (Equinoctial Solve-For)				
TC-1.1	0.348	0.339	0.990	1.010
TC-1.2	0.772	0.500	1.951	1.953
A Priori Trajectory (Uncorrected)	20.878	9.029	140.889	141.078

*Sampling rate was every 2 minutes.

Table 3.26. Comparison of Estimated Trajectory vs. Truth Ephemeris During 21 Hour Predict Span for Battin Test Cases

Test Case Description	Maximum Sampled* Position Error (km)			
	Radial	Cross Track	Along Track	Delta-R
Direct Int. Method (Cartesian Solve-For)				
TC-1.1	0.348	0.339	1.819	1.830
TC-1.2	0.769	0.499	7.921	7.925
Linearization Method (Cartesian Solve-For)				
TC-1.1	0.341	0.339	1.827	1.838
TC-1.2	0.770	0.500	7.928	7.932
Linearization Method (Keplerian Solve-For)				
TC-1.1	0.348	0.336	1.829	1.841
TC-1.2	0.776	0.241	7.930	7.934
Linearization Method (Equinoctial Solve-For)				
TC-1.1	0.348	0.339	1.827	1.838
TC-1.2	0.770	0.500	7.928	7.932
A Priori Trajectory (Uncorrected)	86.228	9.031	920.454	924.499

*Sampling Rate was every 15 minutes.

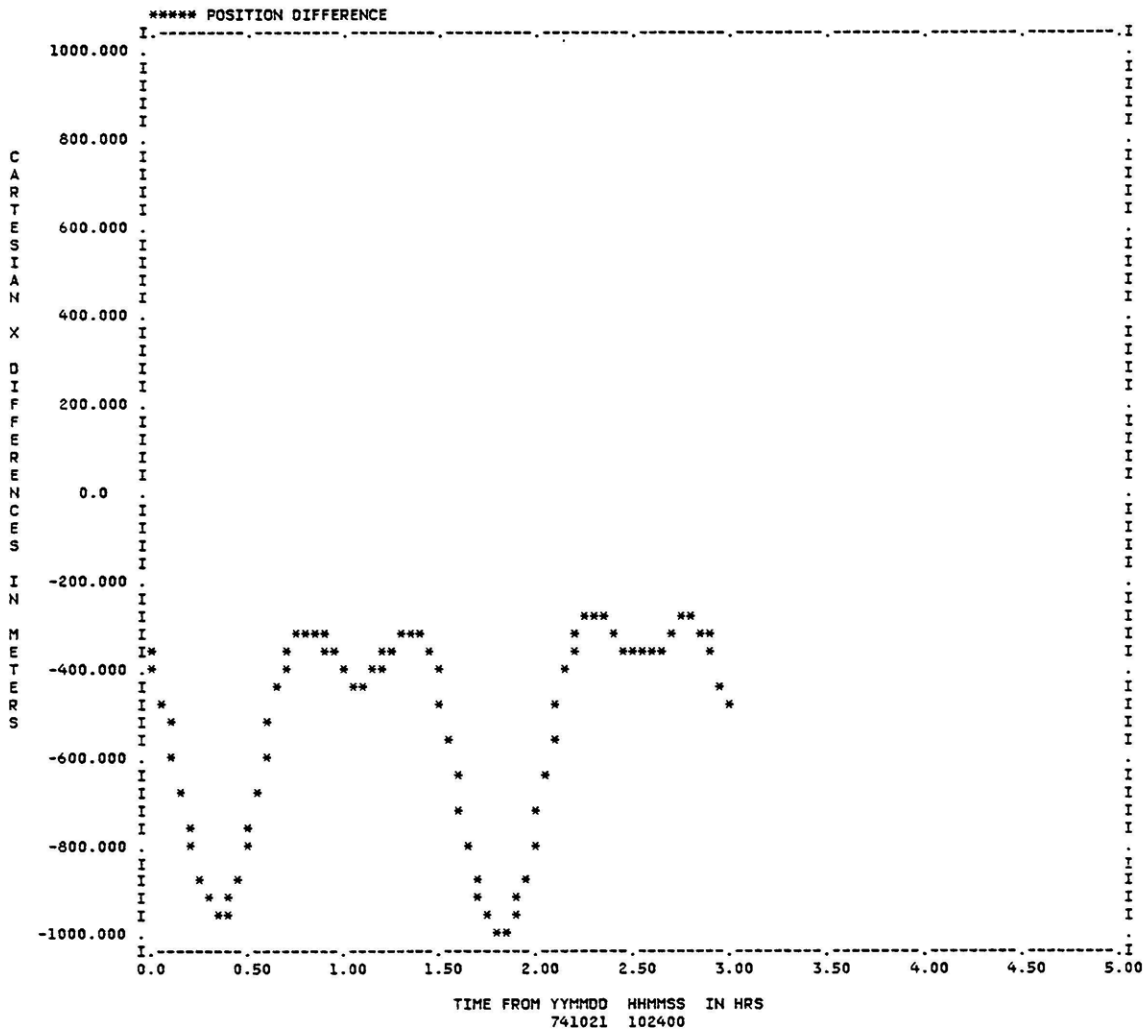
It is also interesting to note that, despite the poorly defined physical nature of the Keplerian elements ω, Ω and M and despite the resulting poor estimate of these quantities as shown in Table 3.23, the prediction results over 24 hours are exactly comparable to those generated from the well defined Cartesian or equinoctial solve-for sets. This seems to imply that although these quantities may be ill-defined physically, their mathematical contribution to the estimation problem is not significant for this particular test case. Hence, the Keplerian elements will provide a good set of solve-for elements, even for the near-circular and near-equatorial cases such as this. However, a caution must be expressed here when using the Keplerian solve-for vector with the PVM estimation algorithm. If the magnitude of the parameter space basis vector corresponding to the eccentricity solve-for parameter is not chosen to be sufficiently small, then during the search algorithm it is possible to get a negative eccentricity which will cause the integration software to fail and the estimation run to terminate. This can be avoided by proper selection of the scale factors input to the PVM estimation algorithm, but that requires some experience on the part of the user. The equinoctial solve-for vector does not have this problem, since the eccentricity is included in the h and k elements and negative values of these elements will not cause the

algorithm to fail. Therefore, it is again recommended that the equinoctial solve-for vector be used in the PVM orbit estimation method.

Figures 3.23 through 3.28 plot the differences in the Cartesian position and velocity components between the ephemeris based on the PVM estimation algorithm and the truth ephemeris. The comparison is taken over the 3 hour observation span. The initial conditions for the PVM ephemeris are based on the epoch equinoctial solve-for vector, but is has been shown that virtually no difference exists between this solve-for vector and the other possible solve-for vectors in the PVM method. It is seen that the initial values of these plots represent the differences in the estimated epoch position and velocity vectors from the truth ephemeris epoch conditions of Table 3.16. Over the 3 hour observation span the maximum position component deviation from the truth ephemeris is slightly less than 1 km in the x-component of the position. Similarly, the maximum velocity deviation occurs in the v_y component and is approximately 0.8 m/s. The GTDS compare software also computes at the compare sample points the delta-r error, which is defined as

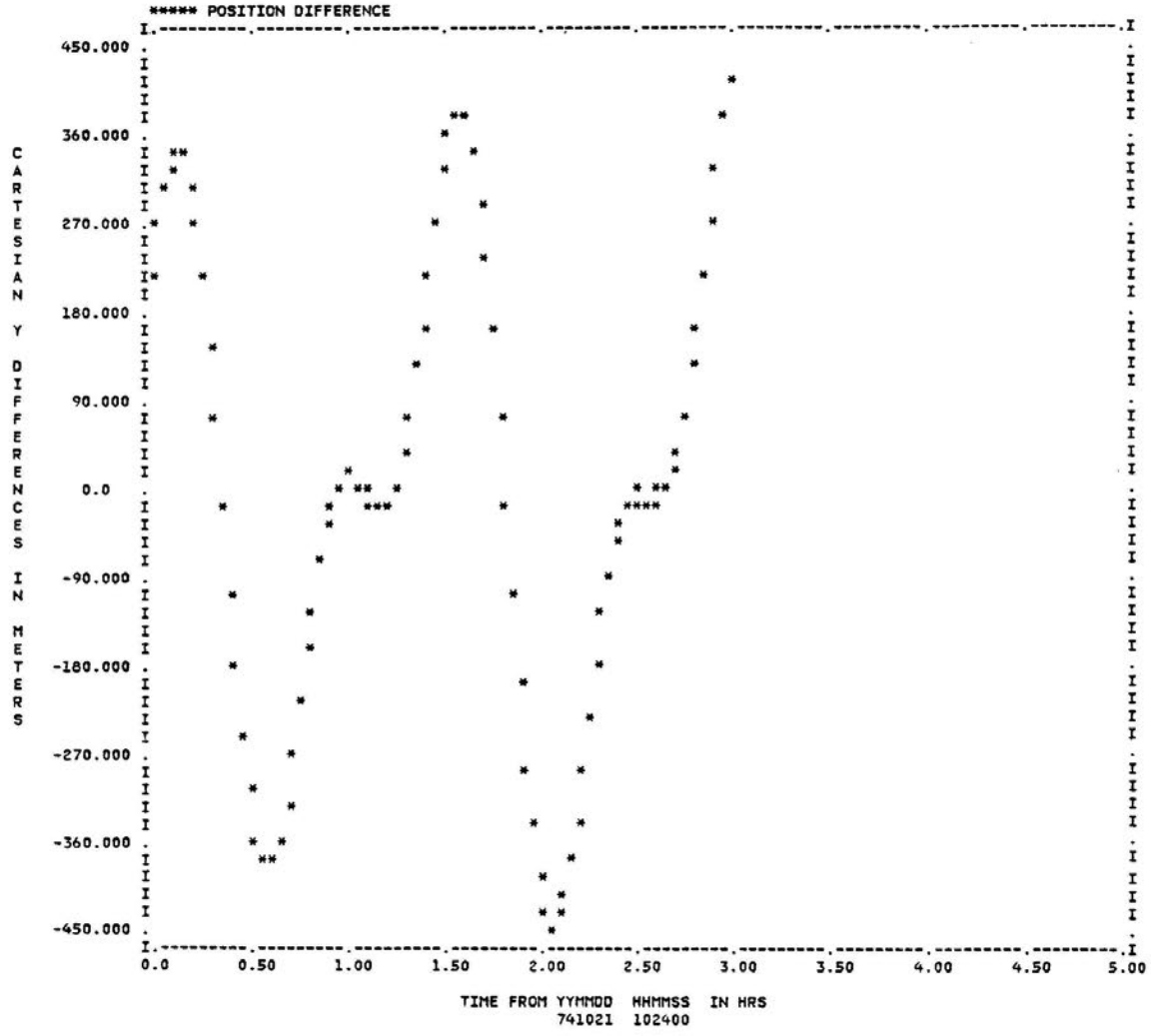
$$|\Delta\vec{r}| = |\vec{r}_{PVM} - \vec{r}_{TE}|$$

Figure 3.23. Difference in "x" Position Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span



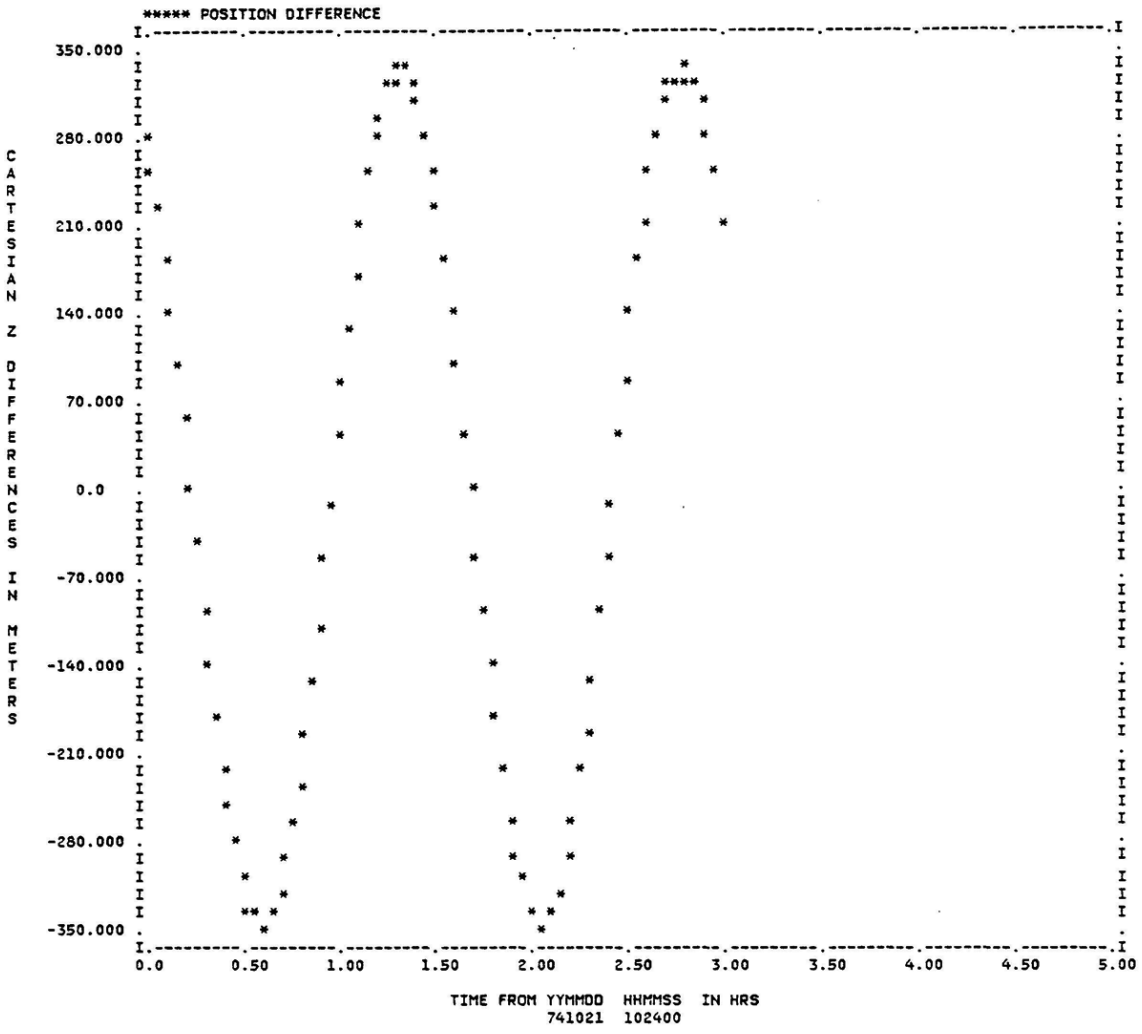
Truth ephemeris based on initial conditions of Table 3.16.
PVM estimation results based on linearized method with equinoctial solve-for vector (see Table 3.26).
Force Model: J_2 zonals, no drag

Figure 3.24. Difference in "y" Position Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span



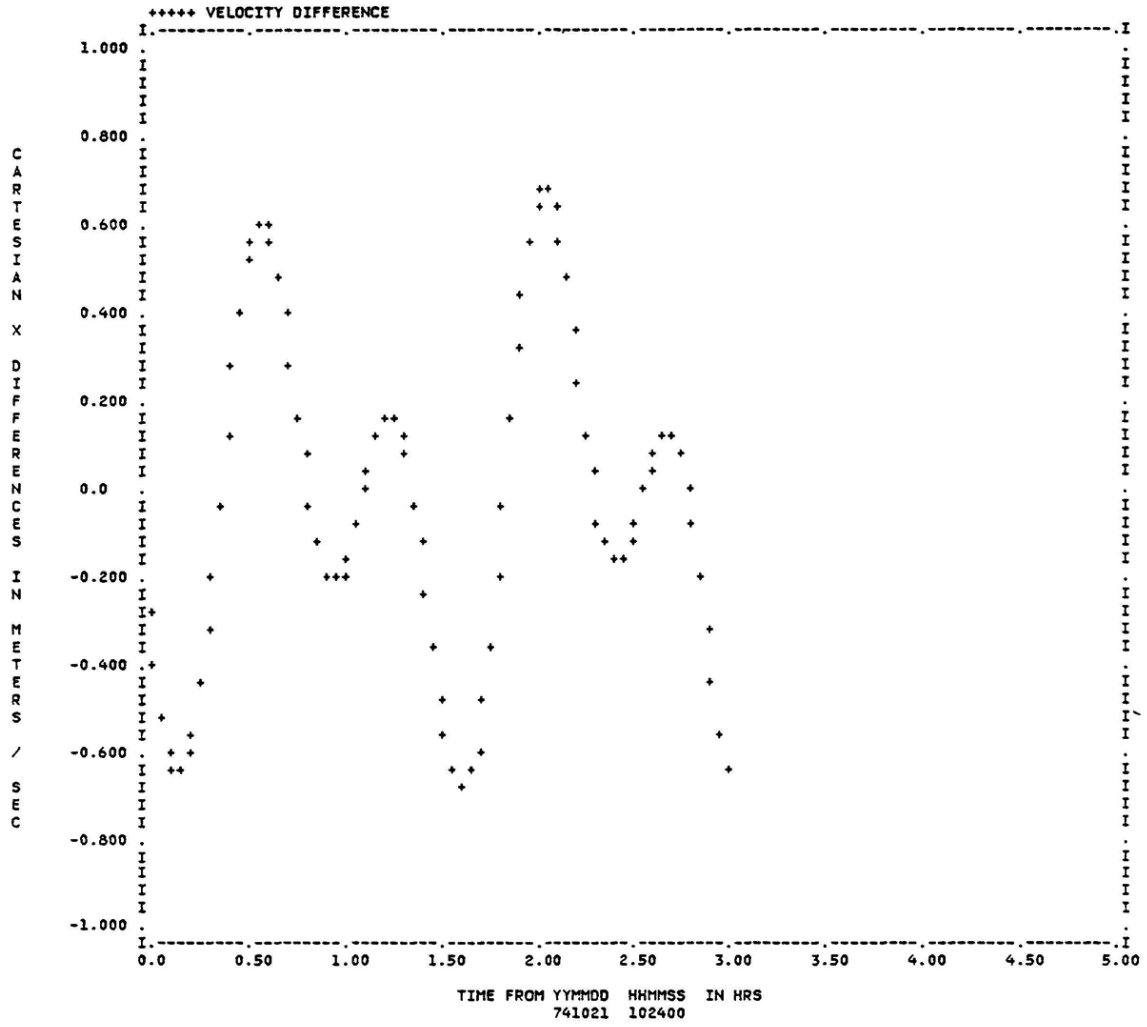
See Figure 3.23 for initial conditions and force model.

Figure 3.25. Difference in "z" Position Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span



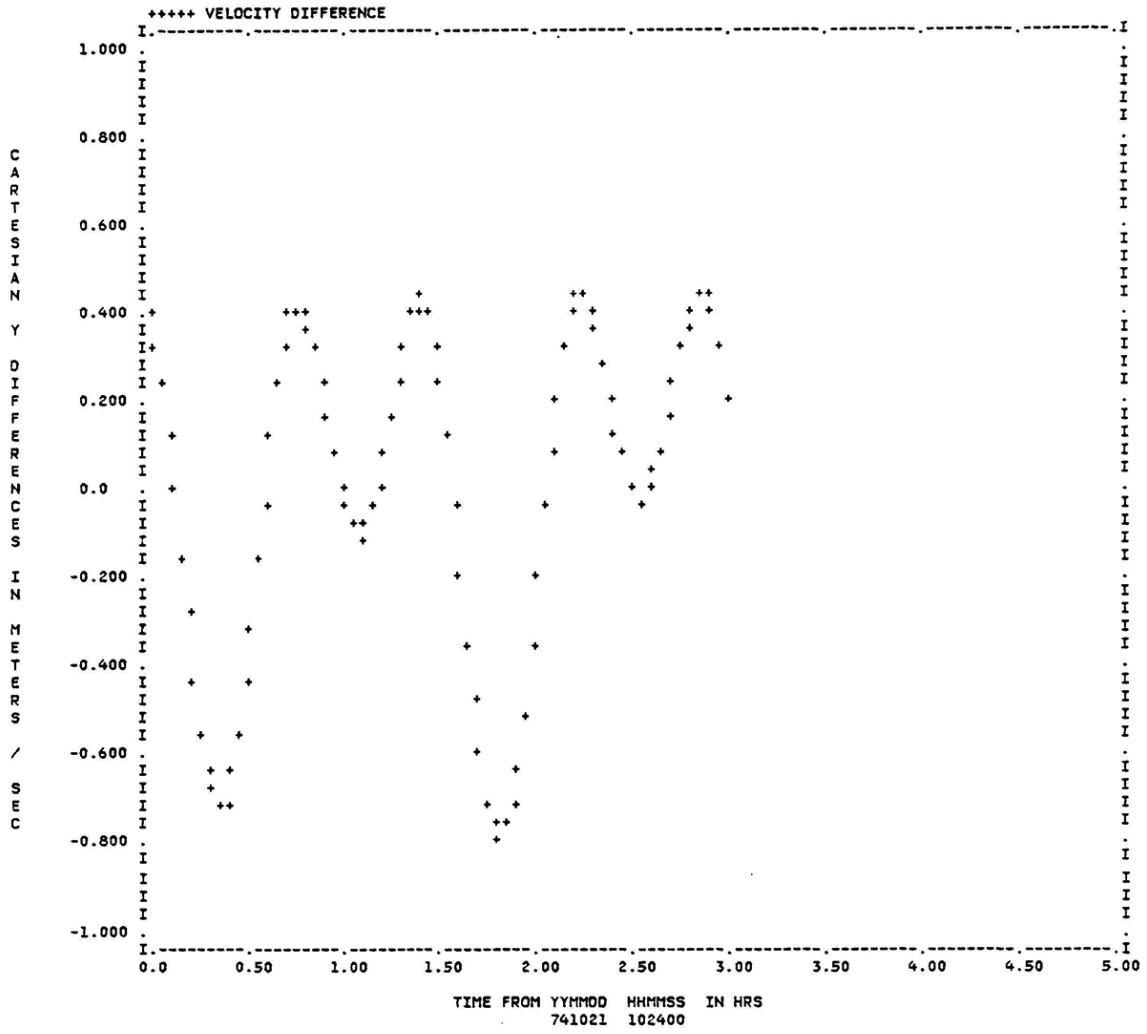
See Figure 3.23 for initial conditions and force model.

Figure 3.26. Difference in "x" Velocity Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span



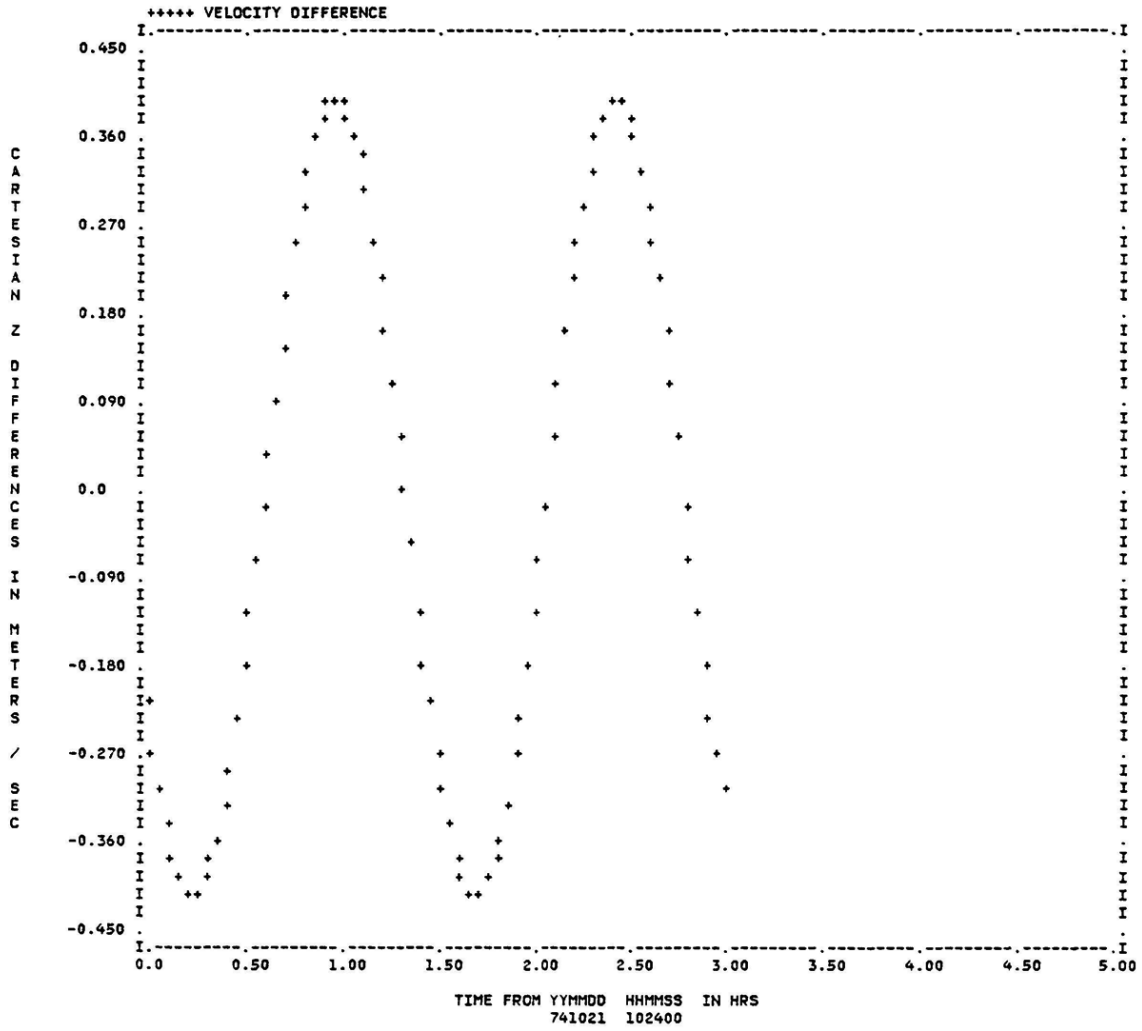
See Figure 3.23 for initial conditions and force model.

Figure 3.27. Difference in "y" Velocity Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span



See Figure 3.23 for initial conditions and force model.

Figure 3.28. Difference in "z" Velocity Component for Battin Test Case: PVM Estimated Trajectory versus Truth Ephemeris During Observation Span



See Figure 3.23 for initial conditions and force model.

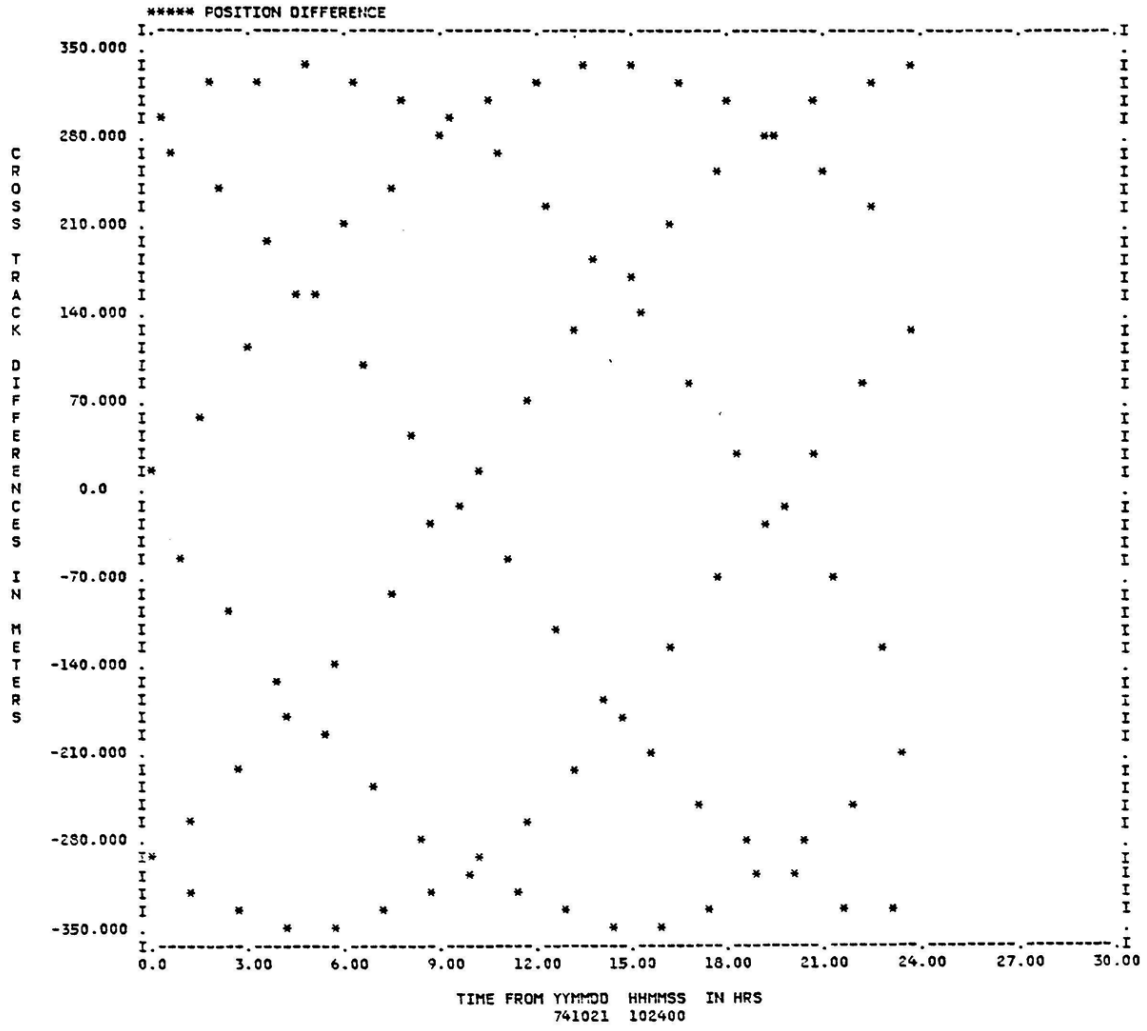
with $\vec{r}_{PVM}$ = ephemeris position vector based on PVM estimation solve-for state

$\vec{r}_{TE}$ = truth ephemeris position vector

For this test case, this magnitude of the vector difference between the estimated and true position vectors achieved a maximum of 1010 meters and a minimum of 291 meters during the observation span. This can be compared to the test case described by Battin, et al. (30), in which the maximum $|\Delta\vec{r}|$ over the 80 measurements was 1800 meters and the minimum was 500 meters, and to another test case in Reference 70, with a maximum of 2700 meters and a minimum of 200 meters. Once again this seems to indicate that the accuracy of the PVM estimation algorithm is comparable to that of the Epoch State Navigation Filter and to the conventional Kalman filter also.

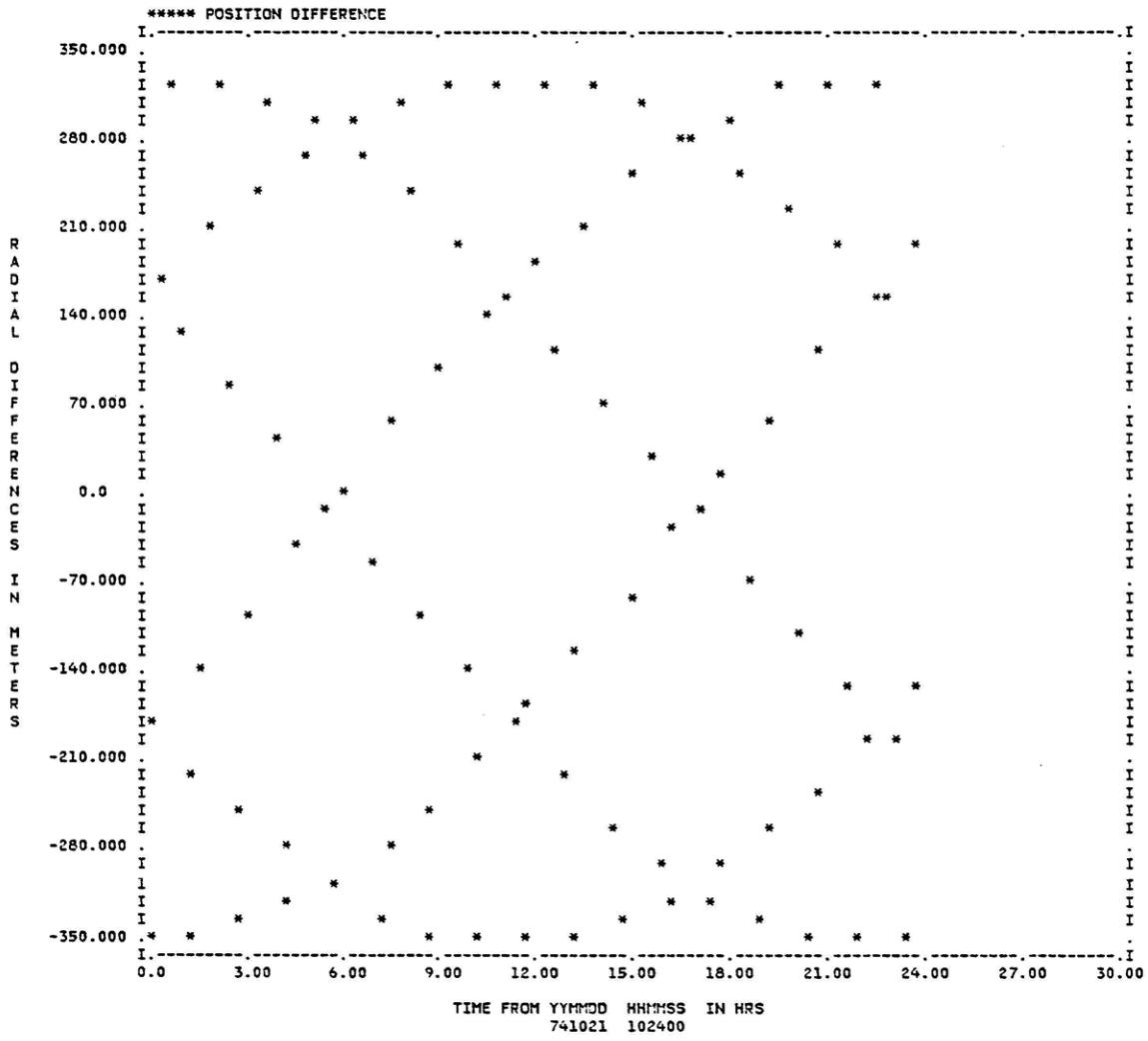
The character of the estimated ephemeris beyond the observation span is also of interest. Figures 3.29, 3.30 and 3.31 present the radial, cross track and along track error histories of the PVM ephemeris compared to the truth ephemeris. The compare span is 1 day and the observation span is the first three hours of this 1 day compare, leaving a 21 hour predict span. For this relatively simple test case, it is possible to identify most of the estima-

Figure 3.29. Radial Difference Over 24 Hour Compare Span (21 Hour Predict) for Battin Test Case: PVM Estimated Ephemeris versus Truth Ephemeris



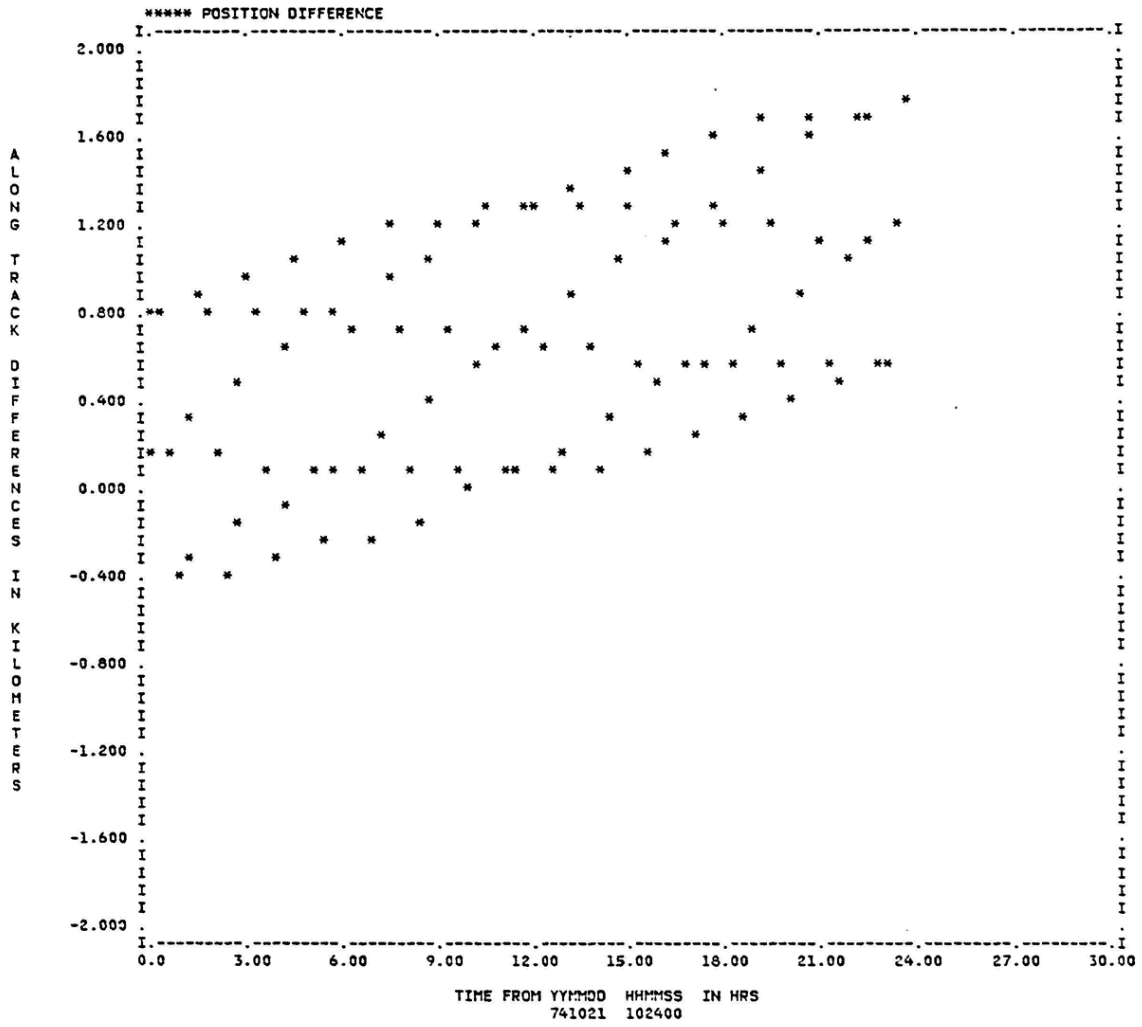
See Figure 3.23 for initial conditions and force model.

Figure 3.30. Cross Track Difference Over 24 Hour Compare Span (21 Hour Predict) for Battin Test Case: PVM Estimated Ephemeris versus Truth Ephemeris



See Figure 3.23 for initial conditions and force model.

Figure 3.31. Along Track Difference Over 24 Hour Compare Span (21 Hour Predict) for Battin Test Case: PVM Estimated Ephemeris versus Truth Ephemeris



See Figure 3.23 for initial conditions and force model.

tion errors which contribute to these error histories. Since the truth ephemeris is circular and equatorial, most of the radial error will come from errors in estimating the elements a, h, and k. An error in estimating a would result in a circular orbit of lower or higher altitude. Errors in h and k would introduce a slight eccentricity into the orbit, such that the maximum difference between the estimated orbit and the true orbit would occur at either apogee or perigee. Using the estimated values of a, h and k given in Table 3.26, it is possible to compute the radial differences at perigee by

$$e_{RAD} = |\vec{r}_{PVM}| - |\vec{r}_{TE}| = a_{PVM}(1 - \sqrt{h^2_{PVM} + k^2_{PVM}}) - a_{TE}$$

$$\simeq -350 \text{ m}$$

This is almost exactly the maximum radial difference found for this test case (see Table 3.26 and Figure 3.29). Similarly, for the particular test case here, the cross track (or out-of-plane) error component appears to be due mostly to errors in estimating p and q. For this near-equatorial case, this is equivalent to an error in estimating the inclination. That is, the estimated orbit is "tipped" slightly out of the equatorial plane by an amount equivalent to

the error in estimating the inclination. The cross track error can therefore be approximated by

$$e_{CT} = a_{PVM} \sin(\Delta i)$$

$$\simeq a_{PVM} \Delta i$$

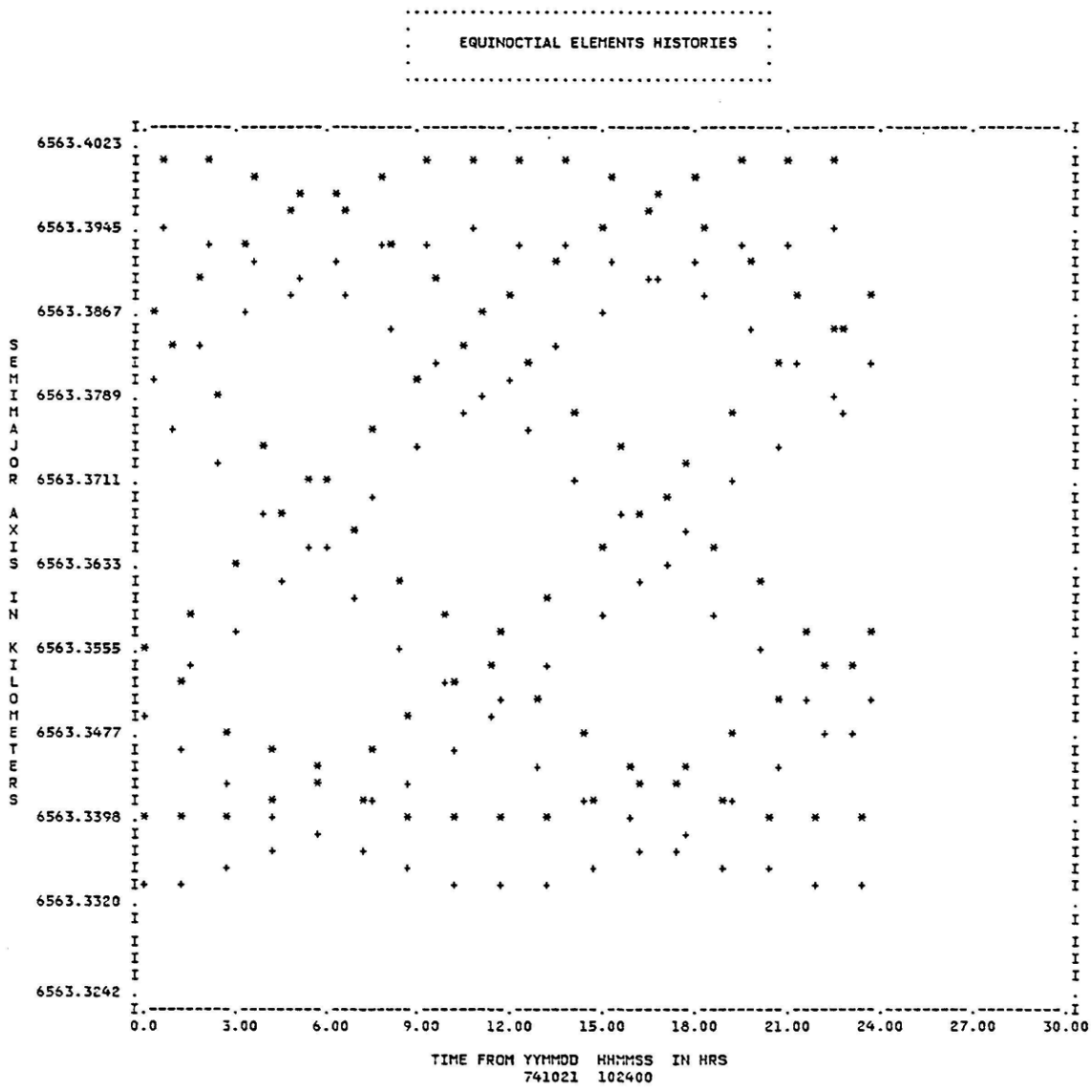
$$\simeq a_{PVM} \left[2 \arctan \left(\sqrt{\Delta p_{PVM}^2 + \Delta q_{PVM}^2}\right)\right]$$

$$e_{CT} \simeq 340 \text{ m}$$

Again this is very near the maximum cross track position error displayed in Table 3.26 and Figure 3.30. The along track error of Figure 3.31 is more difficult to analyze, since it contains errors in estimating the epoch values of h, k and λ and also contains a secular growth due to the error in estimating the semimajor axis and hence the mean motion.

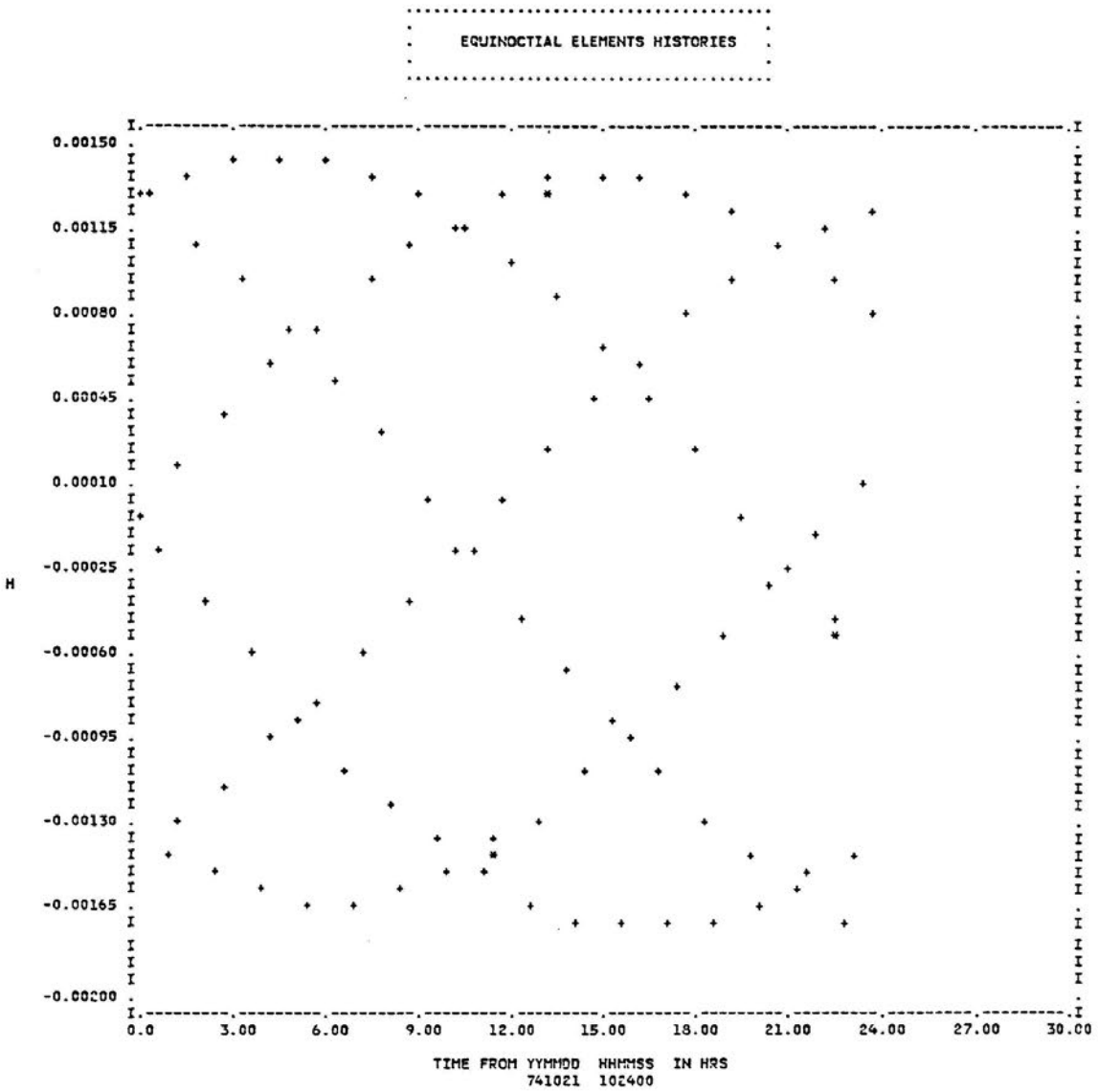
To conclude the analysis of the Battin test case numerical results, Figures 3.32 through 3.37 present comparisons of the equinoctial element histories over the 24 hour compare span. Since the epoch equinoctial elements were the solve-for parameters in this particular test run, these plots give some feel for how the estimation errors propagate

Figure 3.32. Semimajor Axis Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case



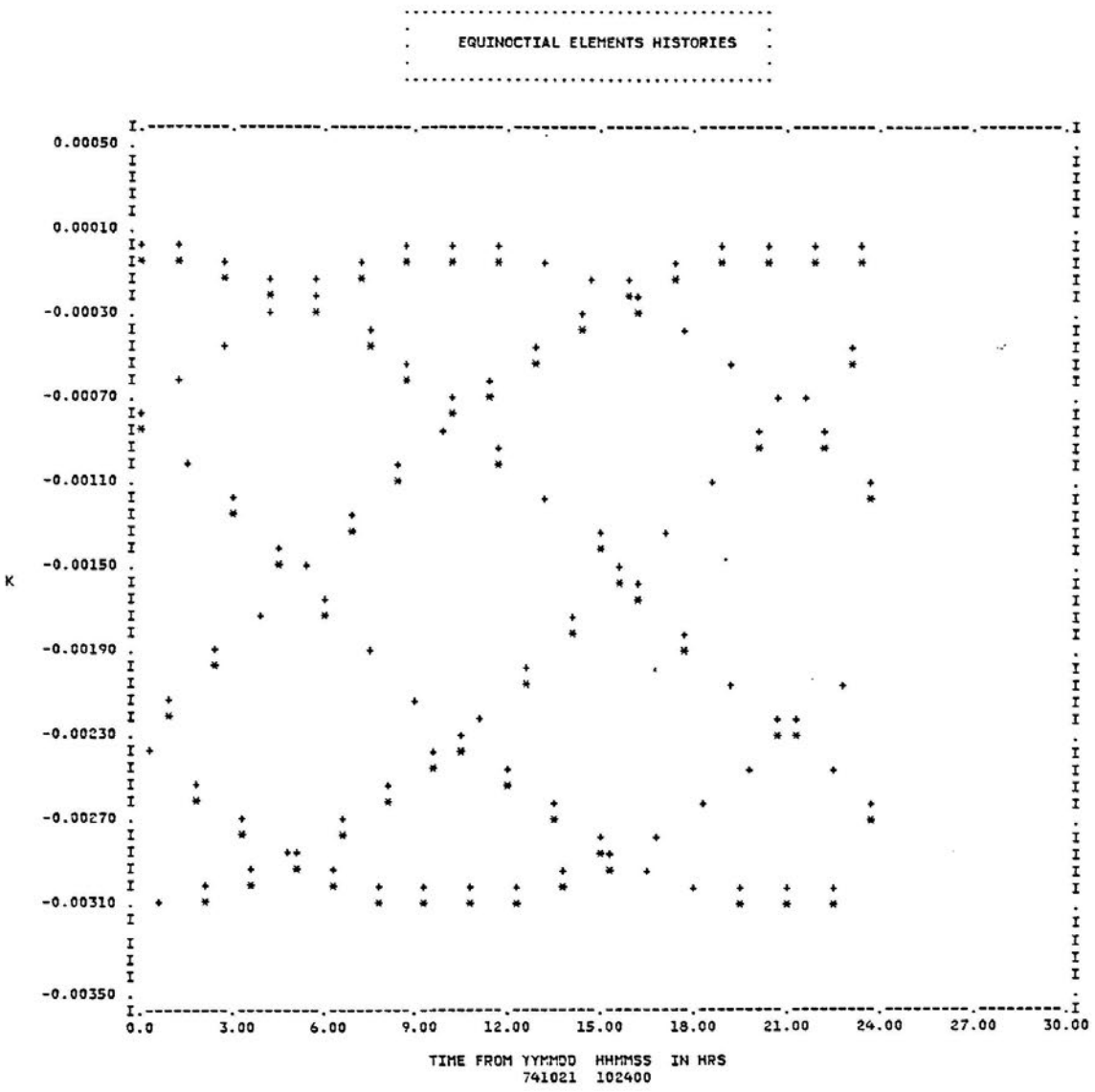
See Figure 3.23 for initial conditions and force model.

Figure 3.33. Equinoctial Element "h" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case



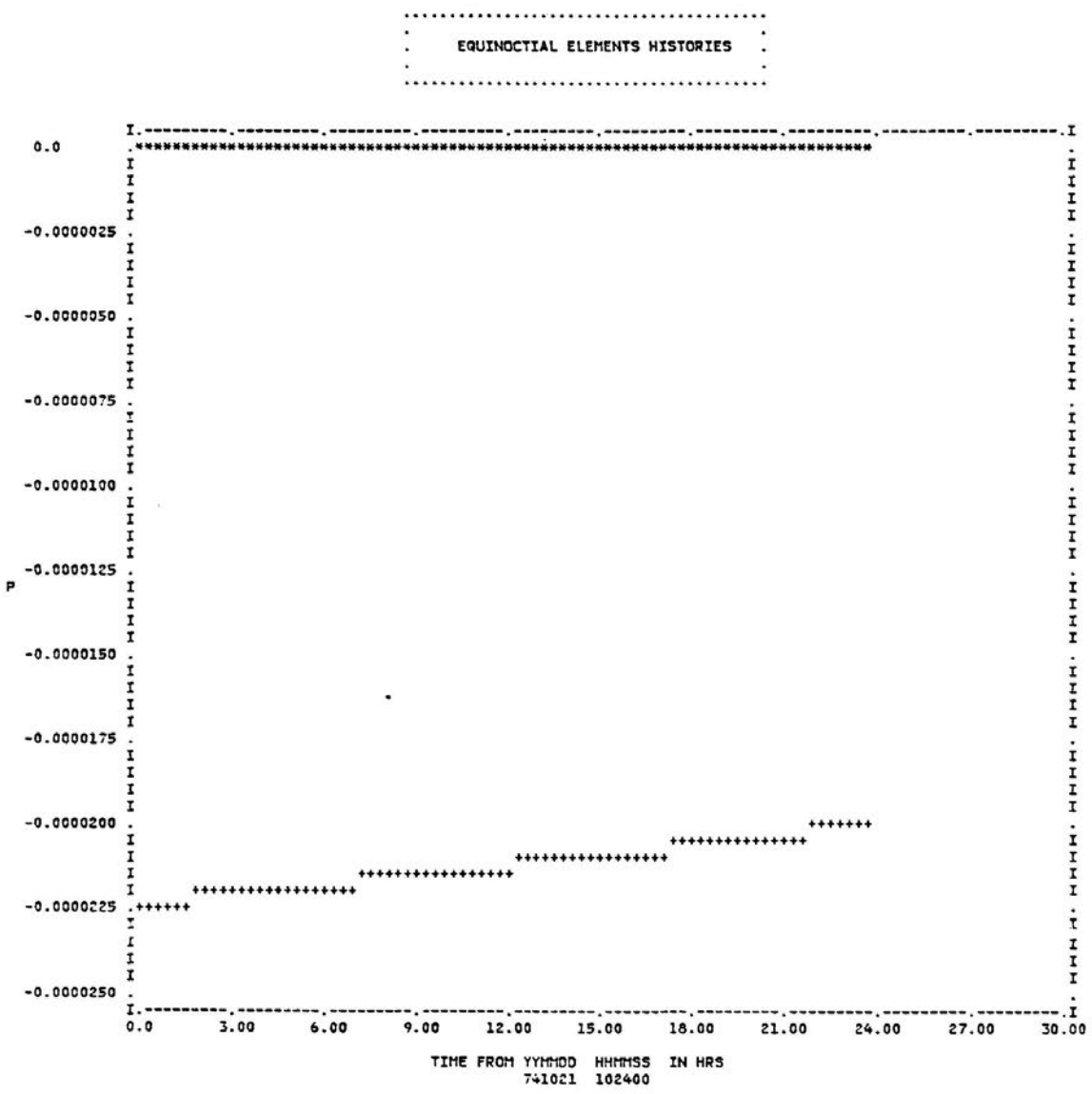
See Figure 3.23 for initial conditions and force model.

Figure 3.34. Equinoctial Element "k" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case



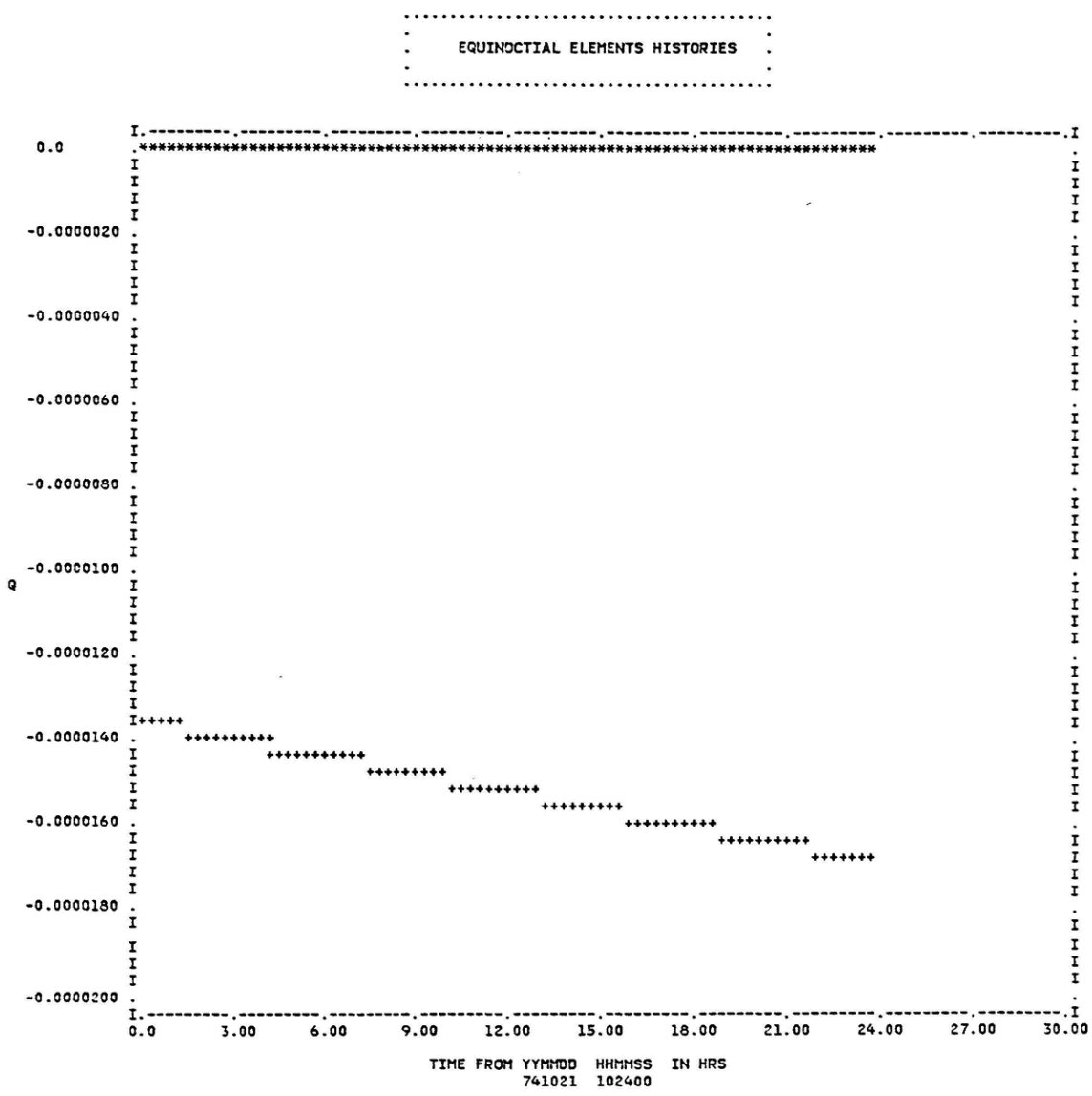
See Figure 3.23 for initial conditions and force model.

Figure 3.35. Equinoctial Element "p" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case



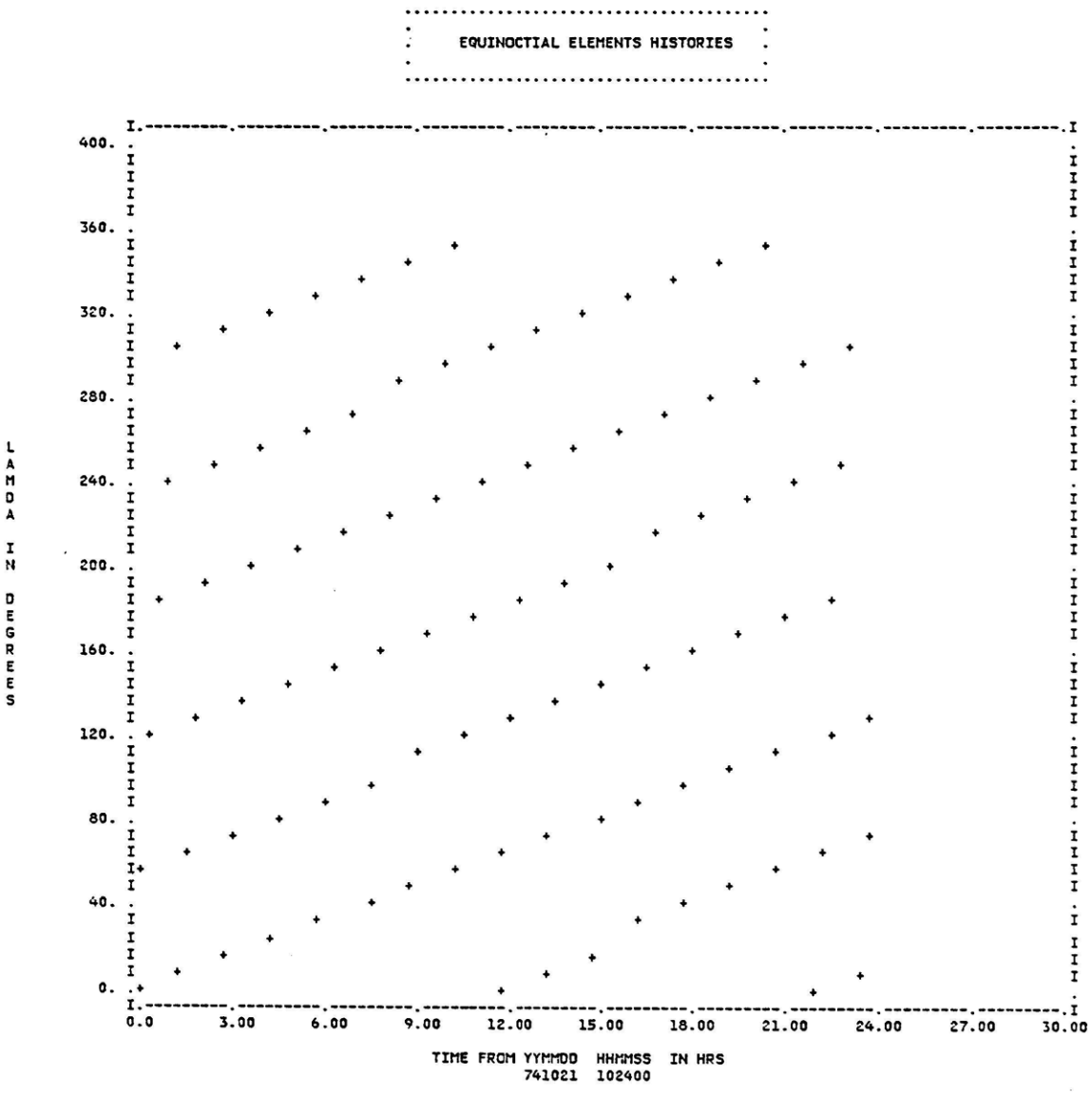
See Figure 3.23 for initial conditions and force model.

Figure 3.36. Equinoctial Element "q" Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case



See Figure 3.23 for initial conditions and force model.

Figure 3.37. Mean Longitude Comparison for 24 Hour Compare Span (21 Hour Predict) for Battin Test Case



See Figure 3.23 for initial conditions and force model.

with time. The plots for a and k reflect initial errors in their estimated values of approximately 7 m and 0.00005 respectively. The initial error in h is too small to be indicated on Figure 3.33. These errors in a, h and k do not change with time, as the force model only includes the J_2 zonal harmonic which has no secular effect on a, e or i. The plots for p and q, however, indicate that the estimation errors for these variables change with time due to the J_2 secular effect on Ω. The errors in the mean longitude were all less than the granularity of the printer plot scale, so no differences appear in Figure 3.37.

3.6.4 Adimurthy and Joy Test Case Results

The previous section indicated that the PVM algorithm provided estimation results comparable in accuracy to the Epoch State Navigation Filter and the conventional Kalman Filter. Additionally, it was shown that the choice of solve-for vector did not materially affect the accuracy of the resulting estimate. However, with respect to the number of iterations to convergence and the overall execution time, the equinoctial solve-for vector gave the best performance. Finally, it was verified that the linearization method provided superior performance over the direct integration method for the required evaluations of the cost functional

and its gradient vector. This superior performance was achieved with little or no degradation in the accuracy of the estimation results.

The work of Adimurthy and Joy (4, 5) indicated that the use of nonlinear optimization techniques may extend the region of convergence in the orbit determination problem. Since the PVM algorithm is a nonlinear function minimization algorithm similar to some of those used by Adimurthy and Joy, it was felt that the PVM method might enjoy some of these improved convergence characteristics. Therefore, this section summarizes the results of the PVM method for the Adimurthy and Joy test case described in Section 3.6.1.2. As in the Battin test case, several combinations of solve-for vectors and methods for evaluating the cost and gradient were used. The PVM results are compared to the estimation results presented in Reference 4.

The initial test runs for this test case were made using the direct integration method and the Cartesian solve-for vector. After 25 minutes of execution time, the method had completed nearly five iterations but had not converged. It was this result, in fact, that motivated the development of the linearization method. Employing

the linearization method on this test case, convergence was achieved for the Cartesian solve-for vector. The results of these tests are summarized in Table 3.27 for the case of 60 position measurements (TC-2.1) and for the case of 30 position and 30 velocity measurements (TC-2.2). The measurement error statistics for these two cases are presented in Table 3.20.

The primary result to note from Table 3.27 is the fact that the linearization approach allowed convergence in approximately 10.5 minutes (18 iterations) of execution time, whereas the direct integration method did not converge in 25 minutes (5 iterations). The resulting estimated values of the epoch position and velocity components agree favorably with their corresponding truth ephemeris values. The standard deviation of the estimation errors in the position components is approximately 10-20 m and, in the velocity components, approximately 1-2 cm/sec. These values appear to be consistent with the measurement errors given in Table 3.20, as do the values of the rms position and velocity errors.

The progress of the PVM algorithm for test case TC-2.1 using linearization and the Cartesian solve-for vector is depicted in Figures 3.38 through 3.45. Figures 3.38 and

Table 3.27. Linearization Method for Adimurthy-and Joy Test Case with Cartesian Solve-For Vector

Solve-for Vector Components	True Solve-for Components	Linearization Method Pos. OBS (TC-2.1) Estimated Values	Linearization Method Pos. OBS (TC-2.1) Est. Error Std. Dev.	Linearization Method Pos./Vel. OBS (TC-2.2) Estimated Values	Linearization Method Pos./Vel. OBS (TC-2.2) Est. Error Std. Dev.
r_x (km)	27.872	27.897	0.0143891	27.874	0.0191274
r_y (km)	6651.742	6651.739	0.0064710	6651.757	0.0089122
r_z (km)	590.641	590.643	0.0120322	590.612	0.0164556
v_x (km/sec)	-6.0443681	-6.0443813	0.0000069	-6.0443671	0.0000099
v_y (km/sec)	0.4740606	0.4740807	0.0000168	0.4740330	0.0000224
v_z (km/sec)	-5.0535911	-5.0535775	0.0000099	-5.0535768	0.0000141
INITIAL COST		3.827×10^{10}		1.827×10^{10}	
CONVERGED COST		5.935×10^{1}		5.021×10^{1}	
INITIAL GRADIENT MAG.		5.747×10^{7}		2.513×10^{7}	
CONVERGED GRADIENT MAG.		1.840×10^{1}		8.133×10^{0}	
NO. OF ITERATIONS		18		18	
EXECUTION TIME		10.483		10.519	
RMS POSITION ERROR		19.842 m		26.760 m	
RMS VELOCITY ERROR		0.021 m/sec		0.028 m/sec	

3.39 plot the component differences between the estimated and true epoch position and velocity vectors. As discussed in the previous section, the algorithm exhibits some transient dynamics in the solve-for parameters during the early iterations. However, by the fourth iteration, these transients have damped out and the algorithm proceeds to converge. Figures 3.40 and 3.41 present plots for the magnitudes of the vector differences between the estimated and true epoch position and velocity vectors for the above test case. Again the early transient behavior is quickly damped and by iteration 6 the magnitudes of the vector differences are very nearly zero. At convergence, the magnitude of the vector difference between the estimated epoch position vector and the true epoch position vector is approximately 25 m. Similarly, the magnitude of the vector difference between the estimated and true epoch velocity vectors at convergence is approximately 3 cm/sec. Both of these values appear consistent with the rms position and velocity errors and with the given measurement error statistics.

Figures 3.42 through 3.45 show the histories of the cost functional and the magnitude of its gradient vector as functions of the iteration number. Figures 3.42 and 3.43 depict these histories over the full 18 iterations required

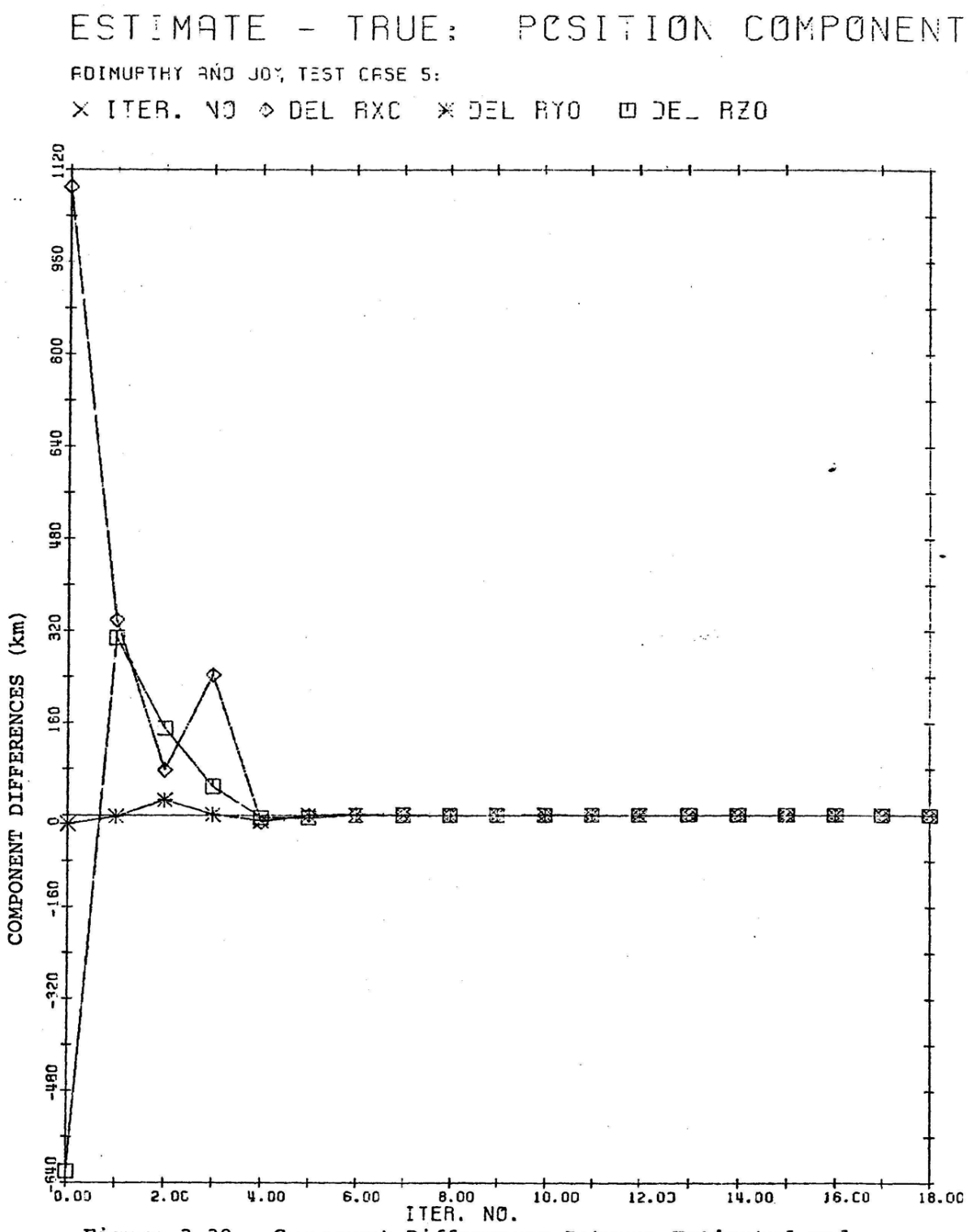


Figure 3.38. Component Differences Between Estimated and True Epoch Position Vectors for Adimurthy and Joy Test Case

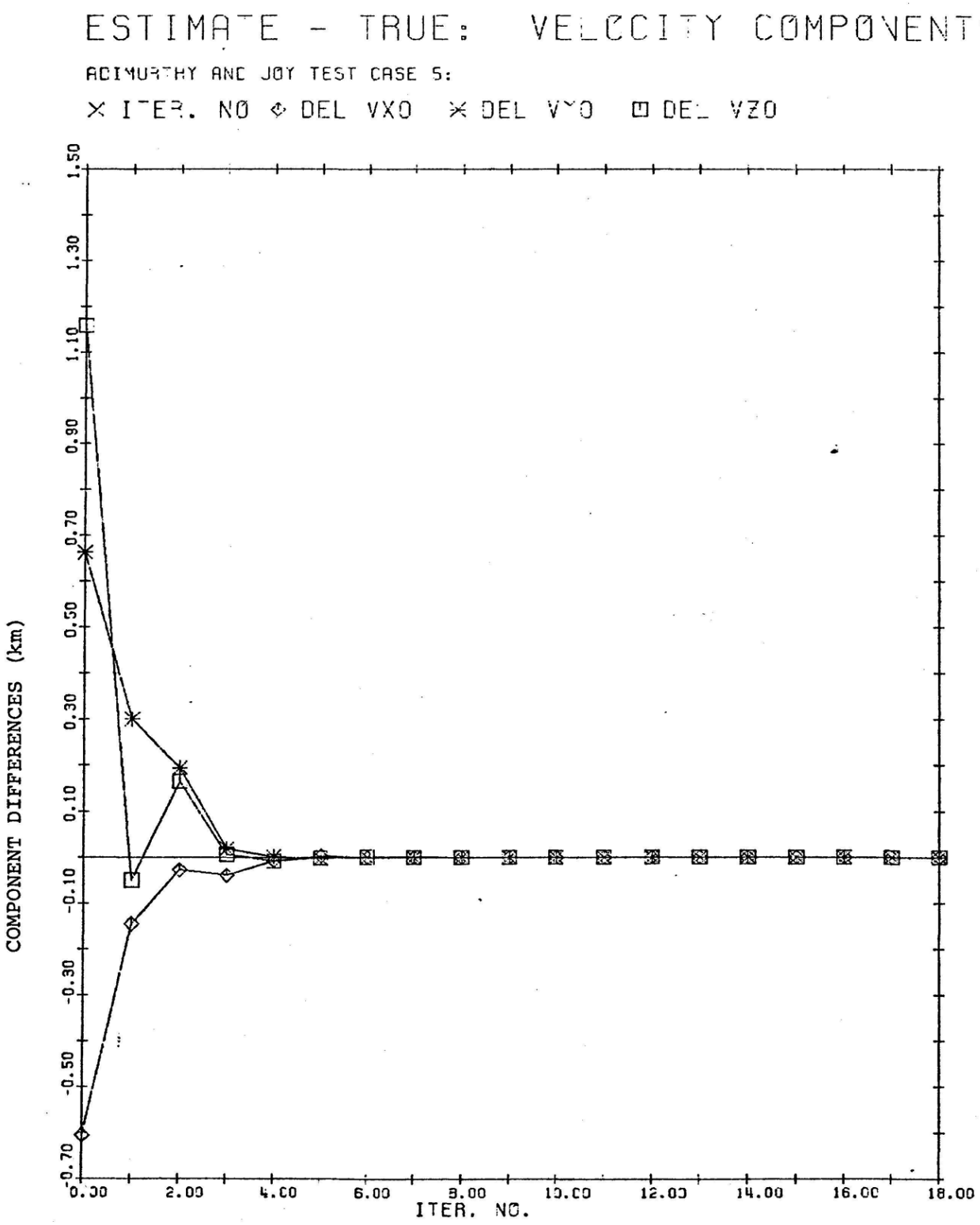


Figure 3.39. Component Differences Between Estimated and True Epoch Velocity Vectors for Adimurthy and Joy Test Case

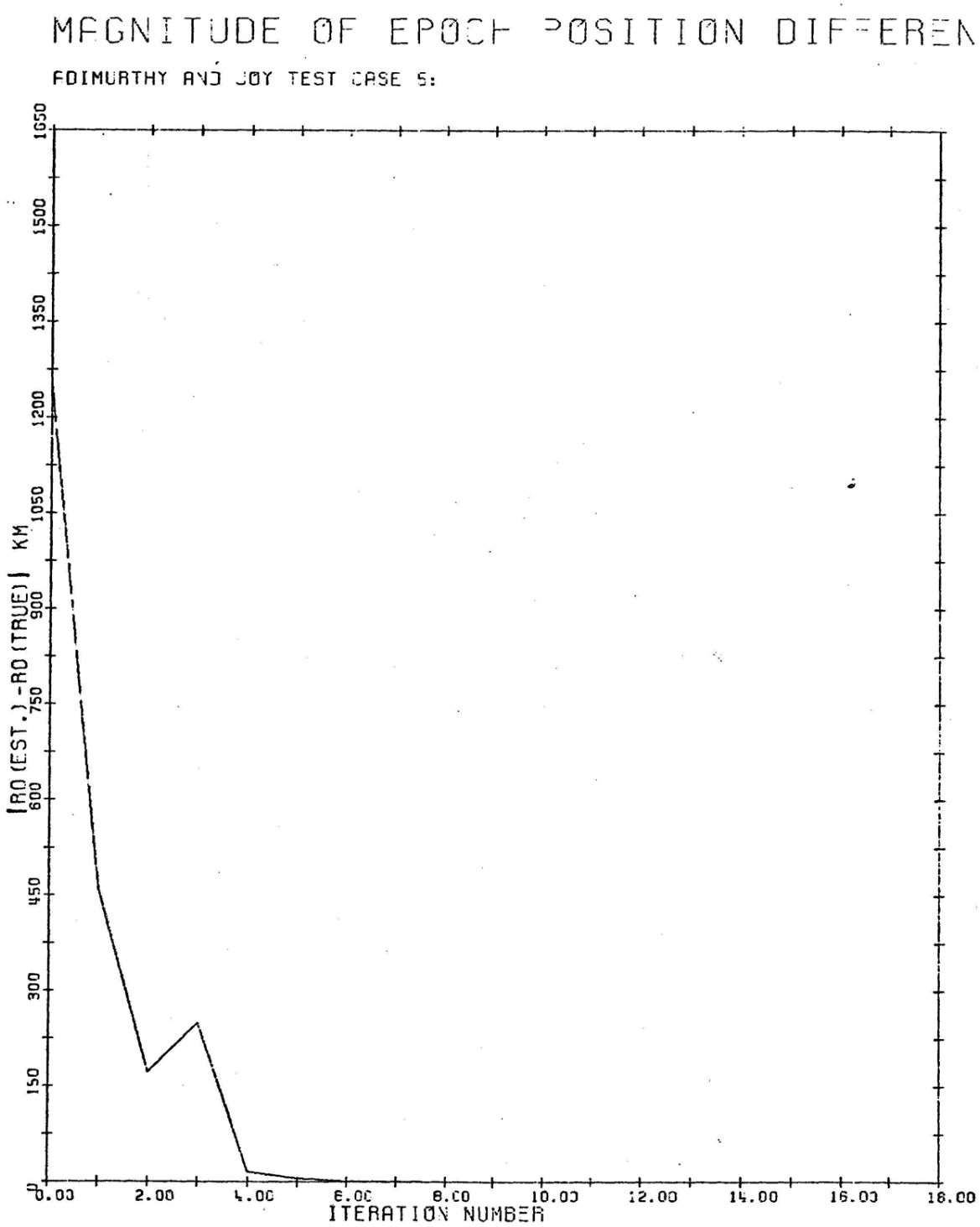


Figure 3.40. Magnitude of Vector Difference Between Estimated and True Epoch Position Vectors for Adimurthy and Joy Test Case

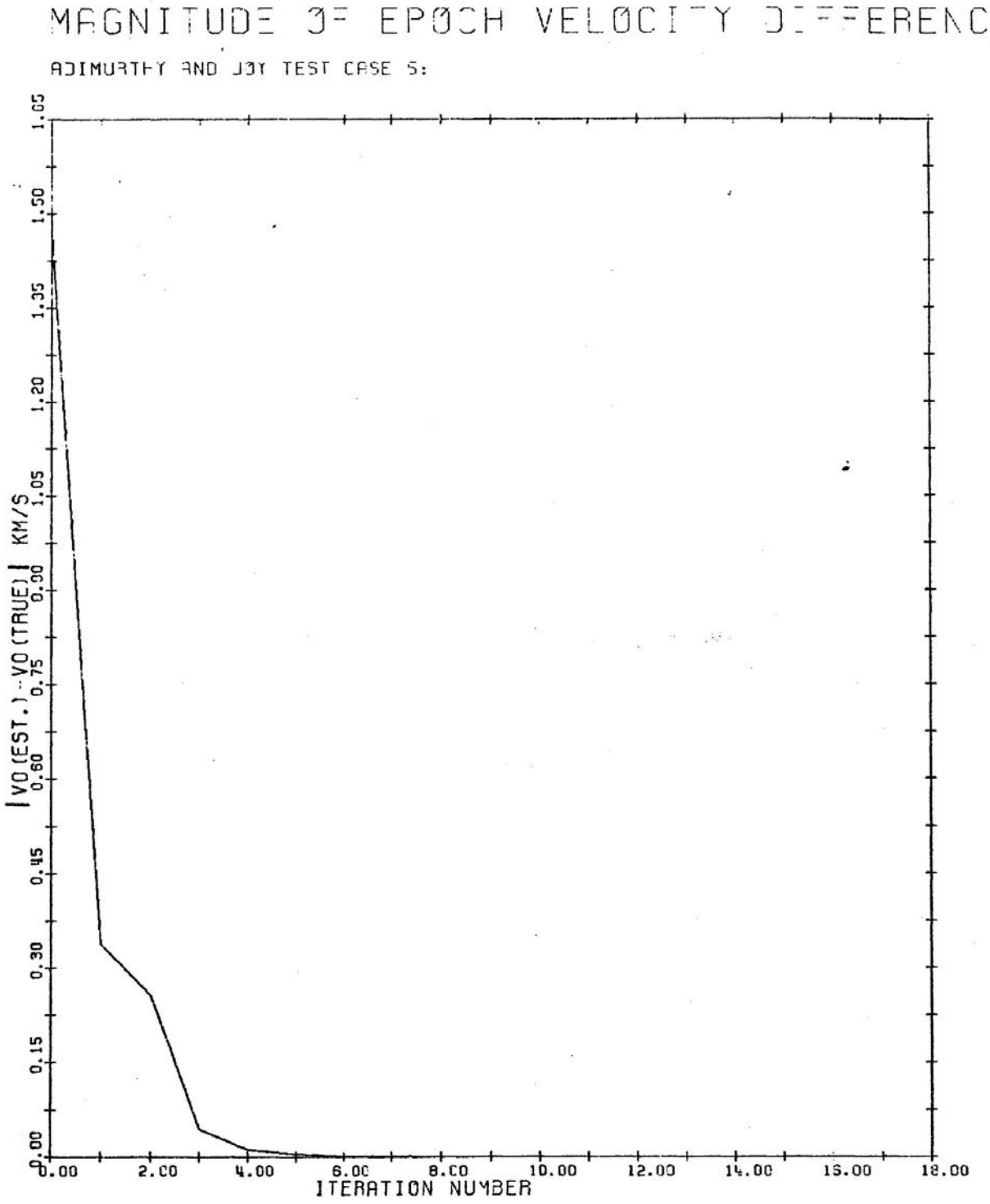


Figure 3.41. Magnitude of Vector Difference Between Estimated and True Epoch Velocity Vectors for Adimurthy and Joy Test Case

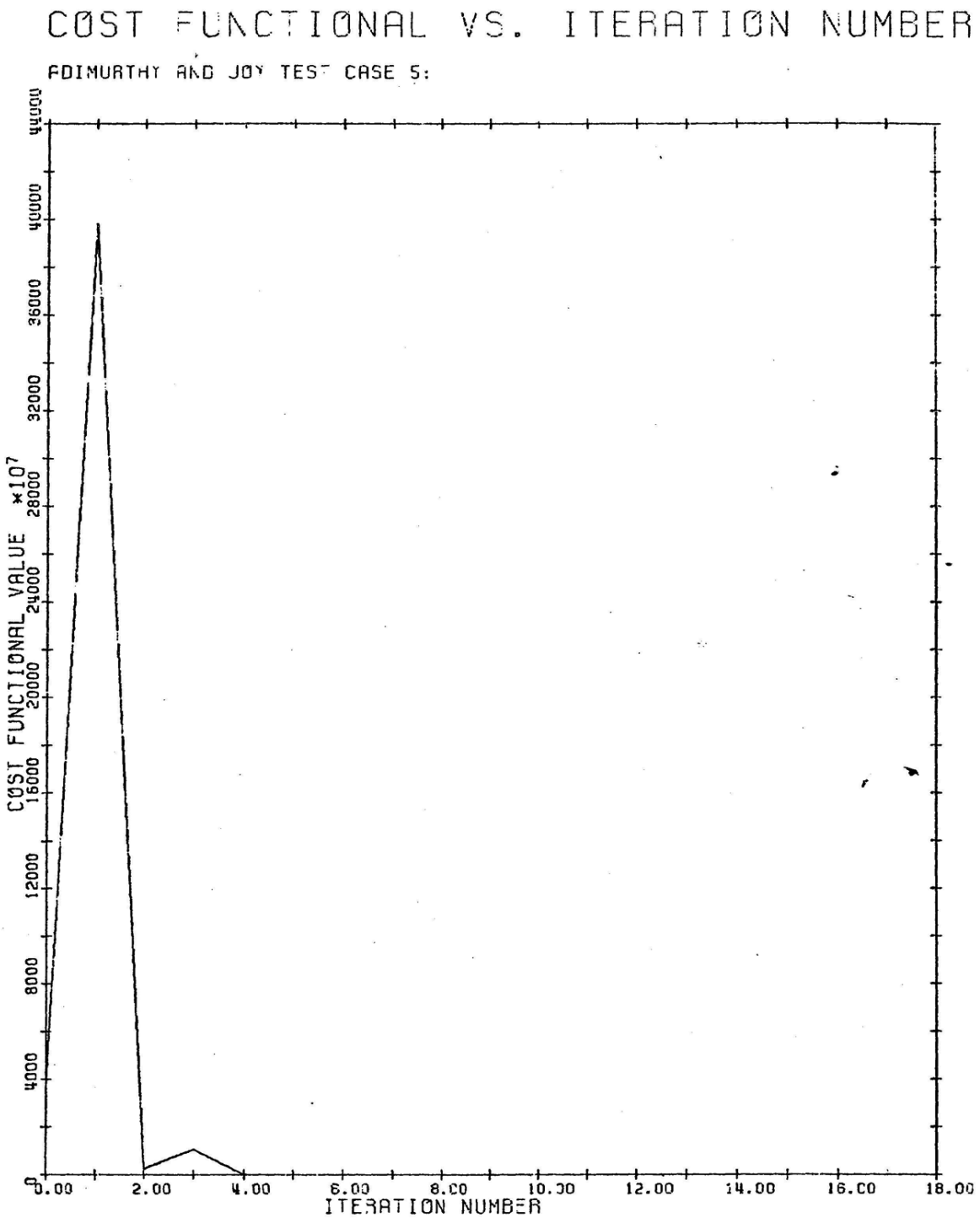


Figure 3.42. Weighted Least Squares Cost Functional vs. Iteration Number for Adimurthy and Joy Test Case

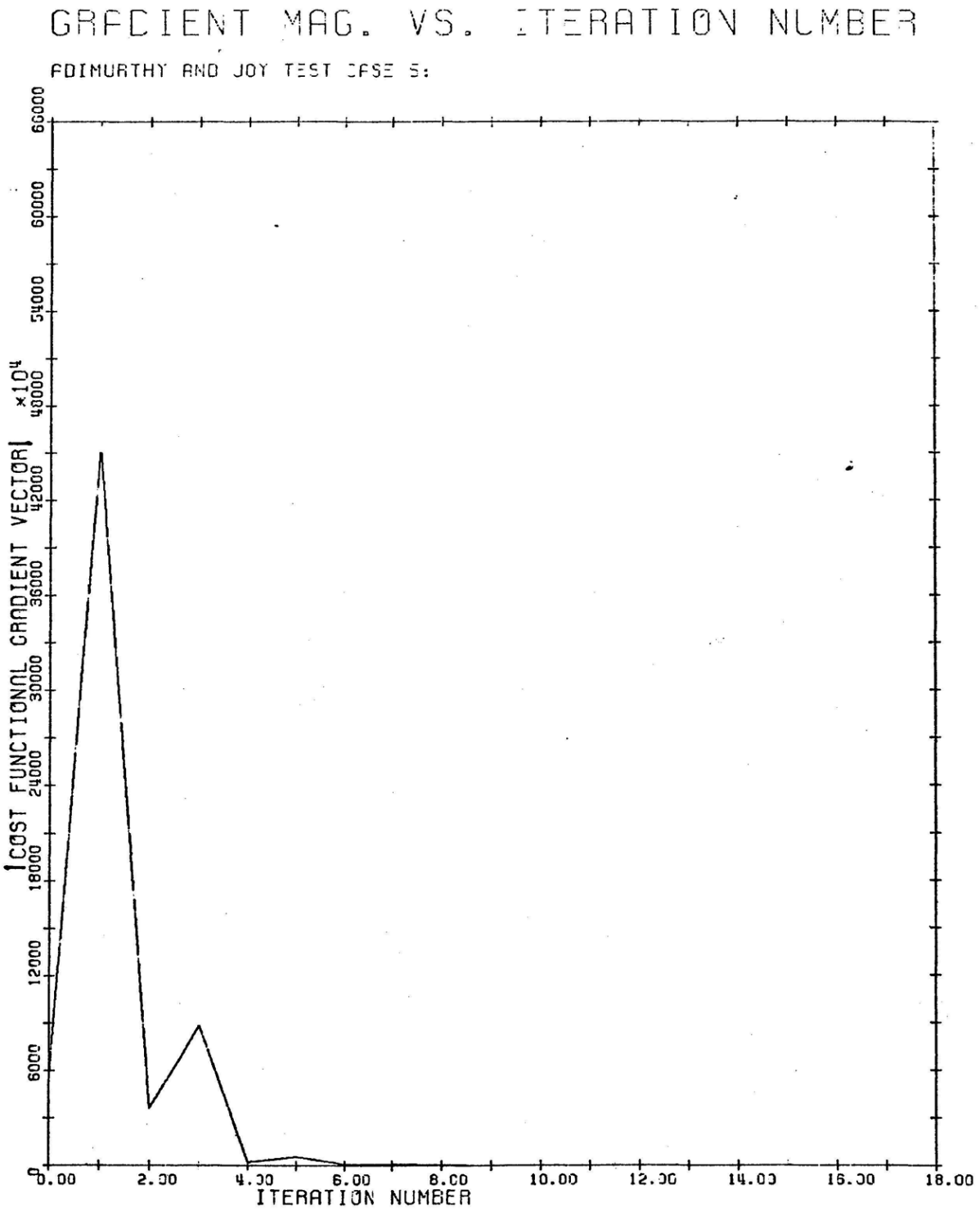


Figure 3.43. Magnitude of Gradient Vector vs. Iteration Number for Adimurthy and Joy Test Case

for the algorithm to converge. The first iteration actually caused the cost and the magnitude of the gradient vector to increase significantly. It is felt that this is due in part to the normal transient behavior of the algorithm. It may also be due in part to the fact that the linearization assumption in the univariate search may be reaching the limits of its validity. For that case, the scalar step-size α_1 to be taken in the search direction $\vec{s}_1$ could be in error. However, the PVM algorithm is sufficiently robust to compensate for this and the next iteration decreases the cost and the magnitude of the gradient vector to well below their *a priori* values. Figures 3.44 and 3.45 show the cost and gradient magnitude over the final 9 iterations for this test case. The slower rate of convergence discussed previously is again exhibited here. The first eleven iterations reduce the cost from 4×10^{10} to 200 and the gradient magnitude from 6×10^{7} to 1×10^{4}. However, the last seven iterations only reduce the cost functional from 200 to 60. The gradient magnitude, on the other hand, is significantly reduced from 1×10^{4} to 20 in the last seven iterations.

The results from the Battin test case indicated that the linearization method provided results as accurate as those found using the direct integration method, but with

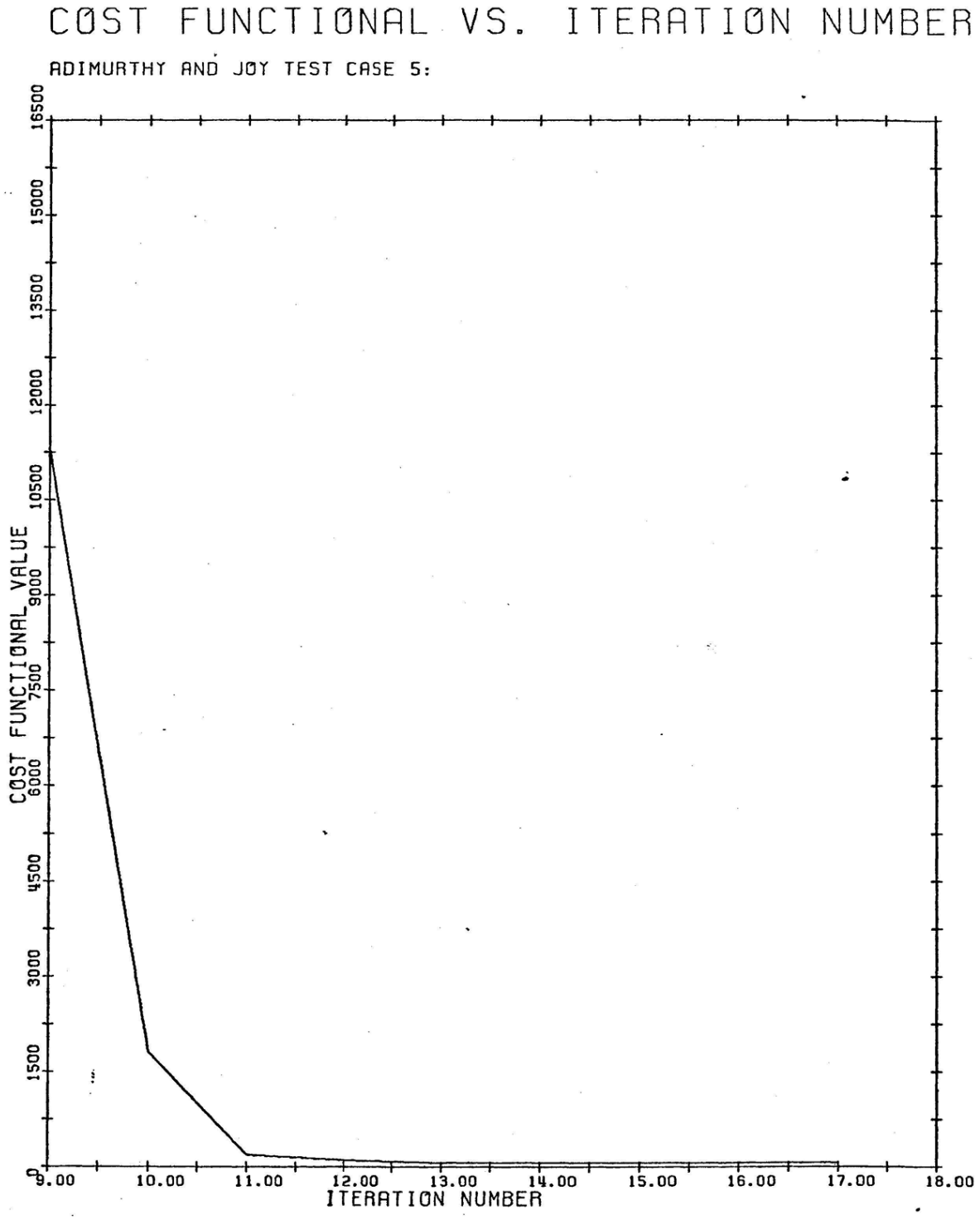


Figure 3.44. Cost Functional vs. Iteration Number for Adimurthy and Joy Test Case: Iterations Nine Through Eighteen

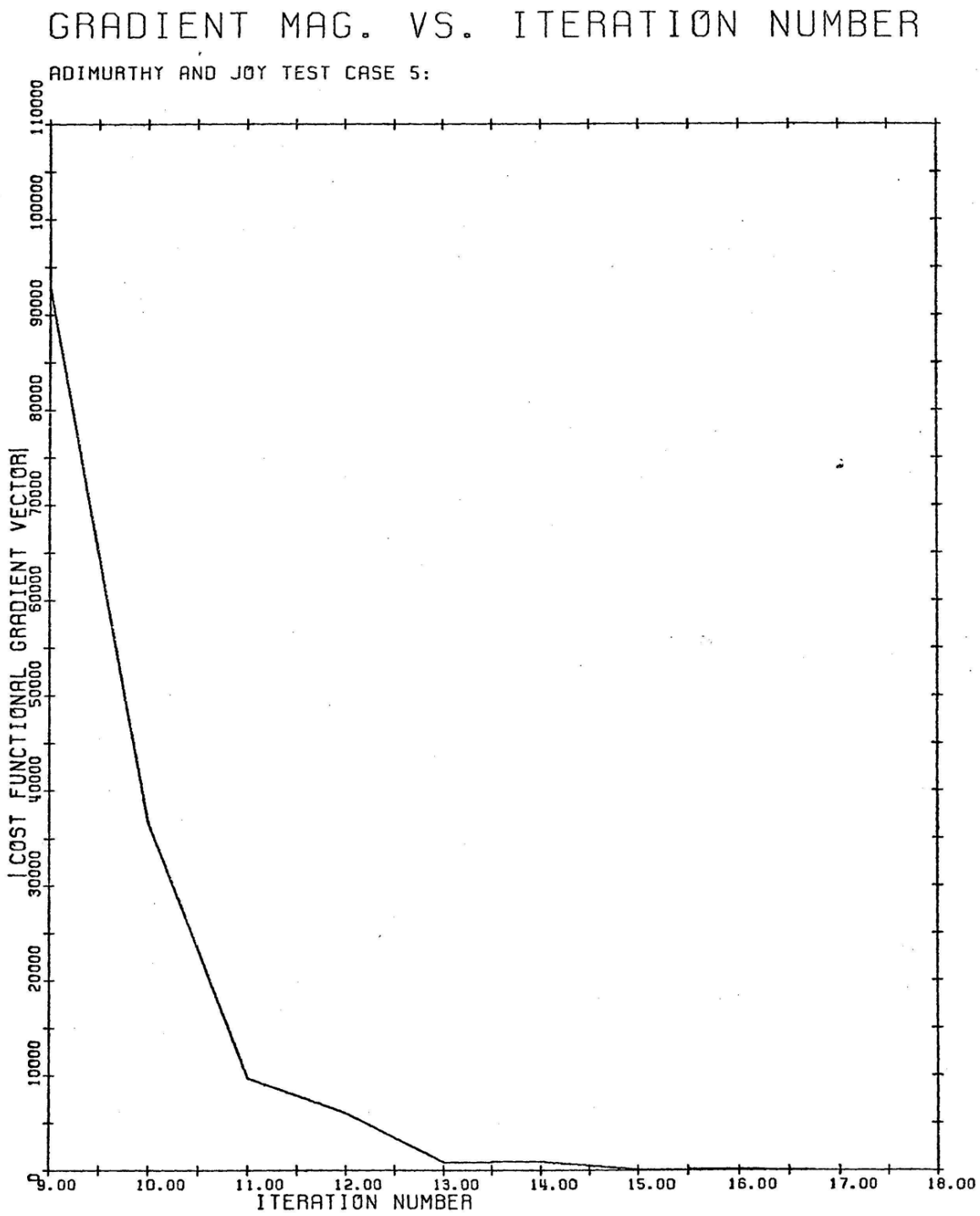


Figure 3.45. Magnitude of Gradient Vector vs. Iteration Number for Adimurthy and Joy Test Case: Iterations Nine through Eighteen

much improved performance. It was also found that the use of the equinoctial solve-for vector with the linearization method gave the best performance results. To test these conjectures using the more strenuous Adimurthy and Joy test case, two runs were made using the simulated data of TC-2.1 and the equinoctial solve-for vector. One run used the direct integration method for evaluating the cost functional and its associated gradient vector, while the other run used the linearization method. These results are summarized in Table 3.28.

Clearly, the most dramatic contrast between these two runs is evident in the execution times. The direct integration run took 23.6 minutes of the execution time while the linearization run executed in approximately 2.7 minutes. Examination of the estimated epoch equinoctial elements and their associated estimation error statistics produced by these two methods reveals virtually no difference in accuracy. The estimated epoch values produced by either method compare very favorably with their true values. It should be noted that, although the direct integration method with equinoctial solve-for vector took 23.6 minutes to converge, it is still quite superior to the direct integration method with Cartesian solve-for vector, which was still far from convergence after 25 minutes of execution

Table 3.28. Direct Integration vs. Linearization Method for Adimurthy and Joy Test Case with Equinoctial Solve-for Vector

Solve-for Vector Components	True Solve-for Components	Direct Integration Method Pos. OBS (TC-2.1) Estimated Values	Direct Integration Method Pos. OBS (TC-2.1) Est. Error Std. Dev.	Linearization Method Pos. OBS (TC-2.1) Estimated Values	Linearization Method Pos. OBS (TC-2.1) Est. Error Std. Dev.
a (km)	6983.208	6983.206	0.0014	6983.206	0.0014
h	0.0436810	0.0436812	0.0000006	0.0436812	0.0000006
k	0.0015925	0.0015924	0.0000006	0.0015924	0.0000006
p	-0.3647492	-0.3647479	0.0000008	-0.3647479	0.0000008
q	0.0369020	0.0369007	0.0000008	0.0369007	0.0000008
λ (deg)	87.912	87.912	0.0001177	87.912	0.0001177
INITIAL COST		3.827×10^{10}		3.827×10^{10}	
CONVERGED COST		5.935×10^{1}		5.935×10^{1}	
INITIAL GRADIENT MAG.		1.906×10^{7}		1.906×10^{7}	
CONVERGED GRADIENT MAG.		9.522×10^{-2}		2.214×10^{-1}	
NO. OF ITERATIONS		6		4	
EXECUTION TIME (MINS.)		23.615		2.644	

time. Again these results tend to confirm the quality of the estimation results produced by the linearization method and the performance advantages of the equinoctial solve-for vector.

To attempt to compare the PVM estimation results to the results of Adimurthy and Joy (4), the Keplerian solve-for vector was used. A direct comparison of results was not possible because the force model used by Adimurthy and Joy contained zonal harmonics J_2 through J_4 and drag while the force model used in the PVM algorithm contained zonal harmonics J_2 through J_8 and no drag. Additionally, Adimurthy and Joy used a nonlinear radar observation model providing range, azimuth and elevation and a pass-oriented observation simulation while the PVM results were based on a linear observation model (observations were Cartesian position components) and these observations were distributed relatively uniformly over two satellite revolutions. Although these are somewhat significant differences, it is felt that the Adimurthy and Joy results represent a legitimate baseline against which the PVM results can be compared.

Table 3.29 presents the comparison of the PVM results to those presented in Table 8 of Reference 4. The PVM results use the linearization method and the Keplerian

Table 3.29. Orbit Determination Accuracies: PVM Results vs. Adimurthy and Joy Results

		Linearization Method Pos. OBS (TC-2.1)		Adimurthy and Joy Results	
				45 OBS./1 Pass	18 OBS/3 Passes
Solve-for Vector Components	True Solve-for Components	Estimated Values	Est. Error Std. Dev.	Est. Error Std. Dev.	Est. Error Std. Dev.
a (km)	6983.208	6983.206	0.001	0.74	0.001
e	0.04371	0.04371	0.0000006	0.00009	0.000012
i (deg)	40.267	40.267	0.00008	0.009	0.0014
ω (deg)	172.135	172.135	0.0008	0.0007	0.0009
Ω (deg)	275.777	275.777	0.00001	0.001	0.00019
M (deg)	0.0	-0.0003694	0.0008	0.0007	0.0007
INITIAL COST		3.827×10^{10}			
CONVERGED COST		5.935×10^{1}			
INITIAL GRADIENT MAG.		2.683×10^{7}			
CONVERGED GRADIENT MAG.		4.398×10^{-2}			
NO. OF ITERATIONS		5			
EXECUTION TIME		3.320			

solve-for vector. The estimated values of the epoch Keplerian elements produced by the PVM method again compare very favorably to their truth ephemeris values. Unfortunately, Adimurthy and Joy do not give the actual element values for comparison, but they do give the standard deviations of the estimation errors for their estimates of the Keplerian elements. These are listed in the last two columns of Table 3.29 for the cases of 45 observations from one pass and 18 observations from three passes. It can be seen that the estimation error statistics from the PVM method are quite comparable to those given in Reference 4. Adimurthy and Joy noted that, when their test algorithms converged, the accuracies of the estimates for the orbital elements were of the same order. The same appears to be true for the PVM algorithm of Straeter.

It is also interesting to note that the PVM algorithm using the linearization method and the Keplerian solve-for vector converged in 3.3 minutes on the Amdahl 470 V6 computer. Thus, for the strenuous Adimurthy and Joy test case the equinoctial solve-for vector gave the most rapid convergence, followed by the Keplerian and then the Cartesian solve-for vectors. This is in contrast to the Battin test case, for which the equinoctial solve-for vector gave the best performance, but the Cartesian solve-for

was superior to the Keplerian. It is felt that this is due to the poorly defined physical nature of some of the Keplerian solve-for parameters in the Battin test case, which causes extra iterations in the PVM method. In general, the test case results indicate that the equinoctial solve-for vector usually gives the best performance results, followed by the Keplerian and the Cartesian solve-for vectors.

The PVM method converged for all three choices of solve-for vector for the Adimurthy and Joy test case. The initial guess for the solve-for parameters (see Table 3.19) is well beyond the region of convergence of the classical differentical corrections method, as has been shown in Reference 4. In fact, of the nine methods tested by Adimurthy and Joy, only Stewart's version of the Davidson-Fletcher-Powell algorithm and Fletcher's dual variable metric algorithm converged for this test case. It should be pointed out that both of these methods are variable metric algorithms, as is Straeter's Parallel Variable Metric algorithm discussed in this work. It is felt that the above results indicate that the PVM method displays exceptionally robust convergence properties with respect to the quality of the *a priori* estimate of the solve-for vector.

Several runs were made to attempt to stress the PVM algorithm beyond the original Adimurthy and Joy test case. The linearization method with equinoctial solve-for vector was used in all runs. The results of Adimurthy and Joy and experience using the differential corrections estimation capability in GTDS-RD indicated that the convergence of orbit estimation methods could be very sensitive to errors in the *a priori* value of the mean anomaly. Therefore, the original *a priori* values for a, h, k, p and q given in Table 3.19 were retained. However, in addition to the original *a priori* error of 10 degrees in mean anomaly, errors of 90, 120, 150 and 180 degrees were included. As before, the force model included zonal harmonics J_2 through J_8. The radius of convergence for the PVM algorithm appears quite extensive, as shown by the results summarized in Table 3.30. In all cases for which the PVM algorithm converged, the final estimated values for the solve-for parameters were virtually identical to those listed in Table 3.28 for the $\Delta M = 10$ degree case. The $\Delta M = 180$ degree case failed to converge because an early estimate of the solve-for vector represented an orbit for which the Picard-Chebyshev integration method could not converge. This may happen due to the transient behavior of the PVM algorithm during the initial few iterations. While this represents

Table 3.30
Convergence Results for the Parallel
Variable Metric Orbit Estimation Algorithm

ΔM (deg)	CONVERGENCE	NUMBER OF ITERATIONS	EXECUTION TIMES (MINS)
10	YES	4	2.664
90	YES	7	4.389
120	YES	7	4.298
150	YES	9	5.591
180	NO	16	20.0

LINEARIZATION METHOD

EQUINOCTIAL SOLVE-FOR VECTOR

TC-2.1 OBSERVATIONAL DATA

FORCE MODEL: $J_2 - J_8$, NO DRAG

a kind of limitation on the algorithm, certainly the case of a 180 degree error in mean anomaly is not very realistic in an operational sense.

Because of concern that the effect of drag may change the character of the estimation problem, a new set of simulated observational data was generated to include drag. The force model for this test case included zonal harmonics J_2 through J_4 and drag, with the drag acceleration calculated as shown in Appendix B. Thus, this set of simulated observational data more closely approximates the Adimurthy and Joy test cases.

Table 3.31 displays the PVM estimation results for two sets of *a priori* solve-for parameters. The first case is simply the basic set of *a priori* values given in Table 3.19. The second case includes an error in mean anomaly of 60 degrees, keeping all other *a priori* values the same. The results indicate that, for this particular orbit, the influence of drag does not degrade the accuracy of the estimation results. In fact, comparison between Tables 3.29 and 3.31 indicates that the converged value of the cost functional in the drag case is actually slightly less than that for the case of zonals J_2 through J_8. This may be due to the fact that neglecting the zonals

Table 3.31. Adimurthy and Joy Test Case Estimation Results with Drag Effects Included in the Force Model

		ΔM = 10° (Table 3.19) Pos. OBS (TC-2.1)		ΔM = 60° Pos./Vel. OBS (TC-2.1)	
Solve-for Vector Components	True Solve-for Components	Estimated Values	Est. Error Std. Dev.	Estimated Values	Est. Error Std. Dev.
a (km)	6983.208	6983.209	0.001	6983.209	0.001
e	0.04371	0.04371	0.0000006	0.04371	0.0000006
i (deg)	40.267	40.267	0.00008	40.267	0.00008
ω (deg)	172.135	172.135	0.0008	172.135	0.0008
Ω (deg)	275.777	275.777	0.0001	275.777	0.0001
M (deg)	0.0	-0.0002748	0.0008	-0.0002748	0.0008
INITIAL COST		3.827×10^{10}		8.156×10^{11}	
CONVERGED COST		5.627×10^{1}		5.627×10^{1}	
INITIAL GRADIENT MAG.		2.683×10^{7}		3.377×10^{8}	
CONVERGED GRADIENT MAG.		5.591×10^{-4}		1.260×10^{-2}	
NO. OF ITERATIONS		6		6	
EXECUTION TIME		3.568		3.551	

J_5 through J_8 in the drag case reduces the amplitudes of the zonal short periodics present in the integrated trajectory. This in turn could reduce the observation residuals and hence reduce the cost functional value. At any rate, it appears that the PVM algorithm will produce accurate estimation results when drag is included in the force model.

However, the inclusion of the drag acceleration seems to significantly decrease the region of convergence for the PVM algorithm. The results for an *a priori* mean anomaly error of 60 degrees are included in Table 3.31. It is seen that the results are identical to the $\Delta M = 10°$ test case, with a very similar run time. For test runs with *a priori* mean anomaly errors greater than 60°, the PVM algorithm as presently formulated could not converge. It appears that drag definitely limits the convergence region for this method. In most cases, this was due again to early iterates of the solve-for vector yielding orbits which could not be handled by the Picard-Chebyshev integrator. For example, the results of the first iteration for the case of $\Delta M = 90°$ yielded a set of epoch Keplerian elements corresponding to a position vector whose magnitude was 200 km less than the mean equatorial radius of the Earth. When this position vector is used as an initial

condition for the numerical integration of the reference trajectory for the next iteration, the numerical integrator breaks down because the exponential drag model returns extremely large drag accelerations. Despite this reduction in the potential radius of convergence due to drag, the PVM algorithm still appears very robust in comparison to classical differential correction methods and even in comparison to most nonlinear function minimization techniqes.

Chapter 4

ALGORITHM PERFORMANCE EVALUATION

Chapters 2 and 3 have formulated algorithms for performing the basic tasks required in an orbit determination system. These algorithms have been structured for a Single Instruction Stream/Multiple Data Stream (SIMD) hardware architecture, as discussed in Chapter 1. Executions of the algorithms in sequential simulations have indicated that these parallel algorithms produce results of very high accuracy. Additionally, the parallel orbit estimation algorithm appears to enjoy a significantly expanded region of convergence over classical methods in the estimation problem.

It was conjectured in the introduction to this thesis that these algorithms, when coupled with the SIMD hardware architecture, would display very favorable performance characteristics. This chapter will examine this potential performance improvement. An evaluation criterion will be formulated and speed-up ratios will be calculated. These speed-up ratios will be based on a set of broad assumptions

and will represent an order-of-magnitude measure of performance improvement. It appears that the potential for a significant performance improvement certainly exists.

4.1 Performance Measures

The target hardware architecture for the algorithms developed in Chapters 2 and 3 was the SIMD array processor depicted in Figure 1.1. As previously stated, the hardware architecture is such that a single stream of instructions is executed simultaneously on multiple streams of data. This parallel execution is accomplished by the identical processing elements and their attendant local memories shown in Figure 1.1.

In order for a program to take advantage of this type of hardware architecture, the program must contain large segments of identical code which could be executed simultaneously. For example, suppose the parallel processor contained N identical data streams. If a program could be decomposed into N identical code segments, then each segment could be executed in one of the N identical processing elements in the parallel processor under the overall supervision of the control unit. The program would therefore execute on the parallel processor in 1/N the amount of time

required for the same program to execute on a sequential processor.

The ability to decompose totally a program into parallel segments is an idealistic case. However, if major portions of the program can be cast into parallel segments then relatively significant performance improvements approaching the limiting value of 1/N can be achieved. This has been the primary factor influencing the structure of the numerical algorithms formulated in Chapters 2 and 3.

A straightforward measure of the performance improvement is needed for both the parallel orbit propagation algorithm and the parallel orbit estimation algorithm. Straeter presents several numerical examples of his basic Parallel Variable Metric (PVM) algorithm in Reference 139. His results tabulate the total number of cost functional and gradient evaluations required for the method to converge on a serial computer. The algorithm is designed such that the cost and gradient evaluations required in the first step of the method can be computed in parallel. Therefore, Straeter also tabulates the effective number of cost and gradient evaluations that would be required for algorithm convergence if the same computations had been done on a parallel processor.

The computations of the cost and gradient represent significant portions of the PVM algorithm. The number of such evaluations required for convergence of a gradient method can be considered as a sort of performance measure for that method. Using the number of cost functional and gradient evaluations as a performance measure allows a comparison of the performance of a parallel execution of the PVM algorithm relative to a sequential execution of the same algorithm. This relative measure will be called the estimator speed-up ratio S_e and will be defined by

$$S_e = \frac{\left[\begin{array}{l}\text{number of cost and gradient evaluations} \\ \text{required for convergence on a serial} \\ \text{computer}\end{array}\right]}{\left[\begin{array}{l}\text{effective number of cost and gradient} \\ \text{evaluations required for convergence} \\ \text{on a parallel computer}\end{array}\right]} \tag{4.1}$$

An analogous performance measure must be formulated for the parallel orbit propagation algorithm. In this case, the major segments of the program amenable to parallel execution are the evaluations of the right-hand sides of the differential equations of motion. The numerical integration of these equations of motion over a given integration span will require differing numbers of forcing function

evaluations, depending on the type of numerical integration algorithm in use and the integration stepsize. The number of these force evaluations required to provide a certain level of accuracy over a given integration span can be considered as a performance measure of the particular integration algorithm in use.

The Picard-Chebyshev integration algorithm is designed such that all forcing function evaluations for a single iteration over a single integration step can be accomplished in parallel. This results in the effective number of forcing function evaluations being greatly reduced in the Picard-Chebyshev method. As in the case of the orbit estimator, a speed-up ratio can be defined which will be indicative of the performance of a parallel execution of the Picard-Chebyshev algorithm relative to a sequential execution of this or any other integration algorithm over some given span. This speed-up ratio will be called the integrator speed-up ratio S_i and will be defined by

$$S_i = \frac{\left[\begin{array}{l}\text{number of force evaluations required} \\ \text{in the integration span for a serial} \\ \text{numerical integration algorithm}\end{array}\right]}{\left[\begin{array}{l}\text{effective number of force evaluations} \\ \text{required in the integration span for} \\ \text{the Picard-Chebyshev numerical inte-} \\ \text{gration algorithm}\end{array}\right]} \tag{4.2}$$

The speed-up ratios given above will serve as relative performance measures for the orbit propagation and orbit estimation algorithms in the following discussions.

4.2 Orbit Propagator Performance Results

The accuracy of the Picard-Chebyshev numerical integration algorithm was tested in Chapter 2.4 for two eccentricities and several different sets of algorithm parameters. The epoch conditions for the low altitude circular test case are given in Table 2.3 while those for the low altitude elliptical test case are given in Table 2.4. Table 2.5 presents a breakdown of the different algorithm parameters for the Picard-Chebyshev (P-C) integration of the low altitude circular test case. Table 2.6 gives similar data for the elliptical test case.

The test runs of Section 2.4 also provide information necessary to compute the integrator speed-up ratio defined by Equation (4.2). The results for the low altitude circular test case are given in Table 4.1. In this table, the run numbers in the first column can be referenced to the test run descriptions in Table 2.5 for a complete description of the associated algorithm parameters. The second column indicates the integration span of interest.

Table 4.1. Speed-Up Ratio Computation for Low Altitude Circular Test Runs

Run #	Int. Span	P-C Algorithm Parameters		Effective Force Evaluations			Speed-Up* Ratio
		Steps/Span	Ave. Iter/Step	Cowell	P-C (Sequential)	P-C (Parallel)†	
1	3 hrs.	2	23	406	3220	46	9:1
2	3 hrs.	2	23	406	3220	46	9:1
3	3 hrs.	4	16	406	4550	65	6:1
4	3 hrs.	2	13	406	1820	26	16:1
5	3 hrs.	4	12	406	3360	48	8:1
6	3 hrs.	4	11	406	3080	44	9:1
7	3 hrs.	2	13	406	1300	26	16:1
8	40 days	1	26	40	1350	26	1.5:1
9	200 days	5	26	200	6600	132	1.5:1

† Assumes the number of parallel data streams in the parallel processor is equal to the number of Chebyshev orthogonality points in the Picard-Chebyshev algorithm

* S_i = [Cowell force evaluations]/[P-C (parallel) force evaluations]

The third column represents the number of integration steps required by the Picard-Chebyshev algorithm to complete the integration span and the fourth column gives the average number of Picard iterations required for convergence of each integration step. The next three columns give the number of force evaluations required by the GTDS-RD Cowell integrator, the sequential execution of the Picard-Chebyshev integrator and the parallel execution of that algorithm. It should be noted that the effective number of force evaluations for the parallel Picard-Chebyshev execution assumes that the number of processing elements in the SIMD hardware architecture is equal to or greater than the number of Chebyshev orthogonality points specified for the integration algorithm. The final column is the integrator speed-up ratio computed by Equation (4.2) for the parallel P-C integrator relative to the high precision Cowell integrator.

The most important aspect of this table is the speed-up ratio column. Table 4.1 indicates that the Picard-Chebyshev integrator, executing on the SIMD target hardware, would perform from 6 to 16 times faster than the Cowell integrator executing on sequential hardware for the high precision equations of motion. The algorithm speed-up ratios for the averaged equations of motion are much

smaller, executing only 1.5 times as fast as the averaged orbit generator in the GTDS-RD system. It is felt that this is due to the previously discussed difficulty in fitting the secular rate in the mean longitude over many satellite revolutions. This limits the integration stepsize in the Picard-Chebyshev method and severely degrades the speed-up ratio. It is felt that this situation could be significantly improved if the averaged equinoctial VOP equations of motion were reformulated in the six slowly varying elements instead of the current formulation of five slowly varying elements and one rapidly rotating phase element. This would remove the large secular rate in the mean mean-longitude due to the mean mean-motion and should allow much larger P-C integration stepsizes and hence better performance relative to the sequential averaged integrator.

Two other interesting characteristics are indicated in Table 4.1. First, comparison between test run 2 (Cowell-drag-5400 second step) and test run 4 (VOP-drag-5400 second step) indicates that the Picard-Chebyshev integrator exhibits significantly better convergence characteristics when integrating the Variation of Parameter type equations of motion. It must be remembered that the zeroth iteration in the Picard-Chebyshev algorithm is taken to be the two-body solution. The next iteration includes the effects of

the selected force model. In the Cowell formulation, the perturbations in the position and velocity components can have relatively large amplitudes and some amount of phase shift. These potentially large differences between the two body solution of the zeroth iteration and the perturbed solution of the first iteration may require additional Picard iterations to converge. On the other hand, the perturbations in the VOP elements are generally of smaller magnitude and therefore speed convergence of the Picard method. A similar characteristic appears to hold for the choice of solve-for vectors in the estimation algorithm.

The second characteristic shown in Table 4.1 concerns the performance penalty incurred when the integration stepsize is reduced. In test run 2, the integration stepsize for the Cowell formulation was 5400 seconds, while test run 3 reduced the stepsize to 2700 seconds. The shorter span allowed the Picard method to converge in 16 iterations/step as compared to the 23 iterations/step exhibited by the longer stepsize of test run 2. However, the two additional steps required to complete the total 3 hour integration span resulted in a total of 65 effective force evaluations for the total integration span and a speed-up ratio of 6:1. In general, this performance penalty will be

incurred anytime the P-C stepsize is reduced because the number of Picard iterations per step does not decrease linearly with the reduction in the stepsize.

Table 4.2 presents similar data for the low altitude elliptical test case. The general characteristics in the data are the same as those discussed for the low altitude circular test case, although the speed-up ratios appear slightly higher. The penalty due to stepsize reduction is again demonstrated. Comparison of test run 2 (Cowell-drag-8100 second step), test run 3 (Cowell-drag-4050 second step) and test run 4 (Cowell-drag-2025 second step) shows that they required 68, 93 and 143 effective force evaluations respectively to complete the 6.75 hour integration span. This caused significant performance penalties, as seen by comparing the resulting speed-up ratios. Due to the point distribution problem discussed in Chapter 2, this penalty may have to be accepted in order to achieve the necessary accuracy for elliptical orbits such as this.

4.3 Orbit Estimator Performance Results

The Parallel Variable Metric (PVM) algorithm of Straeter (139) was adapted for the orbit estimation problem in Chapter 3. The primary parallelism in this algorithm

Table 4.2. Speed-Up Ratio Computation for Low Altitude Elliptical Test Runs

Run #	Int. Span	P-C Algorithm Parameters		Effective Force Evaluations			Speed-Up†† Ratio
		Steps/Span	Ave. Iter/Step	Cowell	P-C (Sequential)	P-C (Parallel)†	
1	6.75 h	3	26	816	5530	79	10:1
2	6.75 h	3	23	816	4760	68	12:1
3	6.75 h	6	16	816	6510	93	9:1
4	6.75 h	12	12	816	10,010	143	6:1
5	6.75 h	3	13	816	2660	38	21:1
6*	6.75 h	3	13	816	2660	38	21:1
7*	6.75 h	6	11	816	4620	66	12:1
8	6.75 h	3	13	816	2660	38	21:1

† Assumes the number of parallel data streams in the parallel processor is equal to the number of Chebyshev orthogonality points in the Picard-Chebyshev algorithm.

†† S_i = [Cowell force evaluations]/[P-C (parallel) force evaluations]

* In these runs, epoch mean anomaly was set to zero degrees.

is the parallel evaluation of the cost functional and its gradient vector in Step 1 of the method. For a p-dimensional solve-for vector, there would be p cost and gradient evaluations which could be performed in parallel. In this study p was limited to 6, so that only the state components are included in the solve-for vector.

The PVM algorithm was tested on two primary test cases. Chapter 3.6 describes the Battin test case and the Adimurthy and Joy test case and presents numerical results indicating the accuracy of the estimation results. Provision was also included in this version of the PVM algorithm to collect certain performance statistics for the algorithm. The following paragraphs summarize this performance data for the various test configurations described in Chapter 3.

Table 4.3 presents the progress of the PVM algorithm at each iteration in the solution of the Battin test case. The three different choices of solve-for vectors are indicated and the values of the cost functional at each iteration are given. The number of cost functional and gradient evaluations for each iteration are given for the sequential execution of the PVM algorithm. Additionally, the effective number of cost and gradient evaluations, taking into

Table 4.3. Progress per Iteration of the PVM Algorithm for The Battin Test Case (Linearized Method/Position-Only Observations)

Iteration	Cartesian Solve-For			Keplerian Solve-For			Equinoctial Solve-For		
	Cost	Function Evals.		Cost	Function Evals.		Cost	Function Evals.	
		Seq.	Par.		Seq.	Par.		Seq.	Par.
0	1.7×10^5	2	2	1.7×10^5	2	2	1.7×10^5	2	2
1	6.4×10^1	22	12	5.7×10^1	22	12	5.3×10^1	22	12
2	4.9×10^1	22	12	5.1×10^1	22	12	4.9×10^1	22	12
3	4.9×10^1	22	12	5.0×10^1	23	13	4.9×10^1	22	12
4	4.9×10^1	22	12	4.9×10^1	24	14			
5				4.9×10^1	23	13			
6				4.9×10^1	23	13			
7				4.9×10^1	23	13			
8				4.9×10^1	23	13			
9				4.9×10^1	23	13			
Total		90	50		208	118		68	38
S_e		1.80:1			1.76:1			1.79:1	

account the algorithm parallelism, are also tabulated. For each candidate solve-for vector the estimation speed-up ratio S_e given by Equation 4.1 is computed.

The results of Table 4.3 show that, for the Battin test case, the equinoctial solve-for vector gives the most rapid convergence of the PVM algorithm, followed by the Cartesian and Keplerian solve-for vectors. As discussed in Chapter 3.6.3, the poor performance of the Keplerian solve-for vector is felt to be due to the ill-defined nature of some of the epoch Keplerian elements for the near-circular, near-equatorial Battin test case. For this case, in which there were relatively small epoch position and velocity errors and large observation errors, the convergence characteristics of the Cartesian solve-for vector are competitive with the equinoctial solve-for vector.

The total number of function (cost and gradient) evaluations required for convergence of the PVM algorithm for both the sequential and parallel executions are given in Table 4.3. These totals include the sequential cost functional evaluations required for the univariate search, as described in Chapter 3.5.2. The estimation speed-up

ratios computed from these totals are relatively small, being slightly less than 2:1 for each choice of solve-for vector. It is felt that the primary cause of these low speed-ups is the low dimension of the solve-for vector. Since the amount of parallel computation in the PVM algorithm is directly related to the dimension of the solve-for vector, the potential speed-up ratios will increase as the dimension of the solve-for vector, and hence the number of parallel processing paths, increases. This could be considered as a potential strength of the algorithm in that its relative performance should improve as the dimension of the solve-for vector and the resulting processing load increases.

Table 4.4 presents similar data for the Adimurthy and Joy test case. It should be recalled that this test case included a very bad *a priori* estimate of the solve-for parameters, but the position-only observational data was relatively precise. In this case, the use of the Cartesian solve-for vector resulted in significantly poorer convergence characteristics for the PVM algorithm compared to either the Keplerian or the equinoctial solve-for vectors. It is felt that this is due in the most part to the scaling of the different solve-for sets. The *a priori*

Table 4.4. Progress per Iteration of the PVM Algorithm for the Adimurthy and Joy Test Case (Linearized Method/No Drag/Position-Only Observations)

Iteration	Cartesian Solve-For			Keplerian Solve-For			Equinoctial Solve-For		
	Cost	Function Evals.		Cost	Function Evals.		Cost	Function Evals.	
		Seq.	Par.		Seq.	Par.		Seq.	Par.
0	3.8×10^{10}	2	2	3.8×10^{10}	2	2	3.8×10^{10}	2	2
1	3.7×10^{11}	23	13	2.3×10^{9}	23	13	5.3×10^{7}	22	12
2	1.6×10^{9}	22	12	9.7×10^{6}	22	12	2.3×10^{3}	22	12
3	1.4×10^{10}	22	12	2.1×10^{3}	22	12	5.9×10^{1}	22	12
4	1.7×10^{7}	22	12	5.9×10^{1}	22	12	5.9×10^{1}	22	12
5	3.5×10^{7}	22	12	5.9×10^{1}	22	12			
6	8.0×10^{5}	22	12						
7	8.7×10^{5}	22	12						
8	4.3×10^{4}	22	12						
9	1.3×10^{4}	22	12						
10	1.7×10^{3}	22	12						
11	2.1×10^{2}	22	12						
12	1.1×10^{2}	22	12						
13	6.0×10^{1}	22	12						
14	6.0×10^{1}	22	12						
15	5.9×10^{1}	22	12						
16	5.9×10^{1}	22	12						
17	5.9×10^{1}	22	12						
18	5.9×10^{1}	22	12						
Total		399	219		113	63		90	50
S_e		1.82:1			1.79:1			1.80:1	

Cartesian solve-for parameters given in Table 3.19 are very much different than their true values given in Table 3.18. In the case of the x component of epoch position, for example, the difference is nearly 1100 km. These large differences cannot be easily removed by the application of the modified metric matrix, as in Equations (3.15) and (3.16). Therefore, removal of these physically large initial position and velocity errors requires more PVM iterations. Additionally, it should be remembered that the convergence rate slows as the algorithm progresses because the components of the search direction vector are getting progressively smaller. This tends to exacerbate the convergence problem. On the other hand, the physical size of the differences in the *a priori* and truth ephemeris epoch elements is much smaller. These small element differences can represent large differences in epoch position and velocity, as shown in Tables 3.18 and 3.19. However, these physically smaller initial element differences are much more easily removed by the PVM search direction vector than the larger position and velocity differences. Additionally, as the algorithm nears convergence, the relatively small components of the search direction vector still can exert a significant effect on the solve-for elements and therefore on the cost and gradient values.

The estimator speed-up ratios listed in Table 4.4 are still approximately 2:1 as in the Battin test case. Again this is due in part to the low dimension of the solve-for vector. A verification of this conjecture can be found in Reference 139. In his numerical results section, Straeter exercises the PVM algorithm on several different test functions. The solve-for vectors in these test cases are of order 2, 3 and 4. He also computes the total number of cost and gradient evaluations in sequential and parallel executions of the PVM algorithm, but his figures ignore those cost evaluations required in the univariate search. Based on his numbers, the speed-up ratios for the solve-for vectors of order 2, 3 and 4 would be 1.3:1, 1.6:1 and 2:1 respectively. If the univariate search is ignored in Table 4.4, then the estimator speed-up ratio for the 6 dimensional state solve-for vector is approximately 3:1. This is generally consistent with the ratios from Straeter and indicates that the relative algorithm performance improves as the solve-for vector dimension increases.

The above discussion also points out that the univariate search has a significant impact on the performance of the PVM estimation algorithm. Table 4.5 demonstrates this effect. This table presents the effective number of cost

Table 4.5. Effect of Sequential Univariate Search on the Parallel Variable Metric Orbit Estimation Algorithm

Test Case	Solve-For Vector	Computation Method†	Function Evaluations				Speed-Up Ratio (S_e)	
			With 1-D Search		Without 1-D Search		With 1-D Search	Without 1-D Search
			Seq.	Par.	Seq.	Par.		
BATTIN	CARTESIAN	DIR (19.305)	119	69	72	22	1.82:1	3.27:1
	CARTESIAN	LIN (2.550)	90	50	58	18	1.80:1	3.22:1
	KEPLERIAN	LIN (5.137)	208	118	128	28	1.76:1	3.37:1
	EQUINOCTIAL	LIN (1.945)	68	38	44	14	1.79:1	3.14:1
ADIMURTHY AND JOY (NO DRAG)	CARTESIAN	LIN (10.483)	399	219	254	74	1.82:1	3.43:1
	KEPLERIAN	LIN (3.320)	113	63	72	22	1.79:1	3.27:1
	EQUINOCTIAL	DIR (23.615)	141	81	86	26	1.74:1	3.31:1
	EQUINOCTIAL	LIN (2.664)	90	50	58	18	1.80:1	3.22:1
$\Delta M = 90°$	EQUINOCTIAL	LIN (4.389)	164	94	100	30	1.74:1	3.33:1
$\Delta M = 120°$	EQUINOCTIAL	LIN (4.298)	156	86	100	30	1.81:1	3.33:1
$\Delta M = 150°$	EQUINOCTIAL	LIN (5.591)	200	110	128	38	1.82:1	3.37:1
ADIMURTHY AND JOY (DRAG)	KEPLERIAN	LIN (3.568)	135	75	86	26	1.80:1	3.31:1
	EQUINOCTIAL	LIN (2.440)	90	50	58	18	1.80:1	3.22:1

† DIR = Direct Integration Method for evaluation of cost and gradient
LIN = Linearization Method for evaluation of cost and gradient

Note: The number in parentheses is the Amdahl 470V6 execution time in minutes required for the sequential execution of the PVM algorithm to converge for the given test case.

and gradient evaluations for each test case both including and excluding the univariate search. It is seen that the exclusion of the sequential cost functional evaluations required by the univariate search results in the speed-up ratios improving from about 1.8:1 to 3.4:1. This indicates that additional parallelism should be introduced into the PVM algorithm in the univariate search to further improve performance.

These performance discussions of the estimation algorithm have essentially ignored the fact that the parallel orbit integration algorithm is intimately intertwined with the PVM estimation algorithm. The P-C integrator represents the deterministic model of the satellite motion required by the estimation algorithm for computation of the observation residuals. The significant improvements in the integrator performance discussed in Chapter 4.2 due to the algorithm parallelism would provide further overall performance improvements. It appears that an orbit determination system based on the algorithms given in Chapters 2 and 3, in conjunction with the target SIMD hardware described in Chapter 1, can provide accurate numerical results with significant improvements in system performance.

Chapter 5

CONCLUSIONS AND FUTURE WORK

This thesis has applied the rapidly advancing state-of-the-art in computer hardware, and specifically in parallel processor architectures, to the computational problem of orbit determination for artificial satellites. The primary motivation for this effort has been the realization that current operational ground support orbit determination capabilities will soon be inadequate to meet expanding requirements. The needs for more accurate orbital information, available to the user in a near real-time environment, and the growing satellite population requiring such support, all have combined to stress severely, and in some cases totally saturate, current support capabilities.

5.1 Summary

A primary concern of operational orbit determination systems is throughput capability, that is, their ability to provide timely orbital information at some acceptable accuracy level for a given number of satellite "customers." New uses of the space environment are developing in both

the civil and the military sectors. These new missions lead to increasing ground support requirements in accuracy, timelines and numbers of users. The throughput capabilities of many of the support systems of today are inadequate to meet these future requirements.

In light of this, certain efforts are currently underway to increase the throughput capabilities of present day support systems. In some cases, these efforts take the form of procurement of new, large sequential computing equipments to replace older, slower computers. In other cases, new orbit determination algorithms for existing sequential computers are being formulated. Such new algorithms may include improved General Perturbation satellite theories with expanded force models, since GP theories are often computationally faster than Special Perturbation techniques. Other algorithmic approaches of importance include the semianalytical satellite theory (49, 82, 110). This approach relies on the Generalized Method of Averaging, such that numerical integration is used only on averaged dynamical equations so large integration steps may be taken. This provides significant computational efficiency. Recovery of the short periodic components of the motion through analytical or numerical means provides the necessary accuracies.

The above algorithmic efforts have been developed under the assumption that the target computer hardware architectures were the large scale sequential architectures presently employed in these facilities. This thesis makes an entirely different assumption concerning future hardware architectures. It is assumed in this work that multiprocessor hardware system architectures will be strong candidates for these orbital support applications because of their potential to provide increased effective computing power at lower costs. The specific multiprocessor hardware architecture used as a target machine for this thesis was the Single Instruction Stream/Multiple Data Stream (SIMD) array processor described and evaluated by Flynn (78, 79).

In this target hardware architecture, as pictured in Figure 1.1, a central control unit issues the same instruction to numerous identical slave processing elements, each of which executes that instruction on its own stream of data. For the case of P identical processing elements, this architecture allows P executions of the issued instruction in the effective time required for one execution of that instruction. On a larger scale, if a task can be broken into P identical code segments, then the task can

be executed in parallel in the effective time required to execute one of these segments sequentially. This would result ideally in a parallel execution time that was P times faster than the sequential execution of the same task. Clearly, the potential for improved performance and throughput is great. However, in order to realize this potential extreme care must be taken in the formulation of the numerical algorithms to be executed. These algorithms must be specifically tailored to contain large blocks of identical code segments in order to take maximum advantage of the available hardware parallelism. The formulation and evaluation of two such algorithms for the orbit determination problem have been the primary thrusts of this thesis.

To provide a credible orbit determination capability, the parallel processor system must perform two basic functions: orbit propagation and orbit estimation. Parallel algorithms for both these functions have been formulated in this thesis. The primary features of these algorithms are a high degree of parallelism, such that significant performance improvement is possible, and high accuracies comparable to current high precision sequential methods.

A Special Perturbation orbit propagation algorithm was developed to integrate numerically the satellite dynamical equations of motion. The Special Perturbation approach was chosen because of the desire to exhibit high accuracies for many different perturbation sources. The algorithm has been applied to the Cowell, Variation of Parameters (VOP) and averaged Variation of Parameters formulations of the satellite equations of motion. The algorithm accuracy has been baselined against high precision numerical integrations and its potential performance in a parallel environment has been evaluated.

The orbit propagation algorithm combines the techniques of Picard iteration and Chebyshev polynomial approximation. The Picard-Chebyshev algorithm approximates over a given time span the right-hand sides of the equations of motion with series of Chebyshev polynomials. Using the properties of Chebyshev polynomials, the series approximations for the integrands are directly integrated to form series approximations for the integration state components. For the Cowell equations, the Chebyshev series for the total acceleration components over some time span (nominally one satellite revolution) are integrated twice, once to determine the series coefficients for the velocity

component series and once again to compute the coefficients for the position component Chebyshev series. In the high precision and averaged VOP formulations only one such integration is required for the six orbital element equations. Following the method of Picard, the results of the current iteration are compared to the previous iteration and, if the differences are less than some prescribed criterion, convergence is assumed and processing stops. The resulting series for the integration state components (position and velocity or orbital elements) can then be evaluated at any point within that integration arc by a simple nested recurrence algorithm. If convergence is not achieved, the results of the current iteration are used as input for the right-hand side evaluations required in the next iteration and computation continues.

The primary parallelism in this algorithm is found in the evaluation of the coefficients for the Chebyshev series approximating the right-hand sides of the equations of motion. This computation requires the evaluation of the total acceleration vector or the element rates (and hence the integration force model) at a set of N + 1 predetermined orthogonality points contained in the current integration step. These computations are identical, the only

difference being the particular orthogonality point in the span for which the evaluation must be accomplished. This makes these N + 1 force evaluations ideal candidates for parallel processing on an SIMD array processor. Under this scheme, the N + 1 force evaluations required in a single Picard iteration can be accomplished in the effective time required to perform a single force evaluation in a sequential integrator. Thus, when the Picard iterative method converges in a reasonable number of iterations, the performance of the parallel Picard-Chebyshev integration algorithm should be much superior to current sequential high precision integrators.

The accuracy of the Picard-Chebyshev algorithm was displayed by comparisons with an analytical solution of the two body problem and with high precision numerical integrations of the perturbed Cowell equations of motion by the GTDS-RD system. The general numerical behavior of the algorithm was found to be quite similar to that of the GTDS-RD high precision integrator. The integration results provided by the Picard-Chebyshev algorithm for low altitude circular orbits agreed very well with the GTDS-RD integrations. In the case of low altitude elliptical orbits with drag perturbations it was found that, for

certain initial conditions, the distribution of the Chebyshev orthogonality ponts did not allow accurate approximation of the drag acceleration over arcs of one satellite revolution. The integration accuracies were greatly improved by reducing the length of the approximation arcs. For these orbital geometries and force models, it appears that the Picard-Chebyshev integration stepsize might be somewhat limited and this impacts the potential performance improvement.

To accomplish the orbit estimation task, the Parallel Variable Metric (PVM) algorithm of Straeter (139) was selected. This algorithm was adopted for the orbit estimation task because it was specifically designed to be used on computer hardware such as that assumed for the target hardware architecture in this thesis. The algorithm is a gradient dependent, nonlinear function minimization technique. The function to be minimized is taken to be the sum of the weighted squares of the observation residuals, much like the classical method of differential corrections. The solve-for parameters are selectable by the user and can include the epoch position and velocity components, the epoch Keplerian elements or the epoch equinoctial elements. The implementation described in this work does not allow

for other parameters, such as drag or atmospheric data, to be included in the solve-for vector.

The computation of the gradient of the cost functional with respect to the solve-for parameters requires calculation of the observation partial derivatives. Therefore, analytical state partial derivatives were formulated in terms of the equinoctial orbital elements. These state partials take into account the secular perturbations in the longitude of the ascending node and the argument of periapsis due to the dominant zonal harmonic in the gravitational potential.

Computation of the observation residuals for the evaluation of the cost functional and its gradient requires a deterministic model of the satellite motion to compute predicted observables based on the current estimate of the solve-for parameter. The Picard-Chebyshev parallel orbit propagation algorithm serves that purpose in the implementation. However, this requires a numerical integration of the perturbed equations of motion for each cost and/or gradient evaluation and is computationally expensive even in the parallel mode. Therefore, a linearized method for determining the necessary observation residuals was formulated. For a p-dimensional solve-for vector, the first

step in the PVM method requires p parallel evaluations of the cost and gradient at p new values of the solve-for vector. These new values of the solve-for vector are computed by slightly perturbing the current "best estimate" of the solve-for vector along the p directions defined by the basis vectors spanning the solve-for parameter space. In the direct integration approach, the cost and gradient associated with each of these new solve-for vectors would require a numerical integration of the equations of motion. In the linearization method, only one numerical integration based on the current "best estimate" of the solve-for vector is made. This establishes the reference trajectory for the linearization. The p perturbations in the solve-for parameters represent perturbations in the epoch state. These perturbations are propagated into perturbations from the reference trajectory at the observation times by use of the state transition matrix. This allows the p numerical integrations to be replaced by vector-matrix operations and results in a reduction of running time by nearly 10:1 in sequential simulations.

The accuracy of the estimation results produced by the PVM algorithm was compared to orbit estimation results appearing in recent literature. The circular equatorial

test case of Battin (30) and the more stringent test case of Adimurthy and Joy (4) were used. In both cases, the PVM method appeared to produce estimates of the state solve-for parameters that agreed quite well with the comparison estimation methods. Additionally, it was found that the PVM algorithm was exceptionally robust in its radius of convergence. This was expected due to the work of Adimurthy and Joy in the use of sequential nonlinear optimization methods for orbit determination.

Estimates were made of the performance of the orbit propagation and orbit estimation algorithms in a parallel processor environment. For the orbit propagator, the performance was estimated by examining the ratio of the number of force evaluations required for a GTDS-RD high precision numerical integration of a given span to the effective number of force evaluations for the parallel Picard-Chebyshev integration of the same span. For the test cases run, this integrator speed-up ratio ranged from 1.5:1 to 21:1. For the orbit estimator, a similar performance measure was formed as the ratio of the number of cost and gradient evaluations required for a sequential execution of the PVM algorithm to the effective number of cost and gradient evaluations required for a parallel execution of the PVM

algorithm. When the cost function evaluations required for the univariate search are considered, the estimator speed-up ratio is about 1.8:1. If these sequential evaluations are not included, the speed-up ratio increases by nearly a factor of 2.

5.2 Conclusions

The primary conclusion drawn from this work is that the orbit determination problem can be structured to contain a large amount of parallelism. This parallelism could be used, in conjunction with a parallel processor hardware architecture, to perform orbit determination support functions much faster than current high precision methods.

It is concluded that the integration accuracies provided by the Picard-Chebyshev parallel orbit propagation algorithm are comparable to the accuracies available from current high precision numerical integrators, such as those contained in the GTDS-RD test bed. However, due to the difficulty in approximating with Chebyshev series the drag acceleration near the perigee of a highly eccentric orbit, it must also be concluded that the Picard-Chebyshev integration stepsize may be quite limited for such an orbital geometry to achieve the necessary accuracy, resulting in a

reduced performance improvement. The performance of the Picard-Chebyshev algorithm on the target SIMD hardware architecture would be far superior to that exhibited by any present sequential numerical integrator.

Similarly, it is concluded that the Parallel Variable Metric orbit estimation algorithm provides orbital state estimates at accuracy levels competitive with classical differential correction methods, linear filter methods and many nonlinear function minimization techniques. It is felt that expansion of the allowed solve-for vector to include atmospheric density model parameters and spacecraft ballistic information would result in more accurate estimation results for low altitude drag perturbed satellites.

The PVM algorithm also has a significant advantage in terms of its radius of convergence. It has been shown that the PVM orbit estimation algorithm exhibits a much wider radius of convergence with respect to the quality of the initial guess of the solve-for parameters for which convergence can be obtained. This characteristic makes the PVM algorithm extremely attractive for use in orbit determination of non-cooperative satellites, early orbit determination or any other application where the *a priori* state vector may be suspect.

It is concluded that in general the use of epoch orbital elements as the solve-for parameters is superior to the use of the epoch position and velocity components in the gradient-dependent nonlinear function minimization algorithm given here. The convergence results of Chapter 3 for the Adimurthy and Joy test case display this quite well. In most cases, use of the epoch Keplerian or epoch equinoctial elements as solve-for parameters yields similar convergence results. However, using the Keplerian elements as the solve-for parameters for orbital geometries for which they are poorly defined results in unnecessary PVM iterations, as shown by the Battin test case results.

The performance improvement displayed by the PVM algorithm is not as dramatic as that of the parallel integration algorithm. However, the nearly 2:1 speed-up ratio due to the parallelism of the PVM algorithm itself, when coupled with the additional high performance of the linearized method of cost function evaluation and the parallel integration algorithm, would result in a much higher effective system performance. It is therefore concluded that the overall performance of a parallel processor orbit determination system in the orbit estimation problem would far exceed that of current batch weighted least

squares differential correction algorithms. The performance should be much better than sequential nonlinear function minimization techniques.

Based on the results in this work, it is concluded that the combination of parallel hardware architectures and parallel algorithms can be applied to the problem of orbit determination for artificial satellites. The resulting system will exhibit accuracies comparable to current high precision methods. The performance of this parallel system will be much better than its current sequential counterparts. Such a system could alleviate much of the throughput burdens plaguing present day orbit determination support facilities.

5.3 Future Work

Several possible areas of future work have been stimulated by this study. They include further work on the particular algorithms presented here and also a new philosophical approach to the overall problem.

The Picard-Chebyshev integration algorithm is relatively complete for integrating the basic equations of motion. However, in many high precision orbit determination appli-

cations, it is required to integrate the variational equations accurately. The Picard-Chebyshev algorithm should be expanded to include this capability. This would require a significant modification to the integrator in order to accommodate series coefficients for the right-hand sides of the variational equations and the integrated results. The basic logic flow of the program would not require modification.

The numerical integration results for the low altitude elliptical test case indicated that the drag acceleration in the perigee region was sometimes difficult to represent accurately by Chebyshev series. This was due to the sparse distribution of Chebyshev orthogonality points in the drag region for certain epoch integration states. This required decreasing the Picard-Chebyshev integration stepsize and resulted in a performance penalty. Further research needs to be done to alleviate this problem for these orbital geometries. A candidate solution is to approximate the drag acceleration by Chebyshev series only within the sensible atmosphere. This approach is much the same as that used in the GTDS-RD drag perturbation model for the averaged equations of motion (104). In that approach, the abscissae of the numerical quadrature necessary

to average numerically the drag perturbations are confined to the drag regime only. It is possible that a similar technique for Chebyshev orthogonality points used to determine the drag acceleration series coefficients could be useful.

Three particular areas of future study are important for the Parallel Variable Metric algorithm. First, a radar observation model should be included in the software. In the current study, the observations were assumed to be components of the inertial position and velocity vectors and therefore linear functions of these vectors. For radar observations, however, the observed quantities are nonlinear functions of the inertial position and velocity vectors at the observation times. It is important to determine the effects this nonlinear observation model may have on the accuracy and performance of the PVM algorithm.

A second area of interest is the necessary work to expand the PVM method to accept more solve-for parameters than merely the epoch state components. Of particular interest are drag-related solve-for parameters, such as the ballistic coefficient, and atmospheric density model parameters. Besides expansion of the existing PVM program

structures, this would require the previously mentioned inclusion of the variational equations in the Picard-Chebyshev integrator. It is felt that expansion of the solve-for vector will provide a more accurate estimation capability and would most certainly be required in an operational environment.

Finally, the performance results of Chapter 4 indicate that the sequential cost function evaluations required in the PVM algorithm univariate search are a limiting factor in the PVM algorithm performance. The Coggins method was chosen for this role because of its simplicity and adequate consideration of the search algorithm parallelism was not taken. For these reasons, work should be undertaken to replace the current sequential univariate search algorithm with a parallel version of this type of algorithm. Several candidate algorithms are presented in Reference 115. Additionally, further study should be made to consider the potential improvements in PVM algorithm performance possible if the more rapidly convergent VOP equations of motion are used in the required numerical integration.

On a system level, two areas of future work can be defined. The performance evaluation of the individual

algorithms done in this thesis was somewhat idealized in that it neglected certain overhead factors inherent in parallel processor hardware architectures. It is felt that an expanded performance evaluation should be performed on the overall parallel processor orbit determination system with the specific intent of determining more accurate performance figures and of identifying potential areas for further parallel processing. Included in this evaluation should be consideration of input/output problems, vector matching and data movement and alignment penalties.

The parallelism approach used in this thesis is closely allied to another concept called overlap, also known as vector streaming or pipelining. Currently, several small array processors are available which are expressly designed to take advantage of pipelined computations. For example, Bucy and Senne (42) recently reported a peripheral array processor such as this provided very low cost for its performance in nonlinear filtering applications. It is felt that the orbit determination algorithms developed in this work could be easily adapted to operate in a pipelined mode on a peripheral array processor. Performing this task would give considerable insight into the ultimate performance of such a system and also could provide a

prototype system for use in tests at operational ground support facilities. Ultimately, this effort should lead to implementation of these or similar algorithms on fully parallel hardware architectures when available or on a current state-of-the-art vector processor (for example, CRAY-1 or CDC STAR-100).

An entirely different philosophy for attacking the fundamental problem also exists. In this thesis, parallelism was induced in the algorithms by creating large identical segments of code. These segments were then executed in parallel by a tightly controlled, synchronous array of slave processing elements. The parallel execution of these code segments results in performance improvements. Conversations with Dr. Andre Deprit have led to another philosophical approach in which parallelism is induced at the task level in the system. Independent tasks would be executed simultaneously on loosely federated, asynchronous processors. Cooperating tasks would be capable of interprocessor communication, such that information from one processor could be passed to the other tasks requiring it. The allocation of tasks for the orbit determination problem, the hardware requirements and the communication protocols have not been determined, but this approach repre-

sents a flexible method of taking advantage of parallelism in the orbit determination problem.

The work in this thesis has pointed out a related topic that deserves future study. Chapters 3 and 4 have shown that the PVM algorithm coupled with the linearization method for the evaluation of the cost functional and gradient vectors gives excellent performance results. The use of numerical integration in the PVM method allows the algorithm to be applied to problems such as the orbit estimation problem, whose basic dynamics are characterized by differential equations with no closed-form solutions. The linearization about the current best estimate of the solve-for vector significantly reduces the number of integrations required by the PVM method and therefore provides marked speed improvement. This speed improvement has been demonstrated in sequential simulations, even using the Picard-Chebyshev integrator which is extremely inefficient in sequential execution. These results indicate that work should be undertaken to include an efficient sequential integration method in the PVM algorithm. It is felt that this combination of PVM algorithm, efficient sequential integrator and linearization method for cost and gradient evaluations could deliver superior performance in a sequential mode. This potential should be investigated.

As a final note, it is becoming increasingly important for the designers of satellite perturbation theories to be concerned with the overall performance of their algorithms. These designers should begin now to give consideration to how well their perturbation theory designs can use the advantages of new, more powerful computer hardware architectures. Current satellite theory design should attempt to accommodate new hardware architectures. Future designs must take full advantage of these architectures. Judicious merging of the satellite theory technologies with new hardware architectures shows the potential to satisfy future orbit determination system requirements.

APPENDIX A

PERTINENT ORBITAL ELEMENT SETS

The Picard-Chebyshev parallel orbit propagation algorithm presented in Chapter 2 was used to integrate the Variation of Parameters (VOP) equations of motion for both the Keplerian and the equinoctial element sets. A brief discussion of the definitions and characteristics of these two element sets is given here. The classical Keplerian elements are well known and extensive discussions of them may be found in numerous texts on Celestial Mechanics (see for example References 28, 60 and 124. Further discussions on the properties of the equinoctial elements may be found in Broucke and Cefola (38), McClain (110), Green (82) and others.

A.1 Keplerian Elements

The solution of the two-body equation of motion

$$\frac{d^2\vec{r}}{dt^2} + \frac{\mu}{r^3}\vec{r} = 0 \qquad \text{(A.1)}$$

yields six constants of integration. These constants of integration are called the elements of the orbit. The unperturbed Keplerian motion takes place in a plane and this motion describes a conic section (see Figure A.1). The shape of the conic section is fully described by the semimajor axis (a) and the eccentricity (e). The magnitude of the position vector in this conic section is given by

$$r = \frac{p}{1 + e \cos f} \tag{A.2}$$

where f = true anomaly

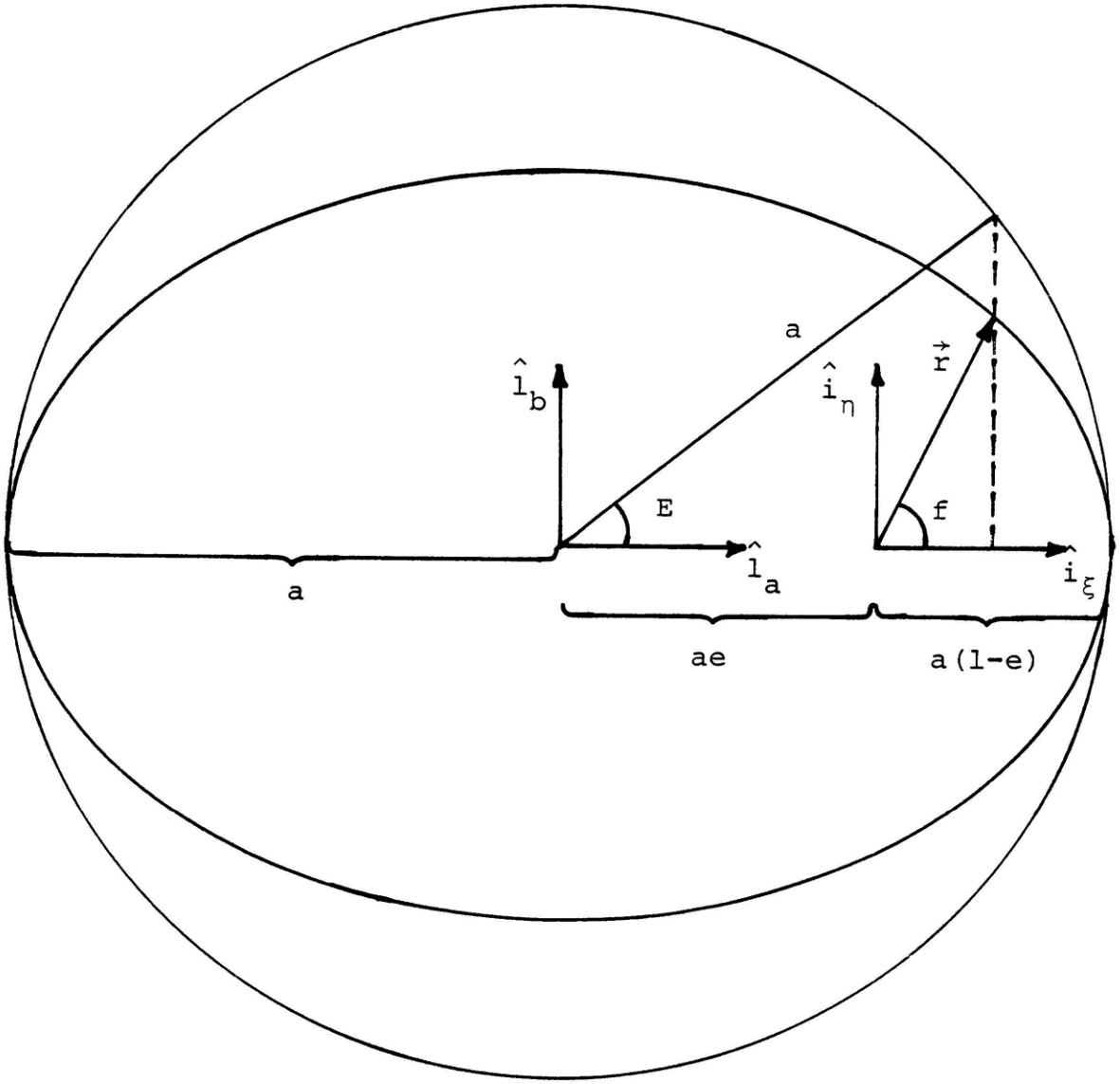
p = semilatus rectum $= \frac{h^2}{\mu}$

$= a(1 - e^2)$

μ = gravitational constant of the central body

h = magnitude of the angular momentum

The relative position in the conic orbit is fully defined by determining the true anomaly (f) as a function of time, where f is referenced to the constant Laplace vector pointing in the direction of pericenter. The angular



$$\hat{i}_\zeta = \hat{i}_\xi x \hat{i}_\eta \text{ (out of page)}$$

Figure A.1. Keplerian Elliptic Orbit in the Plane ($0 \leq e < 1$)

momentum vector $\vec{h}$ is one of the constant integrals of the two body motions, given by

$$\vec{h} = \vec{r} \times \frac{d\vec{r}}{dt} \tag{A.3}$$

The velocity can be written in radial and transverse components, such that

$$\frac{d\vec{r}}{dt} = \dot{r}\, \hat{l}_r + r \frac{df}{dt} \hat{l}_f \tag{A.4}$$

Thus, the magnitude of the angular momentum vector is seen to be

$$h = r^2 \frac{df}{dt} \tag{A.5}$$

By substituting Equation (A.2) into Equation (A.5), the following differential relation for f results (28)

$$\sqrt{\frac{\mu}{p^3}}\, dt = \frac{df}{(1 + e \cos f)^2} \tag{A.6}$$

Integration of this equation determines the time dependence of the true anomaly and also yields a constant of integration called the time of pericenter passage (τ). It is

often more convenient to use as this element a fictitious anomaly called the mean anomaly (M) defined by

$$M = n(t - \tau) \tag{A.7}$$

where n = mean motion = $\mu^{1/2} a^{-3/2}$

With M defined as above, the eccentric anomaly E can be determined by solving Kepler's equation

$$E - e \sin E = M \tag{A.8}$$

Several methods exist for solving this implicit equation, including successive substitution and series expansions. Once the eccentric anomaly E is determined, the true anomaly can be found through the relations

$$\cos f = \frac{\cos E - e}{1 - e \cos E} \tag{A.9a}$$

$$\sin f = \frac{\sqrt{1 - e^2} \sin E}{1 - e \cos E} \tag{A.9b}$$

or more simply

$$\tan \frac{f}{2} = \sqrt{\frac{1 + e}{1 - e}} \tan \frac{E}{2} \tag{A.9c}$$

In this work the semimajor axis (a) and eccentricity (e) are taken to be the orbital elements that describe the size and shape of the Keplerian conic orbit. The mean anomaly (M) locates the satellite position in the Keplerian conic through the solutions to Equations (A.8) and (A.9c).

It now remains to orient the orbit in space. As seen in Figure A.2, an inertial reference frame is established, centered at the center of mass of the central body and with its x axis pointing in the direction of the vernal equinox. In order to spatially orient the $\hat{i}_\xi$, $\hat{i}_\eta$, $\hat{i}_\zeta$ orbit plane coordinates with respect to the inertial frame, three Euler angle rotations are used. The first rotation is about the inertial z axis through an angle $+\Omega$, called the longitude of the ascending node. This rotation results in a rotation matrix of the form

$$R_\Omega = \begin{pmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The second rotation is about the displaced x axis through the inclination angle (+i). This rotation results in a second rotation matrix such that

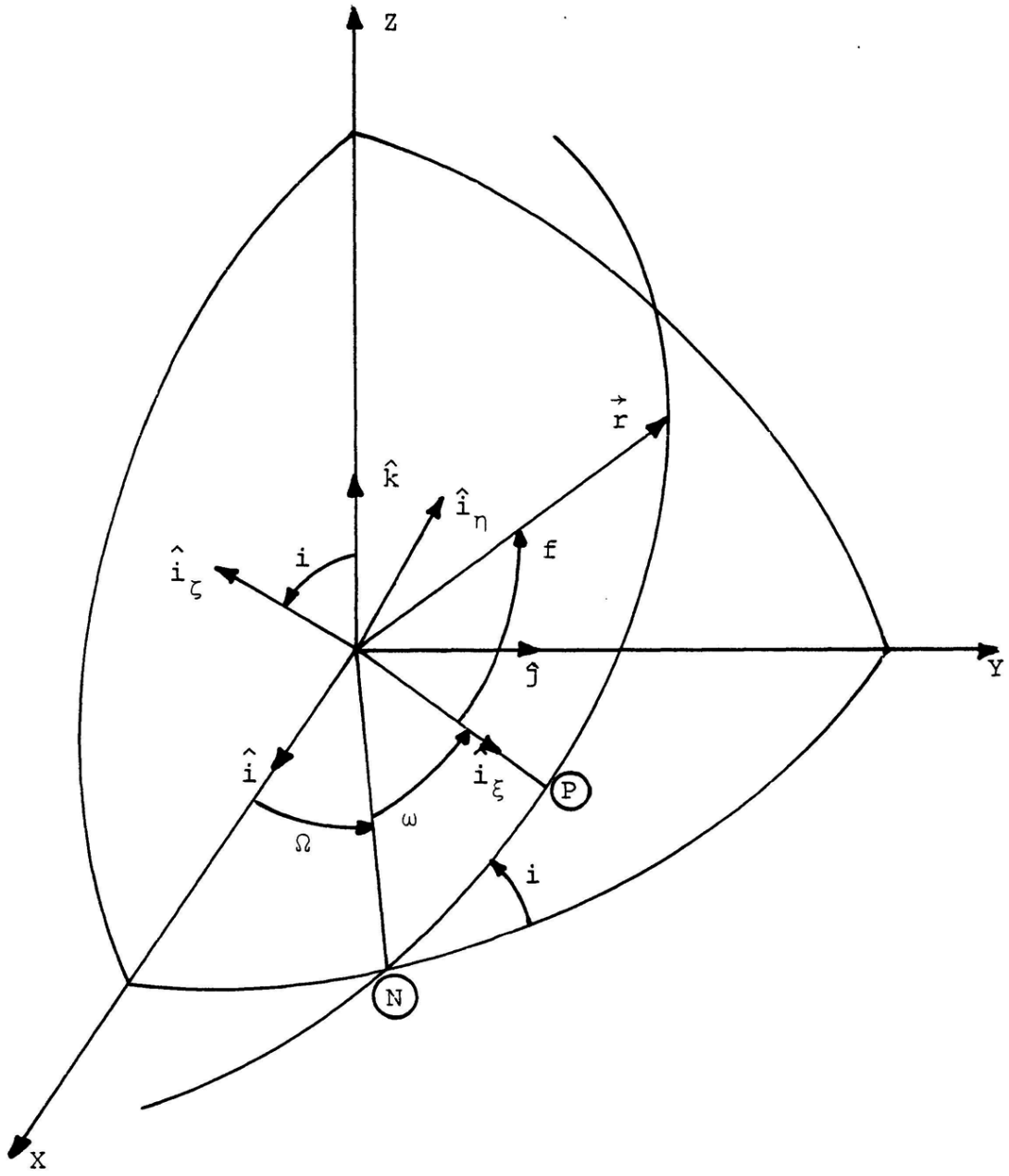


Figure A.2. Spatial Orientation of the Keplerian Orbit

$$R_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{pmatrix}$$

Finally, the third rotation is made about the displaced z axis through an angle $+\omega$, called the argument of periapsis. The resulting rotation matrix is given by

$$R_\omega = \begin{pmatrix} \cos \omega & -\sin \omega & 0 \\ \sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The overall rotation matrix can then be found as the product of these three rotation matrices, such that

$$R = R_\Omega R_i R_\omega$$

$$R = \begin{bmatrix} c_\Omega c_\omega - s_\Omega s_\omega c_i & -c_\Omega s_\omega - s_\Omega c_\omega c_i & s_\Omega s_i \\ s_\Omega c_\omega + c_\Omega s_\omega c_i & -s_\Omega s_\omega + c_\Omega c_\omega c_i & -c_\Omega s_i \\ s_\omega s_i & c_\omega s_i & c_i \end{bmatrix} \qquad \text{(A.10)}$$

where $c_\alpha \triangleq \cos(\alpha)$

$s_\alpha \triangleq \sin(\alpha)$

Thus the three Euler angles Ω, i and ω serve to orient the Keplerian orbit in space.

From Figure A.1, the position vector can be written in the orbit plane coordinate frame as a function of true anomaly

$$\vec{r}_{OP} = r \cos f \, \hat{i}_{\xi} + r \sin f \, \hat{i}_{\eta} \qquad (A.11)$$

where r is found from Equation (A.2). It can also be written in terms of the eccentric anomaly as

$$\vec{r}_{OP} = a(\cos E - e) \, \hat{i}_{\xi} + a\sqrt{1 - e^2} \sin E \, \hat{i}_{\eta} \qquad (A.12)$$

Differentiating this yields the velocity expressed in the orbit plane coordinate frame

$$\dot{\vec{r}}_{OP} = \frac{na^2}{r} (-\sin E \, \hat{i}_{\xi} + \sqrt{1 - e^2} \cos E \, \hat{i}_{\eta}) \qquad (A.13)$$

where r is found by

$$r = a(1 - e \cos E) \tag{A.14}$$

These vectors can then be transformed into the inertial coordinate frame by using the rotation matrix of Equation (A.10), such that

$$\vec{r} = R \begin{pmatrix} a(\cos E - e) \\ a\sqrt{1 - e^2} \sin E \\ 0 \end{pmatrix} \tag{A.15a}$$

$$\dot{\vec{r}} = R \begin{pmatrix} -\frac{na^2}{r} \sin E \\ \frac{na^2}{r} \sqrt{1 - e^2} \cos E \\ 0 \end{pmatrix}$$

Note that the elements of the rotation matrix R are only functions of i, ω, Ω while the components of the orbit plane position and velocity vectors are only functions of a, e and M (implicitly through E). If it is desired to compute state partial derivatives, this is a convenient form of the position and velocity vectors for that purpose.

The Keplerian orbit element set used in this thesis is therefore given by

$$
\begin{aligned}
a &= \text{semimajor axis} \\
e &= \text{eccentricity} \\
i &= \text{inclination} \\
\omega &= \text{argument of pericenter} \\
\Omega &= \text{longitude of the ascending node} \\
M &= \text{mean anomaly}
\end{aligned}
\tag{A.16}
$$

A.2 Equinoctial Elements

As shown in References 82 or 110, the equinoctial elements are defined in terms of the classical Keplerian elements above as

$$
\begin{aligned}
a &= a \\
h &= e \sin(\omega + I\Omega) \\
k &= e \cos(\omega + I\Omega) \\
p &= \tan^{I}\left(\frac{i}{2}\right) \sin\Omega \\
q &= \tan^{I}\left(\frac{i}{2}\right) \cos\Omega \\
\lambda &= M + \omega + I\Omega
\end{aligned}
\tag{A.17}
$$

where I = retrograde factor

$= 1$ for $0^\circ \leq i \leq 90^\circ$

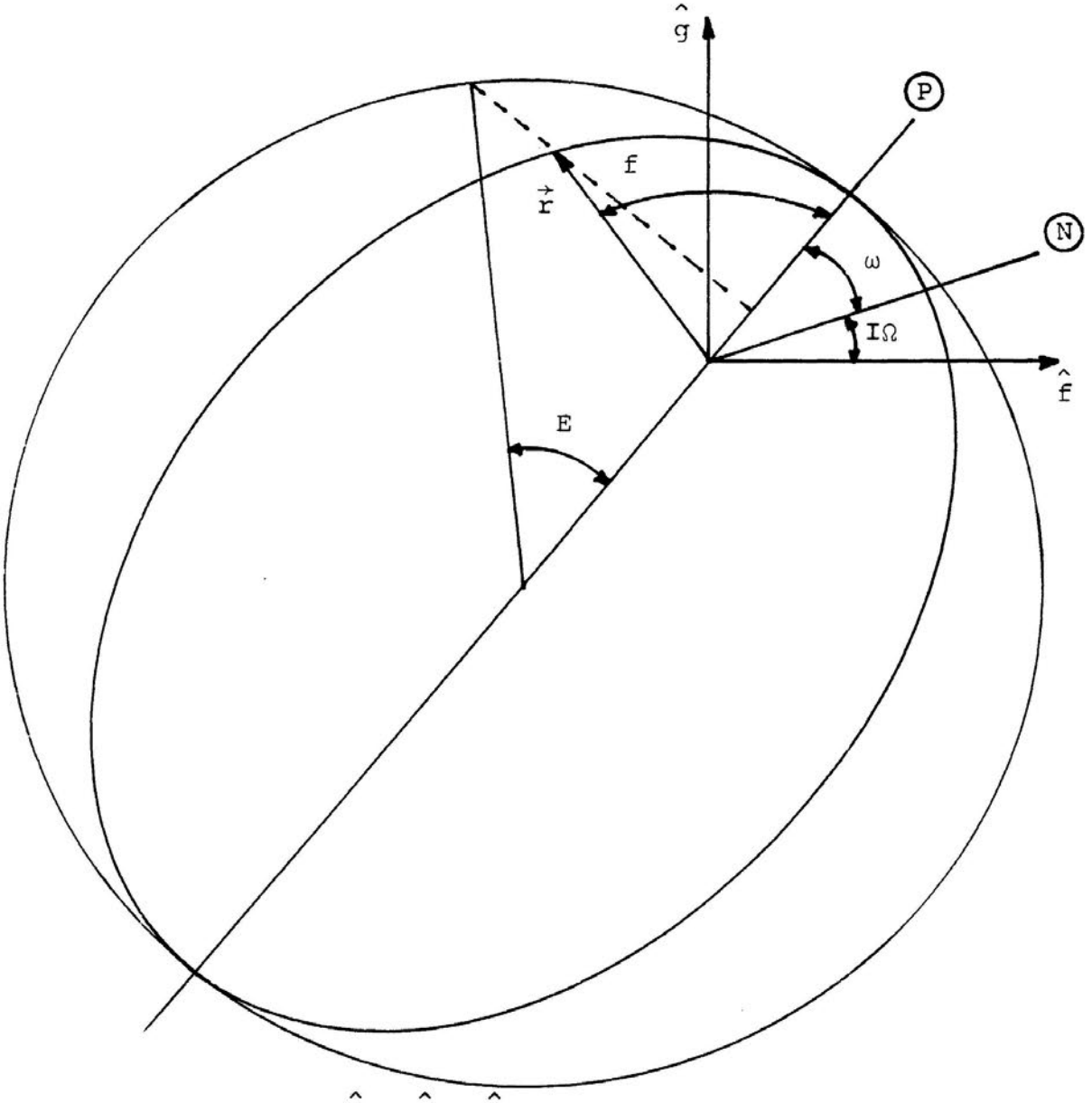
$= -1$ for $90^\circ < i \leq 180^\circ$

This choice of the retrograde factor is arbitrary, but insures that no singularities exist in the element definitions or in the VOP equations of motion.

The equinoctial reference frame is a right-handed orthogonal system defined by unit vectors $\hat{f}$, $\hat{g}$ and $\hat{w}$. Unit vectors $\hat{f}$ and $\hat{g}$ lie in the orbit plane, with $\hat{f}$ directed toward a point in the orbit displaced from the ascending node by the angle $-\Omega$ for the direct element set and $+\Omega$ for the retrograde element set. Unit vector $\hat{w}$ points in the direction of the orbital angular momentum vector and $\hat{g}$ is defined by

$$\hat{g} = \hat{w} \times \hat{f}$$

Figure A.3 shows the equinoctial reference frame for the direct element set. In order to spatially orient the equinoctial reference frame with respect to an inertial reference frame three Euler angle rotations are again employed. Beginning with the equinoctial reference frame aligned with the inertial frame, the first rotation is about the inertial z axis through the longitude of the ascending node Ω. The rotation matrix for this rotation is



$\hat{w} = \hat{f} \times \hat{g}$ (out of page)

L = True Longitude = $f + \omega + I\Omega$

Figure A.3. Equinoctial Reference Frame for the Direct Equinoctial Set in the Plane

$$R_{\Omega} = \begin{pmatrix} \cos\ \Omega & -\sin\ \Omega & 0 \\ \sin\ \Omega & \cos\ \Omega & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The second rotation is about the displaced x axis through the inclination angle i and the resulting rotation matrix is

$$R_{i} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\ i & -\sin\ i \\ 0 & \sin\ i & \cos\ i \end{pmatrix}$$

The final rotation is made about the displaced z axis through an angle $-\Omega$ for the direct element set or $+\Omega$ for the retrograde element set. The associated rotation matrix is given by

$$R_{-I\Omega} = \begin{pmatrix} \cos(-I\Omega) & -\sin(-I\Omega) & 0 \\ \sin(-I\Omega) & \cos(-I\Omega) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

By noting that

$$\cos(-I\Omega) = \cos\ I\Omega = \cos\ \Omega$$
$$\sin(-I\Omega) = -\sin\ I\Omega = -I\ \sin\ \Omega$$

the above rotation matrix can be rewritten as

$$R_{-I\Omega} = \begin{pmatrix} \cos\Omega & I\sin\Omega & 0 \\ -I\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure A.4 displays the results of these rotations for the direct equinoctial element set. The overall rotation matrix can then be determined as the product of these three individual matrices such that

$$R = R_{\Omega} R_{i} R_{-I\Omega}$$

$$R = \begin{pmatrix} c_{\Omega}^{2} + Is_{\Omega}^{2}c_{i} & Ic_{\Omega}s_{\Omega}(1 - Ic_{i}) & s_{\Omega}s_{i} \\ c_{\Omega}s_{\Omega}(1 - Ic_{i}) & I(s_{\Omega}^{2} + Ic_{\Omega}^{2}c_{i}) & -c_{\Omega}s_{i} \\ -Is_{\Omega}s_{i} & c_{\Omega}s_{i} & c_{i} \end{pmatrix} \qquad (A.18)$$

By inverting the equinoctial element definitions for p and q and solving for sin i, cos i, sin Ω and cos Ω, the above rotation matrix can be written in terms of p and q

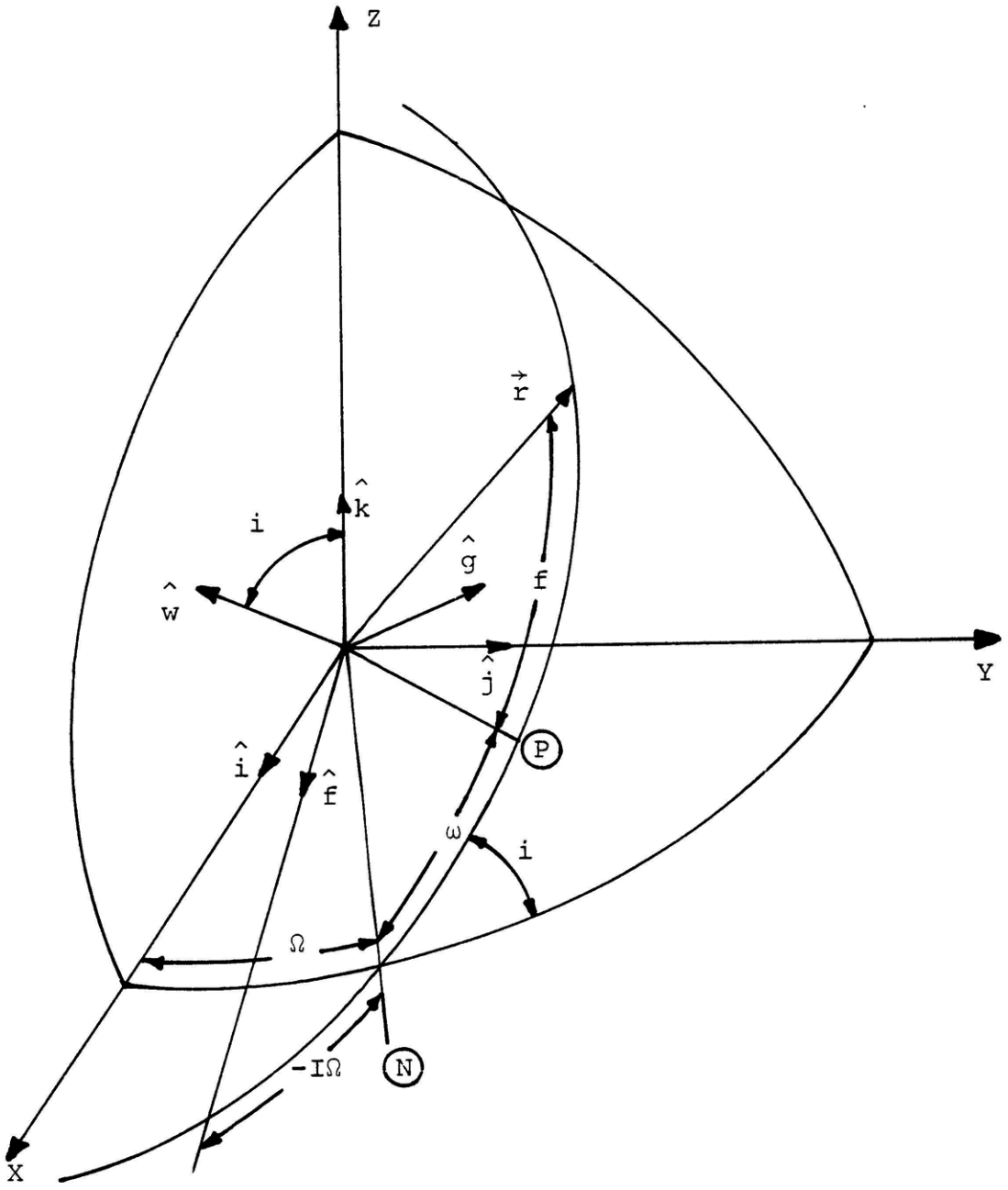


Figure A.4. Spatial Orientation of the Equinoctial Reference Frame for the Direct Equinoctial Set

$$R = \frac{1}{1 + p^2 + q^2} \begin{pmatrix} 1 - p^2 + q^2 & 2pqI & 2p \\ 2pq & (1 + p^2 - q^2)I & -2q \\ -2pI & 2q & (1 - p^2 - q^2)I \end{pmatrix}$$

(A.19)

Examination of Figure A.3 indicates that the position vector coordinatized in the equinoctial reference frame can be written as

$$\vec{r}_{eq} = X \hat{f} + Y \hat{g} \qquad \text{(A.20)}$$

where

$$X = r \cos L \qquad \text{(A.21a)}$$

$$Y = r \sin L \qquad \text{(A.21b)}$$

$$L = \text{true longitude} = f + \omega + I\Omega \qquad \text{(A.21c)}$$

By inverting the equinoctial element definitions for h, k and L and substituting the results into Equation (A.2), the magnitude of the position vector is given by

$$r = \frac{a(1 - h^2 - k^2)}{1 + k \cos L + h \sin L} \tag{A.22}$$

By differentiating Equation (A.20), the velocity vector can be written as

$$\dot{\vec{r}}_{eq} = \dot{X}\,\hat{f} + \dot{Y}\,\hat{g} \tag{A.23}$$

where

$$\dot{X} = \frac{-na(h + \sin L)}{\sqrt{1 - h^2 - k^2}} \tag{A.24a}$$

$$\dot{Y} = \frac{na(k + \cos L)}{\sqrt{1 - h^2 - k^2}} \tag{A.24b}$$

Again these vectors can be coordinatized in the inertial frame by application of the rotation matrix of Equation (A.19), such that

$$\vec{r} = R \begin{pmatrix} X \\ Y \\ 0 \end{pmatrix} \tag{A.25a}$$

$$\dot{\vec{r}} = R \begin{pmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{pmatrix} \tag{A.25b}$$

It should be noted that the columns of R are the components of the equinoctial reference frame unit vectors $\hat{f}$, $\hat{g}$ and $\hat{w}$ in the inertial frame, such that

$$R = \begin{pmatrix} \vdots & \vdots & \vdots \\ \hat{f} & \hat{g} & \hat{w} \\ \vdots & \vdots & \vdots \end{pmatrix} \qquad \text{(A.26)}$$

As in the Keplerian case, the form of Equations (A.25a) and (A.25b) is the most convenient for computations of the two body partial derivatives of the position and velocity vectors with respect to the equinoctial elements (see Chapter 3.3.2).

A.3 Lagrange Coefficient Two Body Analytical Solution

Reference 28 presents a simple derivation of an analytical two body solution which does not resort to series expansions. The position vector in the orbit plane coordinate system is given by Equation (A.11) as

$$\vec{r} = r \cos f \, \hat{i}_\xi + r \sin f \, \hat{i}_\eta \qquad \text{(A.27)}$$

By differentiating this expression, the velocity vector can be found

$$\begin{aligned} \dot{\vec{r}} &= (\dot{r} \cos f - r\dot{f} \sin f)\hat{i}_\xi \\ &+ (\dot{r} \sin f + r\dot{f} \cos f)\hat{i}_\eta \end{aligned} \qquad \text{(A.28)}$$

But $$r = \frac{p}{1 + e \cos f} = \frac{a(1 - e^2)}{1 + e \cos f} \qquad (A.28)$$

(Note: p = semilatus rectum)

so $$\dot{r} = \frac{p\, e \dot{f} \sin f}{(1 + e \cos f)^2}$$

$$= \frac{r^2 e}{p} \dot{f} \sin f$$

From Equation (A.5)

$$\dot{f} = \frac{h}{r^2}$$

$$\Rightarrow \qquad \dot{r} = \frac{eh}{p} \sin f$$

But $$p = \frac{h^2}{\mu}$$

Thus $$\dot{r} = \frac{\mu e}{h} \sin f \qquad (A.29)$$

By substituting this into Equation (A.28) and using Equations (A.2) and (A.5), the velocity expression is given by

$$\dot{\underset{\sim}{r}} = \sqrt{\frac{\mu}{p}} \; [-\sin f \, \hat{i}_\xi + (e + \cos f) \, \hat{i}_\eta] \qquad (A.30)$$

When Equations (A.27) and (A.30) are evaluated at the epoch, the result is

$$\vec{r}_o = r_o \cos f_o \, \hat{i}_\xi + r_o \sin f_o \, \hat{i}_\eta \qquad \text{(A.31a)}$$

$$\dot{\vec{r}}_o = \sqrt{\frac{\mu}{p}} \left[-\sin f_o \, \hat{i}_\xi + (e + \cos f_o) \hat{i}_\eta \right] \qquad \text{(A.31b)}$$

These equations can now be solved for the unit vectors $\hat{i}_\xi$ and $\hat{i}_\eta$ in terms of $\vec{r}_o$ and $\dot{\vec{r}}_o$ such that

$$\hat{i}_\xi = \frac{e + \cos f_o}{p} \vec{r}_o - \frac{r_o}{\sqrt{\mu p}} \sin f_o \, \dot{\vec{r}}_o \qquad \text{(A.32a)}$$

$$\hat{i}_\eta = \frac{\sin f_o}{p} \vec{r}_o + \frac{r_o}{\sqrt{\mu p}} \cos f_o \, \dot{\vec{r}}_o \qquad \text{(A.32b)}$$

The above relations can then be substituted into Equations (A.27) and (A.28), such that the position and velocity at some time t can be found in terms of the epoch position and velocity

$$\vec{r} = R \vec{r}_o + G \dot{\vec{r}}_o \qquad \text{(A.33a)}$$

$$\dot{\vec{r}} = F_t \vec{r}_o + G_t \dot{\vec{r}}_o \qquad \text{(A.33b)}$$

where $F = 1 - \frac{r}{p}[1 - \cos(f - f_o)]$

$$G = \frac{rr_o}{h} \sin(f - f_o)$$

(A.34)

$$F_t = \frac{\vec{r}_o \cdot \dot{\vec{r}}_o}{pr_o}[1 - \cos(f - f_o)] - \frac{h}{pr_o} \sin(f - f_o)$$

$$G_t = 1 - \frac{r_o}{p}[1 - \cos(f - f_o)]$$

with $h = |\vec{r}_o \times \dot{\vec{r}}_o|$

These coefficients are called the Lagrange coefficients. The position vector magnitude r in these coefficients is computed by

$$r = \frac{pr_o}{r_o + (p - r_o)\cos(f - f_o) - \frac{h}{\mu}\vec{r}_o \cdot \dot{\vec{r}}_o \sin(f - f_o)}$$

(A.35)

which is easily derived from Equation (A.2) in a similar manner to that above. This is the analytical two body solution programmed in this thesis and subroutine 'LGRNGE' (see Appendix D) computes the necessary coefficients.

APPENDIX B

SIMPLIFIED FORCE MODELS FOR THE STAND-ALONE PARALLEL PROCESSOR ORBIT DETERMINATION SYSTEM

Significant complexities and interface difficulties were anticipated in attempting to implement the Parallel Variable Metric orbit estimation algorithm into the GTDS-RD software test bed. Because of this, it was decided to implement a stand-alone version of the integrator and estimator. It was therefore necessary to generate certain simplified force models for use with this stand-alone version. This appendix briefly describes the implementation of a gravitational model including zonal harmonics J_2 through J_8 and a very rudimentary drag model.

B.1 Zonal Gravitational Potential Model

The formulae for computing the gravitational force of attraction on the satellite are taken directly from Chapter 1 of Reference 28. It is assumed that the gravitational force per unit mass acting on the satellite at a point with spherical coordinates r, ϕ, θ can be derived as the gradient of the gravitational potential function.

By making several simplifying assumptions, Battin presents the following formula for the gradient of the gravitational potential

$$\vec{\nabla}V = -\frac{Gm}{r^2}\left\{\hat{i}_r - \sum_{k=2}^{\infty} J_k \left(\frac{r_{eq}}{r}\right)^k \left[P'_{k+1}(\cos\phi)\hat{i}_r - P'_k(\cos\phi)\,\hat{i}_z\right]\right\} \qquad \text{(B.1)}$$

where

G	=	gravitation constant
m	=	mass of central body (assuming satellite mass is negligible)
r_{eq}	=	equatorial radius of central body
r,ϕ,θ	=	spherical coordinates of satellite position
$\hat{i}_r$	=	unit vector from center of mass toward the satellite
$\hat{i}_z$	=	unit vector along central body axis of symmetry
J_k	=	k^{th} zonal harmonic coefficient
$P'_k(\cos\phi)$	=	derivative of the k^{th} Legendre polynomial with respect to its argument $\cos\phi = \hat{i}_r \cdot \hat{i}_z$

Battin also provides a recursive formula for computing the necessary derivatives of the Legendre polynomials as

$$(k - 1)\ P'_k(\nu) = (2k - 1)\ \nu P'_{k-1}(\nu) - kP'_{k-2}(\nu) \qquad (B.2)$$

with starting values $P'_0(\nu) = 0$

and $P'_1(\nu) = 1$

The formula (B.1) above is limited to $k \leq 8$ for the stand-alone version. No tesseral or sectoral harmonics are considered. The necessary central body information for the Earth is stored in common block /ERTSHP/ and the harmonic coefficients for the Earth are stored in common block /ZNLCOF/. These blocks are documented in Appendix D.

B.2 Drag and Atmospheric Models

The drag and atmospheric models used in the stand-alone software version are very simple. The drag acceleration is computed by the formula (46)

$$\vec{a}_{drag} = -\frac{1}{2}\ \frac{C_D A}{m}\ \rho\ \vec{v}_{rel}\ |\vec{v}_{rel}| \qquad (B.3)$$

where C_D = coefficient of drag

A = spacecraft reference area (km^2)

m = spacecraft mass (kg)

$\vec{v}_{rel}$ = spacecraft velocity relative to the atmosphere

ρ = atmospheric density

The above formula assumes a spherical spacecraft. The velocity of the satellite relative to the rotating atmosphere can be computed by

$$\vec{v}_{rel} = \vec{v} - \vec{v}_{A/I} \tag{B.4}$$

where $\vec{v}_{rel}$ = velocity of spacecraft relative to rotating atmosphere

$\vec{v}$ = velocity of spacecraft relative to inertial space

$\vec{v}_{A/I}$ = velocity of the atmosphere relative to inertial space

If it is assumed that the atmosphere rotates as a rigid body with the rotational velocity of the Earth, then the velocity of the atmosphere at the satellite inertial position $\vec{r}$ can be found by

$$\vec{v}_{A/I} = \vec{\omega} \times \vec{r} \tag{B.5}$$

with $\vec{\omega}$ = rotational velocity of the Earth

$= \omega_e \hat{i}_z$

Thus, for input satellite inertial position and velocity vectors $\vec{r}$ and $\vec{v}$, the velocity of the satellite relative to the atmosphere is given by

$$\vec{v}_{rel} = \vec{v} - \vec{\omega} \times \vec{r} \tag{B.6}$$

The drag acceleration is then computed by Equation (B.3) once the atmospheric density ρ has been found.

An exponential density model of the form

$$\rho = \rho_o \, e^{-\left(\frac{r - r_o}{\lambda}\right)} \tag{B.7}$$

where ρ = atmospheric density at radial distance $r = \sqrt{\vec{r} \cdot \vec{r}}$

ρ_o = density at reference distance r_o

r_o = reference distance for density computation

λ = atmospheric scale height

is assumed for this work. Thus, for an input satellite inertial position vector $\vec{r}$, the atmospheric density at that position can be computed given the reference distance, the density at the reference distance and the atmospheric

scale height. The value of ρ is then used in the drag acceleration calculation of Equation (B.3).

APPENDIX C

GODDARD TRAJECTORY DETERMINATION SYSTEM (GTDS-RD) SOFTWARE MODIFICATIONS

The initial implementation of the Picard-Chebyshev integration algorithm was accomplished in the research and development version of the Goddard Trajectory Determination System (GTDS-RD). The software interfaces required to apply the Picard-Chebyshev algorithm to the Cowell equations of motion already exist due to the work of Feagin. Additionally, the existence of the extensive force models for the Variation of Parameters (VOP) equations of motion greatly simplified the implementation of the Picard-Chebyshev algorithm for these formulations.

This appendix describes briefly the modifications made to existing GTDS-RD subroutines to implement the Picard-Chebyshev orbit propagation algorithm. Additionally, the new subroutines required for this capability are described. Input of the parameters necessary to initialize the Picard-Chebyshev algorithm for the VOP formulation required augmentation of certain GTDS-RD keyword cards. These changes

are described in this appendix. Finally, program listings for those GTDS-RD programs modified are included.

C.1 Modified GTDS-RD Subroutines

Several GTDS-RD subroutines required modification in order to incorporate the Picard-Chebyshev integrator over the full set of Special Perturbation formulations. General descriptions of these subroutines may be found in References 46 and 155. Only those modifications particular to the Picard-Chebyshev implementation will be discussed here.

The first subroutine requiring modification was subroutine 'SETRUN'. This subroutine acts as the primary input processor which reads the GTDS keyword cards in the input deck and performs appropriate initialization of data variables. 'SETRUN' was modified to process augmented fields on certain keyword cards. The ORBTYPE keyword card processing has been modified to allow input of the Picard-Chebyshev integration parameters. The format of the modified keyword card images for both the Cowell and the VOP formulations are given in Section C.3 of this appendix. Subroutine 'SETRUN' loads the necessary input data for the Picard-Chebyshev integrator into two new common data blocks. The first block is common /NPCNTL/, which contains the

order of the Chebyshev series, the number of orthogonality points, the identifier for the particular equations of motion being integrated and the terminal iteration number. The second block is common /CNVRG/, which contains the convergence criteria for the six state equations being integrated by the Picard-Chebyshev method and also the convergence criterion used in the subroutine 'KEPLR' for the iterative solution of the Kepler equation. Finally, subroutine 'SETRUN' makes use of the twentieth entry in the GTDS-RD common block /SWITCH/ as a flag to indicate that the Picard-Chebyshev integrator is to be used with the VOP formulation (1 = true, 2 = false).

Once the input cards are processed, GTDS-RD prints out a report of the initial conditions. Subroutine 'OUTOG1' controls the generation of the orbit generator initial conditions report to the on-line printer. This subroutine was modified to correct a compilation error due to misdimension of an array which was detected by the Fortran IV (H extended) compiler. Subroutine 'OUTOG1' calls subroutine 'OUTGEN' to print the orbit generator options portion of the initial conditions report. Subroutine 'OUTGEN' was modified to print an indication when the Picard-Chebyshev integrator is used in conjunction with the VOP orbit generator option.

Once the actual ephemeris generation has begun for the Cowell formulation, control is passed to subroutine 'ORBITT'. In the original implementation by Feagin, 'ORBITT' called subroutine 'COPS' to perform the Picard-Chebyshev integration for each arc and subroutine 'CHIRP' to evaluate the resulting position and velocity Chebyshev series at required intermediate output points. These calls have been replaced by calls to new subroutines 'PCHEBY' and 'CSRSTA' to perform the above functions. Additionally, 'ORBITT' makes use of new common blocks /NPCNTL/ and /COVAL/. Common block /NPCNTL/ has been previously described. Common block /COVAL/ contains the array of the integrand series coefficients, the array of series coefficients of the integrated results and the array of the evaluations of the integrands at the Chebyshev orthogonality points. These are the arrays [A], [BC] and [FORC] described in Figures 2.5, 2.6 and 2.4 respectively. Subroutine 'PCHEBY' is documented in the next section of this appendix.

When the subroutine 'ORBITT' detects a flight section crossing condition, 'ORBITT' calls subroutine 'CROSST'. This subroutine controls the execution of a section crossover for the Picard-Chebyshev integrator. Subroutine 'CROSST' must interpolate for the integration state at the sectioning point. Under the previous implementation,

'CROSST' called 'CHIRP' to evaluate the position and velocity Chebyshev series. The implementation described in this document replaces that call with a call to the new subroutine 'CSRSTA,' which is described in Appendix D.

More extensive changes were required in order to incorporate the Picard-Chebyshev integrator into the VOP orbit generator. GTDS-RD has the capability to integrate the high precision VOP equations of motion in several different orbital parameter sets. The primary element sets of interest in this work were the Keplerian elements, the equinoctial elements with mean longitude as the fast variable and the equinoctial elements with eccentric longitude as the fast variable. Additionally, the capability also exists in GTDS-RD to integrate the averaged VOP equations of motion, which are formulated in mean equinoctial elements with the mean mean-longitude as the fast variable. The implementation of the Picard-Chebyshev orbit propagator described here allows integration of all the above VOP formulations.

Prior to beginning the actual VOP integration, the current GTDS-RD software must initialize the multistep integrator. This is done by subroutine 'MULSTP', which computes the necessary starting values of the time derivatives

of the parameters being integrated and the sums required by the Adams-Cowell multistep integrator. In general, this type of starter is not required for a single step integrator such as the Picard-Chebyshev method. Therefore, 'MULSTP' has been modified to check the flag ICHEBY (IND(20) in common /SWITCH/). If this flag is set true (ICHEBY = 1), it means the Picard-Chebyshev algorithm is to be used to integrate the VOP equations and the multistep starter may be bypassed. For the averaged VOP equations, a call to 'AVRINT' is made to set up the appropriate averaging force models. 'MULSTP' then loads the variation of parameter elements from array VPELMT into array PARARY for temporary storage and returns immediately, bypassing the multistep starter software.

Once the starter has been completed, control is passed to subroutine 'ORBITV'. This subroutine drives the integration or interpolation of the VOP equations of motion. If flag ICHEBY in common /SWITCH/ is set true, the multistep integrator ('CSTEPX') is bypassed and a call is made to the Picard-Chebyshev integrator ('PCHEBY') when an integration step is to be taken. Additionally, subroutine 'ORBITV' calls 'CSRSTA' to evaluate the integrated element Chebyshev series results at any required intermediate output points.

As in the Cowell integration driver 'ORBITT', the subroutine 'ORBITV' must detect and respond to flight section crossing conditions. When a sectioning condition is detected by 'ORBITV', it calls subroutine 'CROSSV' to interpolate the orbital elements at the section crossing. Subroutine 'CROSSV' has been modified to use the new Chebyshev series evaluation subroutine 'CSRSTA' to perform this required interpolation.

The major restriction with the current implementation of the Picard-Chebyshev capability in the GTDS-RD system is that there is presently no capability to integrate variational equations. Therefore, if state partial derivatives are desired they must be computed using the GTDS-RD analytical partial derivative option only.

C.2 New GTDS-RD Subroutines

Only two new subroutines were created that interface directly with the GTDS-RD system. Subroutine 'PCHEBY' was based on the existing GTDS-RD subroutine 'COPS'. The purpose of 'PCHEBY' is to integrate by the method of Picard iteration the components of a vector differential equation whose right hand sides have been approximated by series of Chebyshev polynomials. This subroutine is listed here

because it is slightly different than its counterpart in the stand-alone software version described in Appendix D. Subroutine 'CSRSTA' is used to compute the integration state vector components of a spacecraft at some time by evaluating the appropriate Chebyshev series.

The description of subroutine 'PCHEBY' is presented in Section 2.3.1 of this thesis. This subroutine is called by either 'ORBITT' or 'ORBITV', depending on which equation set is being integrated. 'PCHEBY' calls subroutines 'VORTPT', 'VMATTX', 'VCOVAL', 'VCOINT', 'F2BDY', 'FORCES', 'EPELMS', 'CSER1', 'KEPLR', 'LGRNGE' and 'EQUMTN'. Subroutines 'FORCES' and 'EQUMTN' are existing GTDS-RD subroutines and are documented in Reference . The other subroutines are new and are also used in the stand-alone software version. They are therefore described in Appendix D.

C.3 Modified GTDS-RD Keyword Cards

Implementation of the Picard-Chebyshev orbit propagator into the GTDS-RD system required some augmentation of the available fields of the ORBTYPE keyword card image. This was done to allow card input of the appropriate control parameters necessary to operate the Picard-Chebyshev

integrator. Figure C.1 presents the modified ORBTYPE card for the Picard-Chebyshev integration of the Cowell equations of motion. Figure C.2 shows the modified format for the Picard-Chebyshev integration of the Variation of Parameters equations of motion.

C.4 Program Listings

The following pages contain the FORTRAN IV listings of the GTDS-RD subroutines modified to incorporate the Picard-Chebyshev capability. Also included are listings of new subroutines 'PCHEBY' and 'EPELMS' because they are somewhat different than their corresponding stand-alone versions. The following additional subroutines are required to execute a GTDS-RD Picard-Chebyshev run: 'CSER1', 'CSRSTA', 'F2BDY', 'KEPLR', 'LGRNGE', 'VCOINT', 'VCOVAL', 'VMATTX' and 'VORTPT'. Listings of these programs are provided in Appendix D.

ORBTYPE (Picard-Chebyshev/Cowell)

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: EPHEM, DC, FILTER
- Detailed format:

Columns	Format	Description
1-8	A8	ORBTYPE - keyword to select orbit generator type
9-11	I3	Orbit generator type: 9 = Picard-Chebyshev integrator/ Cowell equations of motion
12-14	I3	Integration step mode: 1 = Fixed step (default) 2 = Regular vary step 3 = Shell mode vary step 4 = Halving-doubling
15-17	I3	Coordinate system orientation: 1 = Mean equator and equinox of 1950.0 (default) 2 = True of reference
18-38	G21.14	Integration stepsize in seconds: Default = 5400 sec.
39-59	G21.14	Picard-Chebyshev control parameters: MM.NNII MM = MORD = order of Chebyshev approximating series (<68) NN = NSUB = number minus one of Chebyshev orthogonality points (<70) II = ITERM = terminal iteration number (<100) Default = 36.4999
60-80	G21.14	Algorithm convergence criterion: Default = 10^{-10}

Figure C.1. Modified ORBTYPE Keyword Card for Picard-Chebyshev/Cowell Orbit Generator

ORBTYPE (Picard-Chebyshev/VOP)

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: EPHEM, DC, FILTER
- Detailed format:

Columns	Format	Description
1-8	A8	ORBTYPE - keyword to select orbit generator
9-11	I3	Orbit generator type: 5 = Variation of Parameters
12-14	I3	Integration step mode: 1 = Fixed step (default) 2 = Regular vary step 3 = Shell mode vary step 4 = Halving-doubling
15-17	I3	Coordinate system orientation: 1 = Mean equator and equinox of 1950.0 (default) 2 = True of reference
18-38	G21.14	Integration stepsize in seconds: Default = 5400 sec (for P-C integrator) = 24 sec (for multistep integrator)
39-59	G21.14	Picard-Chebyshev integration flag: Blank = Picard-Chebyshev not used 1.JJ = Picard-Chebyshev integrator used and 10^{-JJ} = algorithm convergence criterion

Figure C.2. Modified ORBTYPE Keyword Card for Picard-Chebyshev/Variation of Parameters Orbit Generator (page 1 of 2)

Columns	Format	Description
60-80	G21.14	Type of VOP parameter/Picard-Chebyshev control parameters: JJ.MMNNII JJ = Type of orbit parameters 2 = Equinoctial (mean longitude) 3 = Equinoctial (eccentric longitude) 8 = Keplerian 12 = Averaged equinoctial (mean longitude) 18 = Averaged Keplerian MM = order of Chebyshev approximating series (<68) NN = number minus one of Chebyshev orthogonality points (<70) II = terminal iteration number (<100)

Figure C.2. Modified ORBTYPE Keyword Card for Picard-Chebyshev/Variation of Parameters Orbit Generator (page 2 of 2)

```
CROSST              JSS1493.GTDS.UPDATE.FORT

      SUBROUTINE CROSST (MSGERR,*)                                          00000010
      IMPLICIT REAL*8(A-H,O-Z)                                              00000020
C     VERSION 2.2                                                           00000030
C     VERSION  OF  MAY 15,  1974                                            00000040
C  THIS SUBROUTINE IS JUST LIKE CROSSC EXCEPT ...                           00000050
C  CHIRP REPLACES INTP FOR PARTIALS INTERPOLATION.                          00000060
C  CSRSTA REPLACES CHIRP FOR STATE INTERPOLATION.                           00000070
C                                                                          00000080
C     THIS ROUTINE CONTROLS EXECUTION OF A SECTION CROSSOVER FOR THE       00000090
C     CHEBYSHEV ORBIT GENERATOR.  WHENEVER A SECTION IS INDICATED, THIS    00000100
C     ROUTINE IS ENTERED TO -                                              00000110
C                            1  INTERPOLATE FOR STATE AND PARTIALS         00000120
C                               AT POINT OF CROSSING                       00000130
C                            2  PRINT CONDITIONS AT END OF SECTION         00000140
C                            3  PRINT CONDITIONS AT START OF NEW SECT      00000150
C                               (2 AND 3 FOR ORBIT GENERATOR ONLY)         00000160
C                            4  CALL SECUPD TO LOAD NEW FORCE MODEL AND    00000170
C                               ALL OTHER SECTION DEPENDENT FLAGS          00000180
C                            5  RESTART THE INTEGRATOR                     00000190
C     AN ENTRY POINT (STARTC) IS ALSO AVAILABLE WHICH IS USED TO START     00000200
C     THE INTEGRATOR ONLY.  IT MAY BE EXECUTED AT THE START OF THE RUN     00000210
C     BY JUST CALLING STARTC OR IT ALSO IS EXECUTED EACH TIME A            00000220
C     NON-TERMINAL CROSSING HAS BEEN INDICATED.                            00000230
C                                                                          00000240
C     IF THE CROSSING CONDITION IS NOTED TO BE TERMINAL, THE ROUTINE       00000250
C     MERELY PRINTS END CONDITIONS (O G ONLY) AND RETURNS.                 00000260
C     A CHECK IS MADE TO SEE IF THE CONDITION FLAG HAS SET FOR MANEUVER    00000270
C     AND IF SO, A CALL IS MADE TO SUBROUTINE MANEUV TO PERFORM THE        00000280
C     MANEUVER, THEN THE INTEGRATOR IS RESTARTED AND THE ROUTINE IS        00000290
C     EXITED.  NO CALL TO SECUPD IS MADE SINCE A MANEUVER IS NOT           00000300
C     CONSIDERED A SECTION.                                                00000310
C                                                                          00000320
C     CALLING SEQUENCE                                                     00000330
C        CALL CROSST(&III)                                                 00000340
C             WHERE III IS THE STATEMENT NUMBER TO BE RETURNED TO          00000350
C             IN CASE OF ERROR                                             00000360
C                                                                          00000370
C                                                                          00000380
C     SUBROUTINES REQUIRED -                                               00000390
C                  MANEUV, CHIRP, ANPART, CUTOS2, SECUPD, CSRSTA           00000400
C                                                                          00000410
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * *   00000420
C                                                                          00000430
      COMMON/COVAL / A(70,6), BC(70,6), FORC(70,6)                         00000440
C                                                                          00000450
      COMMON/DCINT / DPDCI(432),    INTDCI(26)                             00000460
C                                                                          00000470
      COMMON/FRC   / DPFRC(1300),   INTFR(47)                              00000480
C                                                                          00000490
      COMMON/SATPOS/ DPSATP(100),   INTSAT(6)                              00000500
C                                                                          00000510
      COMMON/SECTN / DPSEC(675),    INSEC(325)                             00000520
C                                                                          00000530
      COMMON/SWITCH/ IND(100),      INTSW(100)                             00000540
C                                                                          00000550
      COMMON/WORKER/ DPWORK(3264),  INWORK(13)                             00000560
C                                                                          00000570
```

```
CROSST              JSS1493.GTDS.UPDATE.FORT

      COMMON/PASS  / DPAS(40)                                                00000580
C                                                                            00000590
      COMMON         PVPD(6,6)                                             00000600
C                                                                            00000610
C * * * * * * * * * *      DIMENSION STATEMENTS      * * * * * * * * * * *00000620
C                                                                            00000630
      DIMENSION      CV(1,1,1),CVD(1,1,1),STA(6)                           00000640
C                                                                            00000650
      DIMENSION      X(3), XD(3), XV(3,20), XVD(3,20), SECTIM(10),         00000660
     *               XTO(3), XDTO(3), PXTXO(6,20), RD(15), HD(15)          00000670
C                                                                            00000680
      DIMENSION      ICROSS(10)                                            00000690
C                                                                            00000700
C * * * * * * * * * *     EQUIVALENCE STATEMENTS     * * * * * * * * * * *00000710
C                                                                            00000720
C     FROM COMMON/DCINT /:                                                   00000730
      EQUIVALENCE    (DPDCI(1),   XTO(1)     ), (DPDCI(4) , XDTO(1)   ),00000740
     *               (DPDCI(7),   PXTXO(1,1) ), (DPDCI(127),TTO       ) 00000750
C                                                                            00000760
C     FROM COMMON/FRC   /:                                                   00000770
      EQUIVALENCE    (DPFRC(1274),REFDA4    )                              00000780
C                                                                            00000790
C     FROM COMMON/SATPOS/:                                                   00000800
      EQUIVALENCE    (DPSATP(94), TTUTC     ), (DPSATP(45), AZERO     ) 00000810
C                                                                            00000820
C     FROM COMMON/SECTN /:                                                   00000830
      EQUIVALENCE    (DPSEC(651), SECTIM(1) ), (INSEC(313), NCONDT    ),00000840
     *               (NSECTN    , INSEC(311)), (INSEC(1)  , ICROSS(1) ) 00000850
C                                                                            00000860
C     FROM COMMON/SWITCH/:                                                   00000870
      EQUIVALENCE    (INTSW(54),  NSTATE    )                              00000880
C                                                                            00000890
C     FROM COMMON/WORKER/:                                                   00000900
      EQUIVALENCE    (DPWORK(1),  X(1)      ), (DPWORK(4),  XD(1)     ),00000910
     *               (DPWORK(3215),T        ), (DPWORK(1),  STA(1)    ),00000920
     *               (INWORK(2),  K         ), (INWORK(5),  NEQ       ),00000930
     *               (DPWORK(3216),H        ), (DPWORK(127),XV(1,1)   ),00000940
     *               (DPWORK(3229),RD(1)    ), (DPWORK(187),XVD(1,1)  ),00000950
     *               (DPWORK(3244),HD(1)    ), (DPWORK(3217),H2       ),00000960
     *               (INWORK(1),IR          ), (DPAS(12)    ,RM       ) 00000970
C                                                                            00000980
C * * * * * * * * * *        LOGICAL VARIABLES        * * * * * * * * * * 00000990
C                                                                            00001000
      LOGICAL*1      SKIP                                                  00001010
C                                                                            00001020
C     CHECK IF THIS ENTRY SHOULD SKIP THE CALL TO THE STARTER                00001030
C     THIS IS DONE WHEN CROSS IS USED BY ROUTINES OTHER THAN                 00001040
C     SECTIONING ROUTINES (SUCH AS ERTS CHECKING FOR NODAL CROSSINGS         00001050
C     FOR PRINTOUT ONLY)                                                     00001060
C                                                                            00001070
      SKIP=.FALSE.                                                           00001080
      IF(ICROSS(NSECTN).EQ.0)SKIP=.TRUE.                                     00001090
C                                                                            00001100
C     FIRST CHECK FOR MANEUVERS - IF CONDITION FLAG INDICATES ONE,           00001110
C     CALL MANEUV, RESTART INTEGRATOR, AND RETURN.                           00001120
C                                                                            00001130
      IF(NCONDT.NE.7)GO TO 10                                                00001140
```

```
      CALL MANEUV (MSGERR,&350)                                          00001150
      RETURN                                                             00001160
   10 CONTINUE                                                           00001170
C                                                                        00001180
C     FIRST COMPUTE STATE AT TIME OF CROSSING                            00001190
C                                                                        00001200
      CALL CSRSTA(STA,T,SECTIM(NSECTN),H,1)                              00001210
C                                                                        00001220
C     IF NUMERICAL PARTIALS OR NO PARTIALS NEEDED, BRANCH AROUND CALL    00001230
C     TO ANALYTICAL PARTIALS ROUTINE.                                    00001240
C                                                                        00001250
      IF(IND(4).EQ.1.OR.NSTATE.EQ.0)GO TO 100                            00001260
C                                                                        00001270
C     COMPUTE STATE PARTIALS ANALYTICALLY                                00001280
C                                                                        00001290
      CALL ANPART(X,XD,SECTIM(NSECTN),2,PVPD)                            00001300
      DO 20 I=1,NSTATE                                                   00001310
      DO 20 J=1,3                                                        00001320
      XV(J,I)=PVPD(J,I)                                                  00001330
      XVD(J,I)=PVPD(J+3,I)                                               00001340
   20 CONTINUE                                                           00001350
C                                                                        00001360
C     IF NO VARIATIONAL EQUATIONS, SKIP INTERPOLATION, OTHERWISE CALL    00001370
C     INTP WHEN IN ANALYTICAL MODE FOR PARTIALS.                         00001380
C                                                                        00001390
      IF(NEQ.EQ.0)GO TO 200                                              00001400
      DO 30 I=1,NEQ                                                      00001410
      N=NSTATE+I                                                         00001420
      CALL CHIRP (XV(1,N),XVD(1,N),CV(1,1,I),CVD(1,1,I),T,SECTIM(NSECTN)00001430
     *     ,H,1)                                                         00001440
   30 CONTINUE                                                           00001450
      GO TO 200                                                          00001460
C                                                                        00001470
C     IF THERE ARE ANY PARTIALS TO BE COMPUTED DO SO( NUMERICAL MODE)    00001480
C                                                                        00001490
  100 IF(NEQ.EQ.0)GO TO 200                                              00001500
      DO 160 I=1,NEQ                                                     00001510
      CALL CHIRP (XV(1,I),XVD(1,I),CV(1,1,I),CVD(1,1,I),T,SECTIM(NSECTN)00001520
     *     ,H,1)                                                         00001530
  160 CONTINUE                                                           00001540
C                                                                        00001550
C     SET PRINT FLAG TO INDICATE FIRST PRINTING, SET T EQUAL TO CROSS    00001560
C     TIME, SAVE NEXT REQUEST TIME, THEN RESET REQUEST TIME TO           00001570
C     CROSS TIME FOR PRINT PURPOSES.                                     00001580
C                                                                        00001590
  200 NPRINT=1                                                           00001600
      T=SECTIM(NSECTN)                                                   00001610
      TSAVE=TTO                                                          00001620
      TTSAVE = TTUTC                                                     00001630
      TTO=T                                                              00001640
C                                                                        00001650
C     LOAD STATE AND PARTIALS INTO PRINT ARRAYS.                         00001660
C                                                                        00001670
  210 XTO(1)=X(1)                                                        00001680
      XTO(2)=X(2)                                                        00001690
      XTO(3)=X(3)                                                        00001700
      XDTO(1)=XD(1)                                                      00001710
```

```
CROSST              JSS1493.GTDS.UPDATE.FORT

      XDTO(2)=XD(2)                                                          00001720
      XDTO(3)=XD(3)                                                          00001730
      IF(NEQ.EQ.0)GO TO 240                                                  00001740
      DO 220 I=1,NEQ                                                         00001750
      PXTXO(1,I)=XV(1,I)                                                     00001760
      PXTXO(2,I)=XV(2,I)                                                     00001770
      PXTXO(3,I)=XV(3,I)                                                     00001780
      PXTXO(4,I)=XVD(1,I)                                                    00001790
      PXTXO(5,I)=XVD(2,I)                                                    00001800
      PXTXO(6,I)=XVD(3,I)                                                    00001810
  220 CONTINUE                                                               00001820
C                                                                            00001830
C     COMPUTE A1 TIME, THEN CONVERT TO UTC FOR CALL TO PRINT ROUTINE         00001840
C     (ORBIT GENERATOR ONLY)                                                 00001850
C                                                                            00001860
 240  ATC = REFDA4 + TTO/86400.D0                                            00001870
      TTUTC = TTO - TDIF(4,3,0.D0,ATC) + AZERO                               00001880
      IF(IND(48).NE.2)  GO TO 245                                            00001890
      CALL OUTCS2                                                            00001900
      IF(SKIP)GO TO 250                                                      00001910
      IF(NPRINT.EQ.2)  GO TO 245                                             00001920
      CALL UPPAGE(2,0,DUM)                                                   00001930
 245  CONTINUE                                                               00001940
C                                                                            00001950
C     IF THIS IS A TERMINAL SECTION - EXIT                                   00001960
C                                                                            00001970
      IF(NCONDT.GT.100)GO TO 999                                             00001980
C                                                                            00001990
C     IF WE HAVEN'T PRINTED START OF NEW SECTION, SET FLAG TO PRINT          00002000
C   THEN UPDATE TO THE NEW SECTION CONDITIONS, THEN GO BACK TO PRINT         00002010
C                                                                            00002020
      IF(NPRINT.EQ.2)GO TO 250                                               00002030
      NPRINT=2                                                               00002040
      CALL SECUPD (MSGERR,&350)                                              00002050
      IF(IND(48).EQ.2)  CALL UNIT(60)                                        00002060
      LINES = 3                                                              00002070
      IF(TSAVE.NE.TTO)   GO TO 210                                           00002080
      GO TO 210                                                              00002090
C                                                                            00002100
C     RESTORE THE REQUEST TIME, THEN RESTART THE INTEGRATOR                  00002110
C                                                                            00002120
  250 TTO=TSAVE                                                              00002130
      TTUTC = TTSAVE                                                         00002140
      GO TO 999                                                              00002150
  350 RETURN1                                                                00002160
  999 RETURN                                                                 00002170
      END                                                                    00002180

******** END OF MEMBER CROSST     218 RECORDS ********
```

```
      SUBROUTINE CROSSV(MSGERR,*)                                          00000010
C                                                                          00000020
C     VERSION 2.2                                                          00000030
C                                                                          00000040
C     PURPOSE                                                              00000050
C         TO CONTROL EXECUTION OF A SECTION CROSSOVER FOR THE VOP          00000060
C         ORBIT GENERATOR. WHENEVER A SECTION IS INDICATED, THIS           00000070
C         SUBROUTINE IS ENTERED TO                                         00000080
C                    1. INTERPOLATE FOR STATE AND PARTIALS AT              00000090
C                       TIME OF CROSSING                                   00000100
C                    2. PRINT CONDITIONS AT END OF SECTION                 00000110
C                    3. PRINT CONDITIONS AT START OF NEW SECTION           00000120
C                    4. CALL SECUPD TO LOAD NEW FORCE MODEL                00000130
C                    5. RESTART THE INTEGRATOR                             00000140
C                                                                          00000150
C         AN ENTRY POINT(STARTV) IS AVAILABLE WHICH IS USED TO START       00000160
C         THE INTEGRATOR                                                   00000170
C     CALLING SEQUENCE                                                     00000180
C         CALL CROSSV(MSGERR,&NNN)                                         00000190
C              WHERE NNN IS THE STATEMENT NO. TO BE RETURNED TO            00000200
C              IN CASE OF ERROR                                            00000210
C         CALL STARTV(MSGERR,&NNN)                                         00000220
C                                                                          00000230
C      INPUT                                                               00000240
C                                                                          00000250
C         VIA COMMON WORKER                                                00000260
C                                                                          00000270
C           XXSUM    = FIRST SUM VECTOR                                    00000280
C           NOPARM   = NUMBER OF ORBITAL ELEMENTS                          00000290
C           K        = DERIVATIVE ARRAY INDICATOR OF VALUE AT T=T(K)       00000300
C           NEQ      = NUMBER OF VARIATIONAL EQUATIONS TO BE INTEGRATED    00000310
C           DPARRY   = DERIVATIVE ARRAY OF VOP ELEMENTS                    00000320
C           T        = CURRENT INTEGRATION TIME                            00000330
C           H        = STEPSIZE                                            00000340
C                                                                          00000350
C         VIA COMMON DCINT                                                 00000360
C                                                                          00000370
C           TTO      = REQUESTED TIME                                      00000380
C                                                                          00000390
C         VIA COMMON FILES                                                 00000400
C                                                                          00000410
C           NOUT     = FRN OF PRINTER OUTPUT                               00000420
C                                                                          00000430
C         VIA COMMON FRC                                                   00000440
C                                                                          00000450
C           REFDA4   = MODIFIED JULIAN DATE CORRESPONDING TO EPOCH         00000460
C                                                                          00000470
C         VIA COMMON SATPOS                                                00000480
C                                                                          00000490
C           AZERO    = DIFFERENCE BETWEEN A.1 TIME AND UTC                 00000500
C                                                                          00000510
C         VIA COMMON SECTN                                                 00000520
C                                                                          00000530
C           NCONDT   = SECTIONING CONDITION CODE                           00000540
C           SECTIM(NSECTN) = TIME AT WHICH SECTIONING OCCURRED FOR         00000550
C                            EACH SECTION                                  00000560
C           NSECTN   = SECTION NUMBER                                      00000570
```

```
C                                                                       00000580
C         VIA COMMON SWITCH                                             00000590
C                                                                       00000600
C           LINES    = CURRENT LINE COUNT                               00000610
C           NSTATE   = NUMBER OF STATE UNKNOWNS                         00000620
C                                                                       00000630
C         VIA COMMON TIMREG                                             00000640
C                                                                       00000650
C           STREG    = CURRENT VALUE OF INDEPENDENT VARIABLE            00000660
C           TREG     = KS TIME DERIVATIVE VECTOR                        00000670
C                                                                       00000680
C       OUTPUT                                                          00000690
C                                                                       00000700
C         VIA COMMON WORKER                                             00000710
C                                                                       00000720
C           XV       = ARRAY OF POSITION PARTIALS AT T=T(K)             00000730
C           VPELMT   = VOP ORBITAL ELEMENTS                             00000740
C                                                                       00000750
C         VIA COMMON DCINT                                              00000760
C                                                                       00000770
C           XTO      = POSITION VECTOR AT TIME TTO                      00000780
C           XDTO     = VELOCITY VECTOR AT TIME TTO                      00000790
C           PXTXO    = PARTIAL DERIVATIVES AT TIME TTO                  00000800
C                                                                       00000810
C         VIA COMMON SATPOS                                             00000820
C                                                                       00000830
C           TTUTC    = OBSERVATION TIME IN A.1 SECONDS FROM EPOCH       00000840
C                                                                       00000850
C       EXTERNAL REFERENCES                                             00000860
C           INTPAR                                                      00000870
C           EQUNPV                                                      00000880
C           DALLPV                                                      00000890
C           CELEM                                                       00000900
C           IDLPV                                                       00000910
C           ANPART                                                      00000920
C           MANEUV                                                      00000930
C           MULSTP                                                      00000940
C           ERROUT                                                      00000950
C                                                                       00000960
C       PROGRAMMER                                                      00000970
C           K.S.NIMITZ, COMPUTER SCIENCES CORPORATION                   00000980
C                                                                       00000990
C       PROGRAM MODIFICATIONS                                           00001000
C           1.  DATE:    04/01/73                                       00001010
C               NAME:    K. S. NIMITZ                                   00001020
C               REASON:  ORIGINAL SUBROUTINE CROSSV                     00001030
C           2.  DATE:    07/15/73                                       00001040
C               NAME:    A. C. LONG                                     00001050
C               REASON:  CONFORM TO GTDS PROGRAMMING STANDARDS          00001060
C           3.  DATE:    04/28/74                                       00001070
C               NAME:    R. S. PAJERSKI                                 00001080
C               REASON:  IMPROVE SECTION HANDLING                       00001090
C           4.  DATE:    06/15/74                                       00001100
C               NAME:    R. S. PAJERSKI                                 00001110
C               REASON:  ADDED NEW ELEMENT SET-TYPE 8 (KEPLERIAN)       00001120
C           5.  DATE:    06/15/79                                       00001130
C               NAME:    CAPTAIN JEFFREY S. SHAVER, USAF                00001140
```

```
CROSSV              JSS1493.GTDS.UPDATE.FORT

C             REASON:  ADD CAPABILITY TO INTEGRATE VOP EQUATIONS WITH   00001150
C                      PICARD-CHEBYSHEV INTEGRATOR                      00001160
C                                                                       00001170
C*********************       START PROGRAM       **********************00001180
C                                                                       00001190
      IMPLICIT REAL*8(A-H,O-Z)                                          00001200
C                                                                       00001210
C * * * * * * * * * *        COMMON BLOCKS        * * * * * * * * * * * 00001220
C                                                                       00001230
      COMMON/DCINT / DPDCI(432)          ,      INTDCI(26)              00001240
C                                                                       00001250
      COMMON/FILES / INFIL(100)                                         00001260
C                                                                       00001270
      COMMON/FRC   / DPFRC(1300)         ,      INTFR(47)               00001280
C                                                                       00001290
      COMMON/INTEG / DUMMY(149)          ,      NORDER                  00001300
C                                                                       00001310
      COMMON/PASS  / DPAS(40)                                           00001320
C                                                                       00001330
      COMMON/SATPOS/ DPSATP(100)         ,      INSAT(6)                00001340
C                                                                       00001350
      COMMON/SECTN / DPSEC(675)          ,      INSEC(325)              00001360
C                                                                       00001370
      COMMON/SWITCH/ IND(100)            ,      INTSW(100)              00001380
C                                                                       00001390
      COMMON/TIMREG/ DTRG(94)                                           00001400
C                                                                       00001410
      COMMON/WORKER/ DPWORK(3264)        ,      INWORK(13)              00001420
C                                                                       00001430
      COMMON         DUM(6,20)           ,      PKPC(6,6)               00001440
C                                                                       00001450
      COMMON         PVPD(6,6)                                          00001460
C                                                                       00001470
C * * * * * * * * *      DIMENSION STATEMENTS      * * * * * * * * * * *00001480
C                                                                       00001490
      DIMENSION   GM(11)                                                00001500
C                                                                       00001510
      DIMENSION   VPELMT(10)    ,   DPARRY(15,10)   ,   XXSUM(10)       00001520
C                                                                       00001530
      DIMENSION   X(3)          ,   XD(3)           ,   XV(3,20)       ,00001540
     *            XTO(3)         ,   XDTO(3)         ,   PXTXO(6,20)    ,00001550
     *            XVD(3,20)      ,   SECTIM(10)                          00001560
C                                                                       00001570
      DIMENSION   U(4)          ,   US(4)           ,   TREG(40)        00001580
C                                                                       00001590
      DIMENSION   ICROSS(10)    ,   VOPTMP(6)                           00001600
C                                                                       00001610
      DIMENSION   VEVOP(6,20)   ,   VEVOPD(15,6,20) ,   XVSUM(6,20)     00001620
C                                                                       00001630
C * * * * * * * * *      EQUIVALENCE STATEMENTS      * * * * * * * * * *00001640
C                                                                       00001650
C     FROM COMMON/DCINT /:                                              00001660
      EQUIVALENCE (DPDCI(1),XTO(1))         , (DPDCI(4),XDTO(1))        ,00001670
     *            (DPDCI(7),PXTXO(1,1))     , (DPDCI(127),TTO)          00001680
C                                                                       00001690
C     FROM COMMON/FILES /:                                              00001700
      EQUIVALENCE (INFIL(22),NOUT)                                      00001710
```

```
C                                                                       00001720
C     FROM COMMON/FRC   /:                                              00001730
      EQUIVALENCE (DPFRC(1274),REFDA4)     ,                            00001740
     *            (DPFRC(2),GM(1))         , (INTFR(1),ICENT)          00001750
C                                                                       00001760
C     FROM COMMON/INTEG /:                                              00001770
      EQUIVALENCE (STPMIN,DUMMY(123))                                   00001780
C                                                                       00001790
C     FROM COMMON/SATPOS/:                                              00001800
      EQUIVALENCE (DPSATP(94),TTUTC)       , (DPSATP(45),AZERO)         00001810
C                                                                       00001820
C     FROM COMMON/SECTN /:                                              00001830
      EQUIVALENCE (DPSEC(651),SECTIM(1))   , (INSEC(313),NCONDT)       ,00001840
     *            (INSEC(1),ICROSS(1))     , (INSEC(311),NSECTN)        00001850
C                                                                       00001860
C     FROM COMMON/SWITCH/:                                              00001870
      EQUIVALENCE (INTSW(54),NSTATE)       , (INTSW(48),LINES)          ,00001880
     *            (IND(20),ICHEBY)                                      00001890
C                                                                       00001900
C     FROM COMMON/TIMREG/:                                              00001910
      EQUIVALENCE (STREG,DTRG(92))         , (TREG(1),DTRG(1))          00001920
C                                                                       00001930
C     FROM COMMON/WORKER/:                                              00001940
      EQUIVALENCE (VPELMT(1),DPWORK(2829)) , (XXSUM(1),DPWORK(2819))   ,00001950
     *            (NOPARM,INWORK(3))       ,                           00001960
     *            (INWORK(2),K)            , (INWORK(5),NEQ)           ,00001970
     *            (DPARRY(1,1),DPWORK(7))  , (DPWORK(3215),T)          ,00001980
     *            (DPWORK(3216),H)         , (DPWORK(127),XV(1,1)),    00001990
     *            (XVD(1,1),DPWORK(187))   ,                           00002000
     *            (X(1)    ,DPWORK(1)  )   ,                           00002010
     *            (XD(1)   ,DPWORK(4)  )   ,                           00002020
     *            (DPWORK(307)             , VEVOP(1,1))               ,00002030
     *            (DPWORK(427)             , VEVOPD(1,1,1))            ,00002040
     *            (DPWORK(2383)            , XVSUM(1,1))               00002050
      LOGICAL*1 SKIP                                                    00002060
C                                                                       00002070
C     CHECK IF THIS ENTRY SHOULD SKIP THE CALL TO THE STARTER           00002080
C     THIS IS DONE WHEN CROSS IS USED BY ROUTINES OTHER THAN            00002090
C     SECTIONING ROUTINES (SUCH AS ERTS CHECKING FOR NODAL CROSSINGS    00002100
C     FOR PRINTOUT ONLY)                                                00002110
C                                                                       00002120
      SKIP=.FALSE.                                                      00002130
      IF(ICROSS(NSECTN).EQ.0)SKIP=.TRUE.                                00002140
      NX = NORDER + 1                                                   00002150
C                                                                       00002160
      IBRNCH = IND(3)                                                   00002170
      STIME=SECTIM(NSECTN)                                              00002180
C                                                                       00002190
C     IS A MANEUVER INDICATED?                                          00002200
C                                                                       00002210
C                              NO                                       00002220
      IF(NCONDT .NE. 7) GO TO 10                                        00002230
      CALL MANEUV(MSGERR,&350)                                          00002240
      GO TO 290                                                         00002250
   10 CONTINUE                                                          00002260
      IF(INTSW(1).NE.1) GO TO 11                                        00002270
C                                                                       00002280
```

```
C     IS THIS A KS RUN? IF SO, CALL THE TIME INTERPOLATOR.                00002290
C                                                                         00002300
      IF(IBRNCH.EQ.6) CALL HEMITR(STIME,SECTIM(NSECTN),K,H,STREG,TREG)    00002310
C                                                                         00002320
C      IS THIS A DS RUN IF SO ,CALL THE INTERPOLATOR                      00002330
C                                                                         00002340
      IF((IBRNCH.EQ.7)) CALL HEMIDS(STIME,SECTIM(NSECTN),K,H,STREG,       00002350
     *TREG,NX,X,XD)                                                       00002360
      IF(IBRNCH.EQ.7)  GO TO 19                                           00002370
   11 CONTINUE                                                            00002380
C                                                                         00002390
C     INTERPOLATE FOR THE ORBITAL ELEMENTS AT THE SECTIONING TIME:        00002400
C     CHECK THE FLAG 'ICHEBY' TO SEE IF THE PICARD-CHEBYSHEV INTEGRATOR   00002410
C     IS BEING USED TO INTEGRATE THE VOP EQUATIONS.  IF SO, USE CSRSTA    00002420
C     TO INTERPOLATE FOR THE ELEMENTS AT THE SECTIONING TIME.             00002430
C                                                                         00002440
      IF (ICHEBY.EQ.1)                  GO TO 12                          00002450
      CALL INTPAR(VPELMT,DPARRY,XXSUM,STREG,STIME,K,.FALSE.,NX,           00002460
     *NOPARM,H)                                                           00002470
      GO TO 13                                                            00002480
   12 CALL CSRSTA(VOPTMP,STREG,STIME,H,1)                                 00002490
      DO 1210  INDX = 1,6                                                 00002500
 1210 VPELMT(INDX) = VOPTMP(INDX)                                         00002510
      DO 1220  INDX = 7,10                                                00002520
 1220 VPELMT(INDX) = 0.D0                                                 00002530
   13 IF(IBRNCH.GT.10) IBRNCH = IBRNCH-10                                 00002540
C                                                                         00002550
C     COMPUTE POSITION AND VELOCITY FROM THE INTERPOLATED ELEMENTS        00002560
C                                                                         00002570
C     TEST FOR TYPE OF ELEMENTS USED AND BRANCH TO THE APPROPRIATE        00002580
C       CONVERSION ROUTINE.                                               00002590
C                                                                         00002600
      GO TO (14,15,15,16,16,17,14,18),IBRNCH                              00002610
   14 CALL DALLPV(VPELMT,X  ,XD  ,IERR)                                   00002620
      GO TO 19                                                            00002630
   15 CALL EQUNPV(VPELMT,X  ,XD  ,IERR,IBRNCH)                            00002640
      GO TO 19                                                            00002650
   16  CALL IDLPV(VPELMT,X,XD,IBRNCH)                                     00002660
      GO TO 19                                                            00002670
   17 CALL DPEL2X(VPELMT,STIME,U,US,X,XD,SE2,CE2)                         00002680
      GO TO 19                                                            00002690
   18 CALL CELEM(VPELMT,GM(ICENT),X,XD)                                   00002700
C                                                                         00002710
C     IS THIS AN ORBIT GENERATOR RUN OR A DC WITH NO STATE PARTIALS?      00002720
C                                                                         00002730
C                                          YES                            00002740
   19 IF(IND(4).EQ.1 .OR. NSTATE.EQ.0) GO TO 100                          00002750
C                                                                         00002760
C COMPUTE ANALYTIC PARTIAL DERIVATIVES                                    00002770
C                                                                         00002780
      CALL ANPART(X  ,XD  ,SECTIM(NSECTN),2,PVPD)                         00002790
      DO 20 I = 1,NSTATE                                                  00002800
      DO 20 J=1,6                                                         00002810
      PXTXO(J,I) = PVPD(J,I)                                              00002820
   20 CONTINUE                                                            00002830
C                                                                         00002840
C     IF NO VARIATIONAL EQUATIONS SKIP INTERPOLATION.  OTHERWISE,         00002850
```

```
C     INTERPOLATE FOR PARTIALS.                                          00002860
C                                                                        00002870
      IF (NEQ.EQ.0 .OR. IND(3).NE.8) GO TO 210                           00002880
      DO 30 I=1,NEQ                                                      00002890
      N=NSTATE+I                                                         00002900
      CALL INTPAR (VEVOP(1,N),VEVOPD(1,1,I),XVSUM(1,I),STREG,STIME,K,    00002910
     *                .TRUE.,NX,NOPARM,H)                               00002920
      N = NSTATE                                                         00002930
   30 CONTINUE                                                           00002940
      GO TO 200                                                          00002950
C                                                                        00002960
C     IF THERE ARE ANY PARTIALS TO BE COMPUTED DO SO (NUMERICAL MODE).   00002970
C                                                                        00002980
  100 IF (NEQ.EQ.0 .OR. IND(3).NE.8) GO TO 210                           00002990
      DO 160 I=1,NEQ                                                     00003000
      CALL INTPAR (VEVOP(1,I),VEVOPD(1,1,I),XVSUM(1,I),STREG,STIME,K,    00003010
     *                .TRUE.,NX,NOPARM,H)                               00003020
      N = 0                                                              00003030
  160 CONTINUE                                                           00003040
C                                                                        00003050
C                                                                        00003060
C     COMPUTE CARTESEAN VARIATIONAL PARTIALS FROM KEPLERIAN VOP          00003070
C     VARIATIONAL PARTIALS                                               00003080
C                                                                        00003090
  200 CONTINUE                                                           00003100
      CALL KPART (X,PKPC)                                                00003110
      CALL INV2 (6)                                                      00003120
      CALL MATMUL (PKPC,VEVOP(1,N+1),PXTXO(1,N+1),6,6,20-N)              00003130
  210 CONTINUE                                                           00003140
C                                                                        00003150
      NPRINT = 1                                                         00003160
      T = SECTIM(NSECTN)                                                 00003170
      TSAVE = TTO                                                        00003180
      TTSAVE= TTUTC                                                      00003190
      TTO = T                                                            00003200
C                                                                        00003210
C     LOAD STATE VECTOR INTO PRINT ARRAYS                                00003220
C                                                                        00003230
  215 CONTINUE                                                           00003240
      XTO(1) = X(1)                                                      00003250
      XTO(2) = X(2)                                                      00003260
      XTO(3) = X(3)                                                      00003270
      XDTO(1) = XD(1)                                                    00003280
      XDTO(2) = XD(2)                                                    00003290
      XDTO(3) = XD(3)                                                    00003300
C                                                                        00003310
C                                                                        00003320
C                                                                        00003330
C     COMPUTE A1 TIME                                                    00003340
C                                                                        00003350
  240 ATC= REFDA4+TTO/86400.D0                                           00003360
      TTUTC= TTO-TDIF(4,3,0.D0,ATC)+AZERO                                00003370
      IF(IND(48).NE.2) GO TO 245                                         00003380
      CALL OUTCG2                                                        00003390
      IF(SKIP)GO TO 250                                                  00003400
      IF(NPRINT.EQ.2) GO TO 245                                          00003410
      CALL UPPAGE(2,0,DUM)                                               00003420
```

```
      WRITE(NOUT,2000) NSECTN                                          00003430
  245 CONTINUE                                                         00003440
C                                                                      00003450
C     IF THIS IS A TERMINAL SECTION - EXIT                             00003460
C                                                                      00003470
      IF(NCONDT .GT. 100) GO TO 999                                    00003480
C                                                                      00003490
      IF(NPRINT .EQ. 2) GO TO 250                                      00003500
      NPRINT = 2                                                       00003510
      CALL SECUPD(MSGERR,&350)                                         00003520
      IF(IND(48).EQ.2) CALL UNIT(60)                                   00003530
      LINES= 3                                                         00003540
      IF(TSAVE.NE.TTO) GO TO 215                                       00003550
C                                                                      00003560
C     RESTORE THE REQUESTED TIME, THEN RESTART THE INTEGRATOR          00003570
C                                                                      00003580
  250 TTO = TSAVE                                                      00003590
      TTUTC= TTSAVE                                                    00003600
      IF(SKIP)GO TO 999                                                00003610
C                                                                      00003620
C     ENTRY STARTV IS USED TO START THE INTEGRATOR                     00003630
C                                                                      00003640
      ENTRY STARTV(MSGERR,*)                                           00003650
  290 SIND = IND(30)                                                   00003660
      H = SIND*DABS(H)                                                 00003670
  300 CONTINUE                                                         00003680
      IND(27) = 1                                                      00003690
      IND(28) = 1                                                      00003700
C                                                                      00003710
C     COMPUTE THE STARTING ARRAY OF DERIVATIVES                        00003720
C                                                                      00003730
  320 CALL MULSTP(IERR)                                                00003740
C                                                                      00003750
C      DID THE STARTING PROCEDURE CONVERGE?                            00003760
C                                                                      00003770
C                               YES                                    00003780
      IF(IERR .EQ. 1) GO TO 999                                        00003790
C                               NO                                     00003800
      IF (IND(26).NE.1) GO TO 370                                      00003810
  340 MSGERR=IERR+306                                                  00003820
  350 RETURN 1                                                         00003830
  370 WRITE(6,2001) H                                                  00003840
      H=H/4.                                                           00003850
      IF(H.LT.STPMIN) GO TO 340                                        00003860
      GO TO 320                                                        00003870
C                                                                      00003880
  999 RETURN                                                           00003890
C                                                                      00003900
C                   FORMAT STATEMENTS                                  00003910
C                                                                      00003920
 2001 FORMAT(/93H ***GTDS MESSAGE***          INTEGRATOR STARTING PROCED00003930
     *URE DID NOT CONVERGE FOR STEP SIZE OF ,1PD17.10,22H  STEP SIZE QUA00003940
     *RTERED./)                                                        00003950
C                                                                      00003960
 2000 FORMAT(/,30X,'**********',3X,'END OF FLIGHT IN SECTION ',I2,      00003970
     *            '**********')                                        00003980
      END                                                              00003990
```

```
******** END OF MEMBER CROSSV      399 RECORDS ********
```

```
EPELMS              JSS1493.GTDS.UPDATE.FORT

      SUBROUTINE EPELMS(RV,T0,GM,ISET,ELMNTS)                          00000010
C                                                                      00000020
C     VERSION OF 08/09/79      **** CHECKED OUT ****                   00000030
C                                                                      00000040
C     PURPOSE                                                          00000050
C         TO COMPUTE THE KEPLERIAN OR EQUINOCTIAL ELEMENTS AT SOME     00000060
C         EPOCH TIME T0, GIVEN THE INERTIAL CARTESIAN COMPONENTS OF    00000070
C         POSITION AND VELOCITY AT THAT TIME.  IF ISET=1, THIS SUB-    00000080
C         ROUTINE WILL RETURN THE KEPLERIAN ELEMENTS AND CERTAIN       00000090
C         AUXILIARY QUANTITIES.  IF ISET=2, THIS SUBROUTINE WILL       00000100
C         RETURN THE NONSINGULAR EQUINOCTIAL ELEMENTS AND CERTAIN      00000110
C         AUXILIARY QUANTITIES.                                        00000120
C                                                                      00000130
C     CALLING SEQUENCE                                                 00000140
C         CALL EPELMS(RV,T0,GM,ISET,ELMNTS)                            00000150
C                                                                      00000160
C     PARAMETERS                                                       00000170
C         RV        I    ARRAY (6X1) CONTAINING THREE INERTIAL CAR-   00000180
C                        TESIAN COORDINATES OF POSITION AND THREE      00000190
C                        INERTIAL CARTESIAN COORDINATES OF VELOCITY    00000200
C                        AT THE EPOCH TIME.                            00000210
C         T0        I    EPOCH TIME.                                  00000220
C         GM        I    GRAVITATIONAL PARAMETER.                     00000230
C         ISET      I    FLAG TO SELECT DESIRED ELEMENT SET.          00000240
C         ELMNTS    O    ARRAY (26X1) CONTAINING APPROPRIATE ELEMENTS 00000250
C                        AND AUXILIARY QUANTITIES.                     00000260
C                        FOR ISET=1:  KEPLERIAN ELEMENTS               00000270
C                        ELMNTS(1)  = SEMIMAJOR AXIS, A                00000280
C                        ELMNTS(2)  = ECCENTRICITY, E                  00000290
C                        ELMNTS(3)  = INCLINATION, I                   00000300
C                        ELMNTS(4)  = ARGUMENT OF PERIAPSIS            00000310
C                        ELMNTS(5)  = LONGITUDE OF THE ASCENDING NODE  00000320
C                        ELMNTS(6)  = TIME OF PERIAPSIS PASSAGE (SEC)  00000330
C                        ELMNTS(7)  = MEAN MOTION (DEG/SEC)            00000340
C                        ELMNTS(8)  = SEMI-LATUS RECTUM (PARAMETER)    00000350
C                        ELMNTS(9)  = LONGITUDE OF PERIAPSIS           00000360
C                        ELMNTS(10) = TRUE ANOMALY AT EPOCH            00000370
C                        ELMNTS(11) = ARGUMENT OF LATITUDE AT EPOCH    00000380
C                        ELMNTS(12) = TRUE LONGITUDE AT EPOCH          00000390
C                        ELMNTS(13) = MAGNITUDE OF ANGULAR MOMENTUM    00000400
C                        ELMNTS(14) = MAGNITUDE OF POSITION VECTOR     00000410
C                        ELMNTS(15) = MAGNITUDE OF VELOCITY VECTOR     00000420
C                        ELMNTS(16) = Z-COMPONENT OF ANGULAR MOMENTUM  00000430
C                        ELMNTS(17) = PERIFOCAL RADIUS                 00000440
C                        ELMNTS(18) = APOFOCAL RADIUS                  00000450
C                        ELMNTS(19) = PERIOD                           00000460
C                        ELMNTS(20) = ENERGY (OSCULATING)              00000470
C                        ELMNTS(21) = ENERGY (ZONAL PERTURBED)         00000480
C                        ELMNTS(22) = ECCENTRIC ANOMALY AT EPOCH       00000490
C                        ELMNTS(23) = MEAN ANOMALY AT EPOCH            00000500
C                        ELMNTS(24) = RIGHT ASCENSION                  00000510
C                        ELMNTS(25) = DECLINATION                      00000520
C                        ELMNTS(26) = MEAN LONGITUDE AT EPOCH          00000530
C                                                                      00000540
C                        FOR ISET=2:  EQUINOCTIAL ELEMENTS             00000550
C                        ELMNTS(1)  = SEMIMAJOR AXIS, A                00000560
C                        ELMNTS(2)  = H                                00000570
```

```
C                       ELMNTS(3)  = K                                      00000580
C                       ELMNTS(4)  = P                                      00000590
C                       ELMNTS(5)  = Q                                      00000600
C                       ELMNTS(6)  = LAMDA, MEAN LONGITUDE AT EPOCH         00000610
C                       ELMNTS(7)  = MEAN MOTION (DEG/SEC)                  00000620
C                       ELMNTS(8)  = SEMI-LATUS RECTUM (PARAMETER)          00000630
C                       ELMNTS(9)  = ECCENTRICITY, E                        00000640
C                       ELMNTS(10) = INCLINATION, I                         00000650
C                       ELMNTS(11) = ARGUMENT OF PERIAPSIS                  00000660
C                       ELMNTS(12) = LONGITUDE OF THE ASCENDING NODE        00000670
C                       ELMNTS(13) = RETROGRADE FACTOR                      00000680
C                       ELMNTS(14) = MAGNITUDE OF POSITION VECTOR           00000690
C                       ELMNTS(15) = MAGNITUDE OF VELOCITY VECTOR           00000700
C                       ELMNTS(16) = Z-COMPONENT OF ANGULAR MOMENTUM        00000710
C                       ELMNTS(17) = PERIFOCAL RADIUS                       00000720
C                       ELMNTS(18) = APOFOCAL RADIUS                        00000730
C                       ELMNTS(19) = PERIOD                                 00000740
C                       ELMNTS(20) = ENERGY (OSCULATING)                    00000750
C                       ELMNTS(21) = ENERGY (ZONAL PERTURBED)               00000760
C                       ELMNTS(22) = ECCENTRIC LONGITUDE AT EPOCH           00000770
C                       ELMNTS(23) = MEAN ANOMALY AT EPOCH                  00000780
C                       ELMNTS(24) = TRUE LONGITUDE AT EPOCH                00000790
C                       ELMNTS(25) = TIME OF PERIAPSIS PASSAGE              00000800
C                       ELMNTS(26) = MAGNITUDE OF ANGULAR MOMENTUM          00000810
C                                                                           00000820
C     EXTERNAL REFERENCES                                                   00000830
C         NONE                                                              00000840
C                                                                           00000850
C     ANALYSIS                                                              00000860
C         CAPTAIN JEFFREY S. SHAVER, USAF                                   00000870
C                                                                           00000880
C     PROGRAMMER                                                            00000890
C         CAPTAIN JEFFREY S. SHAVER, USAF                                   00000900
C                                                                           00000910
      IMPLICIT REAL*8(A-H,O-Z)                                              00000920
C                                                                           00000930
C * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * * * 00000940
C                                                                           00000950
      DIMENSION     RV(6)     ,     ELMNTS(26)                              00000960
      PI   = 3.14159265358979320                                            00000970
      DTR  = 180.D0/PI                                                      00000980
      REQ  = 0.63781400D04                                                  00000990
      SHJ2 = -0.108265D-02                                                  00001000
      R0 = DSQRT(RV(1)**2 + RV(2)**2 + RV(3)**2)                            00001010
      ELMNTS(14) = R0                                                       00001020
      V0 = DSQRT(RV(4)**2 + RV(5)**2 + RV(6)**2)                            00001030
      ELMNTS(15) = V0                                                       00001040
      RDOTV = RV(1)*RV(4) + RV(2)*RV(5) + RV(3)*RV(6)                       00001050
C                                                                           00001060
C     COMPUTE ANGULAR MOMENTUM VECTOR COMPONENTS AND MAGNITUDE              00001070
C                                                                           00001080
      HX = RV(2)*RV(6) - RV(3)*RV(5)                                        00001090
      HY = RV(3)*RV(4) - RV(1)*RV(6)                                        00001100
      HZ = RV(1)*RV(5) - RV(2)*RV(4)                                        00001110
      ELMNTS(16) = HZ                                                       00001120
      ELMNTS(13) = DSQRT(HX**2 + HY**2 + HZ**2)                             00001130
C                                                                           00001140
```

```
C     COMPUTE ECCENTRICITY VECTOR COMPONENTS AND MAGNITUDE                00001150
C                                                                         00001160
      EX = (RV(5)*HZ - RV(6)*HY)/GM - RV(1)/R0                            00001170
      EY = (RV(6)*HX - RV(4)*HZ)/GM - RV(2)/R0                            00001180
      EZ = (RV(4)*HY - RV(5)*HX)/GM - RV(3)/R0                            00001190
      ELMNTS(2) = DSQRT(EX**2 + EY**2 + EZ**2)                            00001200
C                                                                         00001210
C     COMPUTE SEMI-LATUS RECTUM, SEMIMAJOR AXIS, MEAN MOTION              00001220
C                                                                         00001230
      ELMNTS(8) = ELMNTS(13)**2/GM                                        00001240
      ELMNTS(1) = ELMNTS(8)/(1.D0-ELMNTS(2)**2)                           00001250
      ELMNTS(7) = DSQRT(GM/ELMNTS(1) ** 3)                                00001260
C                                                                         00001270
C     COMPUTE PERIFOCAL AND APOFOCAL RADII, PERIOD, ENERGY (OSCULATING)   00001280
C                                                                         00001290
      ELMNTS(17) = ELMNTS(1) * (1.D0 - ELMNTS(2))                         00001300
      ELMNTS(18) = ELMNTS(1) * (1.D0 + ELMNTS(2))                         00001310
      ELMNTS(19) = (2.D0 * PI/ELMNTS(7))/3.6D03                           00001320
      ELMNTS(20) = -GM/(2.D0 * ELMNTS(1))                                 00001330
C                                                                         00001340
C     COMPUTE ENERGY (J2 PERTURBED):  CONSTANT OF MOTION IN J2 PROBLEM    00001350
C                                                                         00001360
      P2 = (3.D0 * (RV(3)/R0)**2 - 1.D0)/2.D0                             00001370
      C0 = (GM * REQ**2 * SHJ2)/R0**3                                     00001380
      ELMNTS(21) = ELMNTS(20) - C0 * P2                                   00001390
C                                                                         00001400
C     COMPUTE THE ORBITAL INCLINATION                                     00001410
C                                                                         00001420
      ELMNTS(3) = DACOS(HZ/ELMNTS(13))                                    00001430
      IF (ISET.EQ.2)  GO TO 50                                            00001440
C                                                                         00001450
C     COMPUTE UNIT VECTOR IN NODAL DIRECTION, IF NODE IS DEFINED          00001460
C                                                                         00001470
      IF (ELMNTS(3).LT.1.D-06)  GO TO 20                                  00001480
      UNITN  = DSQRT(HY**2 + HX**2)                                       00001490
      UNITNX = -HY/UNITN                                                  00001500
      UNITNY =  HX/UNITN                                                  00001510
C                                                                         00001520
C     COMPUTE LONGITUDE OF ASCENDING NODE, IF INCLINATION IS NONZERO      00001530
C                                                                         00001540
      ELMNTS(5) = DACOS(UNITNX)                                           00001550
      IF (UNITNY.LT.0.D0) ELMNTS(5) = 2.D0 * PI - ELMNTS(5)               00001560
C                                                                         00001570
C     COMPUTE ARGUMENT OF PERIAPSIS, I.NE.0 AND E.NE.0                    00001580
C                                                                         00001590
      IF (ELMNTS(2).LT.1.D-06)  GO TO 10                                  00001600
      ELMNTS(4) = DACOS((UNITNX*EX + UNITNY*EY)/ELMNTS(2))                00001610
      IF (EZ.LT.0.D0) ELMNTS(4) = 2.D0 * PI - ELMNTS(4)                   00001620
C                                                                         00001630
C     COMPUTE LONGITUDE OF PERIAPSIS, I.NE.0 AND E.NE.0                   00001640
C                                                                         00001650
      ELMNTS(9) = ELMNTS(4) + ELMNTS(5)                                   00001660
C                                                                         00001670
C     COMPUTE TRUE ANOMALY AT EPOCH, I.NE.0 AND E.NE.0                    00001680
C                                                                         00001690
      ELMNTS(10) = DACOS((EX*RV(1)+EY*RV(2)+EZ*RV(3))/(R0*ELMNTS(2)))    00001700
      IF (RDOTV.LT.0.D0) ELMNTS(10) = 2.D0 * PI - ELMNTS(10)              00001710
```

```
C                                                                       00001720
C     COMPUTE ARGUMENT OF LATITUDE, TRUE LONGITUDE AT EPOCH             00001730
C                                                                       00001740
      ELMNTS(11) = ELMNTS(4) + ELMNTS(10)                               00001750
      ELMNTS(12) = ELMNTS(5) + ELMNTS(11)                               00001760
      GO TO 40                                                          00001770
C                                                                       00001780
C     BEGIN COMPUTATIONS FOR INCLINATION.NE.0 BUT ECCENTRICITY.EQ.0     00001790
C                                                                       00001800
   10 ELMNTS(4)  = 0.D0                                                 00001810
      ELMNTS(9)  = ELMNTS(5)                                            00001820
      ELMNTS(11) = DACOS((UNITNX*RV(1) + UNITNY*RV(2))/R0)              00001830
      IF (RV(3).LT.0.D0)  ELMNTS(11) = 2.D0 * PI - ELMNTS(11)           00001840
      ELMNTS(10) = ELMNTS(11)                                           00001850
      ELMNTS(12) = ELMNTS(5) + ELMNTS(11)                               00001860
      GO TO 40                                                          00001870
C                                                                       00001880
C     BEGIN COMPUTATIONS FOR INCLINATION.EQ.0 BUT ECCENTRICITY.NE.0     00001890
C                                                                       00001900
   20 IF (ELMNTS(2).LT.1.D-06)  GO TO 30                                00001910
      ELMNTS(5)  = 0.D0                                                 00001920
      ELMNTS(4)  = DACOS(EX/ELMNTS(2))                                  00001930
      IF (EY.LT.0.D0)  ELMNTS(4) = 2.D0 * PI - ELMNTS(4)                00001940
      ELMNTS(9)  = ELMNTS(4)                                            00001950
      ELMNTS(10) = DACOS((EX*RV(1)+EY*RV(2)+EZ*RV(3))/(R0*ELMNTS(2)))   00001960
      IF (RDOTV.LT.0.D0)  ELMNTS(10) = 2.D0 * PI - ELMNTS(10)           00001970
      ELMNTS(11) = ELMNTS(4) + ELMNTS(10)                               00001980
      ELMNTS(12) = ELMNTS(11)                                           00001990
      GO TO 40                                                          00002000
C                                                                       00002010
C     BEGIN COMPUTATIONS FOR INCLINATION.EQ.0 AND ECCENTRICITY.EQ.0     00002020
C                                                                       00002030
   30 ELMNTS(4)  = 0.D0                                                 00002040
      ELMNTS(5)  = 0.D0                                                 00002050
      ELMNTS(9)  = 0.D0                                                 00002060
      ELMNTS(12) = DACOS(RV(1)/R0)                                      00002070
      IF (RV(2).LT.0.D0)  ELMNTS(12) = 2.D0 * PI - ELMNTS(12)           00002080
      ELMNTS(10) = ELMNTS(12)                                           00002090
      ELMNTS(11) = ELMNTS(12)                                           00002100
C                                                                       00002110
C     COMPUTE TIME OF PERIAPSIS PASSAGE                                 00002120
C                                                                       00002130
   40 CE0 = (ELMNTS(2) + DCOS(ELMNTS(10)))/                             00002140
     1                        (1.D0 + ELMNTS(2) * DCOS(ELMNTS(10)))     00002150
      SE0 = DSQRT(1.D0 - ELMNTS(2)**2) * DSIN(ELMNTS(10))/              00002160
     1                        (1.D0 + ELMNTS(2) * DCOS(ELMNTS(10)))     00002170
      E0  = DATAN2(SE0,CE0)                                             00002180
      IF (E0.LT.0.D0) E0 = 2.D0 * PI + E0                               00002190
      ELMNTS(22) = E0                                                   00002200
      EPM = E0 - ELMNTS(2) * DSIN(E0)                                   00002210
      IF (EPM.GE.2.D0 * PI)  EPM = EPM - 2.D0 * PI                      00002220
      ELMNTS(23) = EPM                                                  00002230
      ELMNTS(6) = (ELMNTS(7) * T0 - EPM)/ELMNTS(7)                      00002240
C                                                                       00002250
C     COMPUTE RIGHT ASCENSION, DECLINATION, MEAN LONGITUDE              00002260
C                                                                       00002270
      ELMNTS(24) = DACOS(RV(1)/DSQRT(RV(1)**2 + RV(2)**2))              00002280
```

```
      IF (RV(2).LT.0.D0)  ELMNTS(24) = 2.D0 * PI - ELMNTS(24)           00002290
      ELMNTS(25) = DARSIN(RV(3)/R0)                                     00002300
      ELMNTS(26) = ELMNTS(9) + ELMNTS(23)                               00002310
      IF  (ELMNTS(9).GE.2.D0*PI)  ELMNTS(9)  = ELMNTS(9)  - 2.D0 * PI   00002320
      IF (ELMNTS(11).GE.2.D0*PI)  ELMNTS(11) = ELMNTS(11) - 2.D0 * PI   00002330
      IF (ELMNTS(12).GE.2.D0*PI)  ELMNTS(12) = ELMNTS(12) - 2.D0 * PI   00002340
      IF (ELMNTS(26).GE.2.D0*PI)  ELMNTS(26) = ELMNTS(26) - 2.D0 * PI   00002350
      GO TO 100                                                         00002360
C                                                                       00002370
C     IF ISET=2, COMPUTE THE EQUINOCTIAL ELEMENTS                       00002380
C     COMPUTE UNIT VECTOR W                                             00002390
C                                                                       00002400
   50 ELMNTS(26) = ELMNTS(13)                                           00002410
      ELMNTS(9)  = ELMNTS(2)                                            00002420
      ELMNTS(10) = ELMNTS(3)                                            00002430
      WX = HX/ELMNTS(13)                                                00002440
      WY = HY/ELMNTS(13)                                                00002450
      WZ = HZ/ELMNTS(13)                                                00002460
C                                                                       00002470
C     COMPUTE RETROGRADE FACTOR                                         00002480
C                                                                       00002490
      IF (ELMNTS(3).GE.0.D0.AND.ELMNTS(3).LE.PI/2.D0) ELMNTS(13) = 1.D0 00002500
      IF (ELMNTS(3).GT.PI/2.D0.AND.ELMNTS(3).LE.PI)   ELMNTS(13) =-1.D0 00002510
C                                                                       00002520
C     COMPUTE EQUINOCTIAL ELEMENTS P AND Q                              00002530
C                                                                       00002540
      ELMNTS(4) =  WX/(1.D0 + ELMNTS(13) * WZ)                          00002550
      ELMNTS(5) = -WY/(1.D0 + ELMNTS(13) * WZ)                          00002560
C                                                                       00002570
C     COMPUTE UNIT VECTORS F AND G                                      00002580
C                                                                       00002590
      PQ = 1.D0 + ELMNTS(4)**2 + ELMNTS(5)**2                           00002600
      FX = (1.D0 - ELMNTS(4)**2 + ELMNTS(5)**2)/PQ                      00002610
      FY = 2.D0 * ELMNTS(4) * ELMNTS(5)/PQ                              00002620
      FZ = -2.D0 * ELMNTS(4) * ELMNTS(13)/PQ                            00002630
      GX = 2.D0 * ELMNTS(4) * ELMNTS(5) * ELMNTS(13)/PQ                 00002640
      GY = (1.D0 + ELMNTS(4)**2 - ELMNTS(5)**2) * ELMNTS(13)/PQ         00002650
      GZ = 2.D0 * ELMNTS(5)/PQ                                          00002660
C                                                                       00002670
C     COMPUTE EQUINOCTIAL ELEMENTS H AND K                              00002680
C                                                                       00002690
      ELMNTS(2) = EX*GX + EY*GY + EZ*GZ                                 00002700
      ELMNTS(3) = EX*FX + EY*FY + EZ*FZ                                 00002710
C                                                                       00002720
C     COMPUTE AUXILIARY QUANTITIES X, Y, BETA, A*SQRT(1-H**2-K**2)      00002730
C                                                                       00002740
      X    = RV(1)*FX + RV(2)*FY + RV(3)*FZ                             00002750
      Y    = RV(1)*GX + RV(2)*GY + RV(3)*GZ                             00002760
      BETA = 1.D0/(1.D0+DSQRT(1.D0-ELMNTS(2)**2-ELMNTS(3)**2))          00002770
      AHK  = ELMNTS(1) * DSQRT(1.D0-ELMNTS(2)**2-ELMNTS(3)**2)          00002780
C                                                                       00002790
C     COMPUTE COS(F), SIN(F), F, LAMDA                                  00002800
C                                                                       00002810
      CSF = ELMNTS(3) + ((1.D0+BETA*ELMNTS(3)**2)*X-ELMNTS(2)*ELMNTS(3) 00002820
     1                  *BETA*Y)/AHK                                    00002830
      SNF = ELMNTS(2) + ((1.D0-BETA*ELMNTS(2)**2)*Y-ELMNTS(2)*ELMNTS(3) 00002840
     1                  *BETA*X)/AHK                                    00002850
```

```
      F = DATAN2(SNF,CSF)                                                    00002860
      IF (F.LT.0.D0)  F = 2.D0 * PI + F                                      00002870
      ELMNTS(6)  = F - ELMNTS(3) * SNF + ELMNTS(2) * CSF                     00002880
      ELMNTS(22) = F                                                         00002890
C                                                                            00002900
C     COMPUTE ARGUMENT OF PERIAPSIS, LONGITUDE OF THE ASCENDING NODE         00002910
C                                                                            00002920
      ELMNTS(12) = DATAN2(ELMNTS(4),ELMNTS(5))                               00002930
      IF (ELMNTS(12).LT.0.D0)       ELMNTS(12) = 2.D0 * PI + ELMNTS(12)      00002940
      ARCHK      = DATAN2(ELMNTS(2),ELMNTS(3))                               00002950
      IF (ARCHK.LT.0.D0)            ARCHK      = 2.D0 * PI + ARCHK           00002960
      ELMNTS(11) = ARCHK - ELMNTS(13) * ELMNTS(12)                           00002970
C                                                                            00002980
C     COMPUTE MEAN ANOMALY AT EPOCH, TIME OF PERIAPSIS PASSAGE               00002990
C                                                                            00003000
      EPM = ELMNTS(6) - ELMNTS(11) - ELMNTS(13) * ELMNTS(12)                 00003010
      IF (EPM.GE.2.D0 * PI)             EPM = EPM - 2.D0 * PI                00003020
      IF (EPM.LT.0.D0)                  EPM = 2.D0 * PI + EPM                00003030
      ELMNTS(23) = EPM                                                       00003040
      ELMNTS(25) = (ELMNTS(7) * T0 - EPM)/ELMNTS(7)                          00003050
C                                                                            00003060
C     COMPUTE TRUE LONGITUDE AT EPOCH                                        00003070
C                                                                            00003080
      EPE = ELMNTS(22) - ELMNTS(11) - ELMNTS(13) * ELMNTS(12)                00003090
      CF0 = (DCOS(EPE) - ELMNTS(9))/(1.D0 - ELMNTS(9) * DCOS(EPE))           00003100
      SF0 = (DSQRT(1.D0-ELMNTS(9)**2)*DSIN(EPE))/                            00003110
     1                                (1.D0-ELMNTS(9)*DCOS(EPE))             00003120
      EPF = DATAN2(SF0,CF0)                                                  00003130
      IF (EPF.LT.0.D0)                  EPF = 2.D0 * PI + EPF                00003140
      ELMNTS(24) = EPF + ELMNTS(11) + ELMNTS(13) * ELMNTS(12)                00003150
      NO2PI = ELMNTS(24)/(2.D0 * PI)                                         00003160
      ELMNTS(24) = ELMNTS(24) - (DFLOAT(NO2PI) * 2.D0 * PI)                  00003170
C                                                                            00003180
C     OUTPUT RESULTS FOR KEPLERIAN CASE                                      00003190
C                                                                            00003200
  100 PRINT 900,T0,RV(1),RV(4),RV(2),RV(5),RV(3),RV(6)                       00003210
      IF (ISET.EQ.2)  GO TO 110                                              00003220
      EL3DEG  = ELMNTS(3) * DTR                                              00003230
      EL4DEG  = ELMNTS(4) * DTR                                              00003240
      EL5DEG  = ELMNTS(5) * DTR                                              00003250
      EL7DEG  = ELMNTS(7) * DTR                                              00003260
      EL9DEG  = ELMNTS(9) * DTR                                              00003270
      EL10DG  = ELMNTS(10)* DTR                                              00003280
      EL11DG  = ELMNTS(11)* DTR                                              00003290
      EL12DG  = ELMNTS(12)* DTR                                              00003300
      EL22DG  = ELMNTS(22)* DTR                                              00003310
      EL23DG  = ELMNTS(23)* DTR                                              00003320
      EL24DG  = ELMNTS(24)* DTR                                              00003330
      EL25DG  = ELMNTS(25)* DTR                                              00003340
      EL26DG  = ELMNTS(26)* DTR                                              00003350
      PRINT 905, ELMNTS(1)  , ELMNTS(2)  , EL3DEG    ,                       00003360
     1           EL4DEG     , EL5DEG     , EL23DG                            00003370
      PRINT 910, ELMNTS(6)  , EL10DG     , EL7DEG    ,                       00003380
     1           EL22DG     , ELMNTS(19) , EL24DG    ,                       00003390
     2           ELMNTS(17) , EL25DG     , ELMNTS(18),  EL12DG              00003400
      PRINT 911, ELMNTS(8)  , EL26DG     , ELMNTS(13),                       00003410
     1           EL9DEG     , ELMNTS(16) , EL11DG    ,                       00003420
```

```
     2           ELMNTS(20) ,  ELMNTS(14) ,  ELMNTS(21),  ELMNTS(15)    00003430
      RETURN                                                            00003440
C                                                                       00003450
C     OUTPUT RESULTS FOR EQUINOCTIAL CASE                               00003460
C                                                                       00003470
  110 EL6DEG  = ELMNTS(6) * DTR                                         00003480
      EL7DEG  = ELMNTS(7) * DTR                                         00003490
      EL10DG  = ELMNTS(10)* DTR                                         00003500
      EL11DG  = ELMNTS(11)* DTR                                         00003510
      EL12DG  = ELMNTS(12)* DTR                                         00003520
      EL22DG  = ELMNTS(22)* DTR                                         00003530
      EL23DG  = ELMNTS(23)* DTR                                         00003540
      EL24DG  = ELMNTS(24)* DTR                                         00003550
      IF (ELMNTS(13).GT.0.D0)                                           00003560
     1PRINT 915, ELMNTS(1)  ,  ELMNTS(2)  ,  ELMNTS(3)  ,               00003570
     2           ELMNTS(4)  ,  ELMNTS(5)  ,  EL6DEG                     00003580
      IF (ELMNTS(13).LT.0.D0)                                           00003590
     1PRINT 920, ELMNTS(1)  ,  ELMNTS(2)  ,  ELMNTS(3)  ,               00003600
     2           ELMNTS(4)  ,  ELMNTS(5)  ,  EL6DEG                     00003610
      PRINT 925, ELMNTS(25) ,  EL23DG     ,  EL7DEG     ,               00003620
     1           ELMNTS(9)  ,  ELMNTS(19) ,  EL10DG     ,               00003630
     2           ELMNTS(17) ,  ELMNTS(13) ,  ELMNTS(18) ,               00003640
     3           EL24DG     ,  ELMNTS(8)  ,  EL22DG     ,               00003650
     4           ELMNTS(26) ,  EL12DG     ,  ELMNTS(16) ,               00003660
     5           EL11DG     ,  ELMNTS(20) ,  ELMNTS(14) ,               00003670
     6           ELMNTS(21) ,  ELMNTS(15)                               00003680
      RETURN                                                            00003690
C                                                                       00003700
C               FORMAT STATEMENTS                                       00003710
C                                                                       00003720
  900 FORMAT(52X,'INITIAL STATE ELEMENT REPORT'//10X,'INITIAL TIME =',  00003730
     1   1PD25.15//10X,'INITIAL POSITION VECTOR',12X,'INITIAL VELOCITY',00003740
     2        ' VECTOR'/10X,'RX =',1PD25.15,3X,'VX =',1PD25.15/10X,     00003750
     3       'RY =',1PD25.15,3X,'VY =',1PD25.15/10X,'RZ =',1PD25.15,3X,00003760
     4       'VZ =',1PD25.15//)                                         00003770
C                                                                       00003780
  905 FORMAT(10X,'THE KEPLERIAN ELEMENT SET ASSOCIATED WITH THIS INITI',00003790
     1         'AL STATE IS:'/10X,'SEMIMAJOR AXIS, A',14X,':',1PD25.15/ 00003800
     2       10X,'ECCENTRICITY, E',16X,':',1PD25.15/10X,'INCLINATION, ',00003810
     3           'I',17X,':',1PD25.15/10X,'ARGUMENT OF PERIAPSIS',10X,  00003820
     4           ':',1PD25.15/10X,'LONGITUDE OF THE ASCENDING NODE:',   00003830
     5           1PD25.15/10X,'MEAN ANOMALY AT EPOCH',10X,':',          00003840
     6           1PD25.15//)                                            00003850
C                                                                       00003860
  910 FORMAT(10X,'THE FOLLOWING AUXILIARY QUANTITIES ARE ALSO FOUND:'/  00003870
     1    10X,'TIME OF PERIAPSIS PASSAGE',6X,':',1PD25.15,5X,'TRUE ANO',00003880
     2    'MALY AT EPOCH',10X,':',1PD25.15/10X,'MEAN MOTION',20X,':',   00003890
     3    1PD25.15,5X,'ECCENTRIC ANOMALY AT EPOCH',5X,':',1PD25.15/10X, 00003900
     4    'PERIOD',25X,':',1PD25.15,5X,'RIGHT ASCENSION',16X,':',       00003910
     5    1PD25.15/10X,'PERIFOCAL RADIUS',15X,':',1PD25.15,5X,'DECLINA',00003920
     6    'TION',20X,':',1PD25.15/10X,'APOFOCAL RADIUS',16X,':',        00003930
     7    1PD25.15,5X,'TRUE LONGITUDE AT EPOCH',8X,':',1PD25.15)        00003940
C                                                                       00003950
  911 FORMAT(10X,'SEMI-LATUS RECTUM (PARAMETER)  :',1PD25.15,5X,        00003960
     1    'MEAN LONGITUDE AT EPOCH',8X,':',1PD25.15/10X,'ANGULAR MOMEN',00003970
     2    'TUM MAGNITUDE',5X,':',1PD25.15,5X,'LONGITUDE OF PERIAPSIS',  00003980
     3    9X,':',1PD25.15/10X,'Z-COMPONENT OF ANGULAR MOMENTUM:',       00003990
```

```
     4    1PD25.15,5X,'ARGUMENT OF LATITUDE AT EPOCH',2X,':',1PD25.15/  00004000
     5    10X,'ENERGY (OSCULATING)',12X,':',1PD25.15,5X,'MAGNITUDE OF ',00004010
     6    'POSITION VECTOR   :',1PD25.15/10X,'ENERGY (J2 PERTURBED)',   00004020
     7    10X,':',1PD25.15,5X,'MAGNITUDE OF VELOCITY VECTOR   :',       00004030
     8    1PD25.15///)                                                  00004040
C                                                                       00004050
  915 FORMAT(10X,'THE DIRECT EQUINOCTIAL ELEMENT SET ASSOCIATED WITH ', 00004060
     1           'THIS INITIAL STATE IS:'/10X,'A',5X,':',1PD25.15/10X,  00004070
     2           'H',5X,':',1PD25.15/10X,'K',5X,':',1PD25.15/10X,       00004080
     3           'P',5X,':',1PD25.15/10X,'Q',5X,':',1PD25.15/10X,       00004090
     4           'LAMDA :',1PD25.15//)                                  00004100
C                                                                       00004110
  920 FORMAT(10X,'THE RETROGRADE EQUINOCTIAL ELEMENT SET ASSOCIATED ',  00004120
     1           'WITH THIS INITIAL STATE IS:'/10X,'A',5X,':',1PD25.15/ 00004130
     2       10X,'H',5X,':',1PD25.15/10X,'K',5X,':',1PD25.15/10X,       00004140
     3           'P',5X,':',1PD25.15/10X,'Q',5X,':',1PD25.15/10X,       00004150
     4           'LAMDA :',1PD25.15//)                                  00004160
C                                                                       00004170
  925 FORMAT(10X,'THE FOLLOWING AUXILIARY QUANTITIES ARE ALSO FOUND:'/  00004180
     1    10X,'TIME OF PERIAPSIS PASSAGE',6X,':',1PD25.15,5X,'MEAN ANO',00004190
     2    'MALY AT EPOCH',10X,':',1PD25.15/10X,'MEAN MOTION',20X,':',   00004200
     3    1PD25.15,5X,'ECCENTRICITY, E           ',5X,':',1PD25.15/10X, 00004210
     4    'PERIOD',25X,':',1PD25.15,5X,'INCLINATION, I ',16X,':',       00004220
     5    1PD25.15/10X,'PERIFOCAL RADIUS',15X,':',1PD25.15,5X,'RETROGR',00004230
     6    'ADE FACTOR',14X,':',1PD25.15/10X,'APOFOCAL RADIUS',16X,':',  00004240
     7    1PD25.15,5X,'TRUE LONGITUDE AT EPOCH',8X,':',1PD25.15/        00004250
     8    10X,'SEMI-LATUS RECTUM (PARAMETER)  :',1PD25.15,5X,'ECCEN',   00004260
     9    'TRIC LONGITUDE AT EPOCH   :',1PD25.15/10X,'ANGULAR MOMENTUM',00004270
     *    ' MAGNITUDE',5X,':',1PD25.15,5X,'LONGITUDE OF THE ASCENDING ',00004280
     *    'NODE:',1PD25.15/10X,'Z-COMPONENT OF ANGULAR MOMENTUM:',      00004290
     *    1PD25.15,5X,'ARGUMENT OF PERIAPSIS        ',2X,':',1PD25.15/  00004300
     *    10X,'ENERGY (OSCULATING)',12X,':',1PD25.15,5X,'MAGNITUDE OF ',00004310
     *    'POSITION VECTOR   :',1PD25.15/10X,'ENERGY (J2 PERTURBED)',   00004320
     *    10X,':',1PD25.15,5X,'MAGNITUDE OF VELOCITY VECTOR   :',       00004330
     *    1PD25.15///)                                                  00004340
C                                                                       00004350
      END                                                               00004360
```

******** END OF MEMBER EPELMS 436 RECORDS ********

EQUMTN JSS1493.GTDS.UPDATE.FORT

```
      SUBROUTINE EQUMTN(OPSEUD)                                          00000010
C     ...................................................................00000020
C     VERSION 2.3                                                        .00000030
C     .     FORTRAN SUBROUTINE FOR IBM 360                               .00000040
C     .  PURPOSE                                                         .00000050
C     .     TO COMPUTE THE TIME DERIVATIVES OF THE ORBITAL PARAMETERS     .00000060
C     .  CALLING SEQUENCE                                                .00000070
C     .     CALL EQUMTN(OPSEUD)                                          .00000080
C     .        WHERE OPSEUD = LOGICAL VARIABLE (=.FALSE. FOR A FULL      .00000090
C     .                       FORCE EVALUATION, =.TRUE. FOR A PARTIAL    .00000100
C     .                       EVALUATION)                                .00000110
C     .                                                                  .00000120
C     .                                                                  .00000130
C     .  INPUT                                                           .00000140
C     .     VIA BLOCK COMMON                                             .00000150
C     .      VARIABLES IN COMMON SWITCH                                  .00000160
C     .      IND(3)=VOP TYPE(=1 FOR HERRICK, =2 FOR EQUINOCTIAL WITH     .00000170
C     .             MEAN LONGITUDE, =3 FOR EQUINOCTIAL WITH ECCENTRIC    .00000180
C     .             LONGITUDE, =4 FOR IDEAL WITH IDEAL TRUE LONGITUDE    .00000190
C     .             =5 FOR IDEAL WITH IDEAL ECCENTRIC LONGITUDE,         .00000200
C     .             =6 FOR KS ELEMENTS, =12 FOR NUMERICAL AVERAGING)     .00000210
C     .      VARIABLES IN COMMON WORKER                                  .00000220
C     .      VPELMT  = ORBITAL ELEMENTS                                  .00000230
C     .      RETRG   = RETROGRADE FACTOR (+1= DIRECT,-1= RETROGRADE)     .00000240
C     .      K       = INDEX OF CURRENT VALUE IN THE DERIVATIVE ARRAY    .00000250
C     .      T       = TIME FROM EPOCH                                   .00000260
C     .      VARIABLES IN COMMON TIMREG                                  .00000270
C     .      STREG   = CURRENT VALUE OF INDEPENDENT VARIABLE             .00000280
C     .  OUTPUT                                                          .00000290
C     .     VIA BLOCK COMMON                                             .00000300
C     .      VARIABLES IN COMMON WORKER                                  .00000310
C     .      X       = POSITION VECTOR IN INERTIAL FRAME                 .00000320
C     .      XD      = VELOCITY VECTOR IN INERTIAL FRAME                 .00000330
C     .      DPARDV  = PARTIAL  DERIVATIVES OF EQUINOCTIAL ELEMENTS      .00000340
C     .      DPARRY  = ARRAY CONTAINING TIME DERIVATIVES                 .00000350
C     .      VARIABLES IN COMMON FRC                                     .00000360
C     .      ANOF    = PERTURBING ACCELERATION                           .00000370
C     .                                                                  .00000380
C     .   EXTERNAL REFERENCES                                            .00000390
C     .      DALLPV                                                      .00000400
C     .      PARDT                                                       .00000410
C     .      ACCEL                                                       .00000420
C     .      EQUNPV                                                      .00000430
C     .      PARTEQ                                                      .00000440
C     .      MATHUL                                                      .00000450
C     .      IDLPV                                                       .00000460
C     .      IPART                                                       .00000470
C     .      AVRAGE                                                      .00000480
C     .      USER                                                        .00000490
C     .      DPEL2X                                                      .00000500
C     .      FUNCTE                                                      .00000510
C     .      CELEM                                                       .00000520
C     .      KPART                                                       .00000530
C     .                                                                  .00000540
C     .   REFERENCE                                                      .00000550
C     .      GTDS SUBROUTINE DESCRIPTION EQUMTN                          .00000560
C     .   METHOD                                                         .00000570
```

```
C     .      METHOD IS OUTLINED IN REFERENCE                                .00000580
C     .   ANALYSIS                                                          .00000590
C     .      P.J. CEFOLA,K.S.NINITZ,A.C.LONG                                .00000600
C     .       COMPUTER SCIENCES CORPORATION.                                 00000610
C     .   PROGRAMMER                                                        .00000620
C     .      A.C.LONG                                                       .00000630
C     .                                                                     .00000640
C     .   PROGRAM MODIFICATIONS                                             .00000650
C     .      DATE:    06/01/72                                             .00000660
C     .      NAME:    A. C. LONG, CSC                                      .00000670
C     .      REASON:  ORIGINAL SUBROUTINE 'EQUMTN'                         .00000680
C     .      DATE:    07/15/73                                             .00000690
C     .      NAME:    A. C. LONG, CSC                                      .00000700
C     .      REASON:  ADD COMMENTS                                         .00000710
C     .      DATE:    12/20/73                                             .00000720
C     .      NAME:    A. C. LONG                                           .00000730
C     .      REASON:  ALL LOGIC RELATING TO NUMERICAL AVERAGING WAS        .00000740
C     .               REMOVED AND REPLACED BY A CALL TO SUBROUTINE         .00000750
C     .               'AVRAGE'.                                            .00000760
C     .      DATE:    01/14/74                                             .00000770
C     .      NAME:    B. DIXON                                             .00000780
C     .      REASON:  TO INCLUDE THE J2 EFFECT FOR THE PSUEDO-EVALUATE     .00000790
C     .               FOR THE EQUINOCTIAL ELEMENTS ONLY.  THE J2 EFFECT.00000800
C     .               CAN BE IMPLEMENTED FOR OTHER INTEGRATORS BY           .00000810
C     .               ANALOGY.  IND(90) = 2 MEANS TO INCLUDE THE J2         .00000820
C     .               EFFECT FOR THE PSUEDO-EVALUATE FOR THE EQUI-          .00000830
C     .               NOCTIAL ELEMENTS.                                     .00000840
C     .      DATE:    06/01/74                                             .00000850
C     .      NAME:    JOHN TODD                                            .00000860
C     .      REASON:  TO UPDATE UTILIZATION OF J2 DUE TO INCREASE OF       .00000870
C     .               HARMONICS FIELD TO 21 X 21.                          .00000880
C     .      DATE:    06/15/74                                             .00000890
C     .      NAME:    R. PAJERSKI                                          .00000900
C     .      REASON:  ADDED NEW ELEMENT TYPE 8                             .00000910
C     .      DATE:    09/15/74                                             .00000920
C     .      NAME:    V. G. KAVEESHWAR AND J. M. MEYER                     .00000930
C     .      REASON:  REMOVE CALLS TO MATMUL, CODING MATRIX MULTI-         .00000940
C     .               PLICATION IN LINE FOR VOP TYPES 2 AND 3              .00000950
C     .      DATE:    06/30/79                                             .00000960
C     .      NAME:    CAPTAIN JEFFREY S. SHAVER, USAF                      .00000970
C     .      REASON:  ADD CALL TO 'EQUNPV' AFTER CALL TO 'AVRAGE' SO       .00000980
C     .               THE INTEGRATED POSITION AND VELOCITY ARE UPDATED..00000990
C     .                                                                    .00001000
C     ....................................................................00001010
C                                                                          00001020
C * * * * * * * * * *          START PROGRAM          * * * * * * * * * * *00001030
C                                                                          00001040
      IMPLICIT REAL*8(A-H,O-Z)                                             00001050
      LOGICAL        OPSEUD                                                00001060
C                                                                          00001070
C * * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * *00001080
C                                                                          00001090
      COMMON/CONST / THOPI                                                 00001100
C                                                                          00001110
      COMMON/FRC   / DPFRC(1300)  ,  INFRC(18)                             00001120
C                                                                          00001130
      COMMON/SWITCH/ IND(100)     ,  IDUMMY(100)                           00001140
```

```
C                                                                       00001150
      COMMON/TIMREG/ DTRG(94)                                           00001160
C                                                                       00001170
      COMMON/WORKER/ DPWORK(3264) ,  IWORK(12)                          00001180
C                                                                       00001190
      COMMON         DUM(120)     ,  PSAVE(6,6)                       00001200
C                                                                       00001210
C * * * * * * * * *      DIMENSION STATEMENTS      * * * * * * * * * *00001220
C                                                                       00001230
      DIMENSION      X(3)         ,  XD(3)         ,  VPELMT(10)   ,   00001240
     *               DPARDV(6,3)  ,  DPARRY(15,10) ,                    00001250
     *               DPARDT(10)   ,  ANCF(3)                            00001260
C                                                                       00001270
      DIMENSION      TREG(40)     ,  U(4)          ,  US(4)            00001280
C                                                                       00001290
      DIMENSION      PVSAVE(6)                                          00001300
C                                                                       00001310
      DIMENSION      POSI(3)      ,  VELI(3)       ,  SPHCOR(6)    ,   00001320
     *               ORBELS(20)                                         00001330
C                                                                       00001340
      DIMENSION      VDX(3)       ,  ANCFJ2(3)     ,  GM(11)       ,   00001350
     *               AB(11)       ,  CJ2NEG(2)                       00001360
C                                                                       00001370
C * * * * * * * * *      EQUIVALENCE STATEMENTS    * * * * * * * * * *00001380
C                                                                       00001390
C     FROM COMMON/FRC   /:                                              00001400
      EQUIVALENCE    (GM(1)       , DPFRC(2)    ),(ICENT  , INFRC(1)   ),00001410
     *               (CJ2NEG(1)   , DPFRC(859)  ),(AB(1)  , DPFRC(24)  ),00001420
     *               (ANCF(1)     , DPFRC(898)  )                       00001430
C                                                                       00001440
C     FROM COMMON/TIMREG/:                                              00001450
      EQUIVALENCE    (STREG       , DTRG(92)    ),(TREG(1), DTRG(1)    ) 00001460
C                                                                       00001470
C     FROM COMMON/WORKER/:                                              00001480
      EQUIVALENCE    (X(1)        , DPWORK(1)   ),(XD(1)  ,  DPWORK(4) ),00001490
     *               (DPARRY(1,1), DPWORK(7)   ),                       00001500
     *               (VPELMT(1)   , DPWORK(2829)),(RETRG  ,DPWORK(2773)),00001510
     *               (DPARDV(1,1), DPWORK(2774)),                       00001520
     *               (K           , IWORK(2)    ),                      00001530
     *               (XMEAN       , DPWORK(2814)),                      00001540
     *               (DPWORK(1)   , PVSAVE(1)   ),                      00001550
     *               (GI          , DPWORK(2839)),(R      ,DPWORK(3081)),00001560
     *               (NOPARM      , IWORK(3)    ),(T      ,DPWORK(3215)) 00001570
      DATA NERR/1/                                                      00001580
      XMU=GM(ICENT)                                                     00001590
      RB=AB(ICENT)                                                      00001600
      XJ2 = CJ2NEG(1)                                                   00001610
      NOUT = 6                                                          00001620
      IND3=IND(3)                                                       00001630
      T=STREG                                                           00001640
C                                                                       00001650
C     FIRST TEST TYPE OF VOP INDICATOR                                  00001660
C     BRANCH TO APPROPRIATE SECTION OF SUBROUTINE                       00001670
C                                                                       00001680
      IF(IND3.GT.10) GO TO 700                                          00001690
      GO TO (50,200,200,300,300,740,800,425),IND3                       00001700
C                                                                       00001710
```

```
C     CALCULATION OF POSITION AND VELOCITY FROM HERRICK ELEMENTS        00001720
C                                                                       00001730
  50  CALL DALLPV(VPELMT,X,XD,NERR)                                     00001740
      IF(NERR.EQ.13) CALL ERROUT(400)                                   00001750
      IF(OPSEUD) GO TO 100                                              00001760
C                                                                       00001770
C     CALCULATION OF PERTURBING ACCELERATION                            00001780
C                                                                       00001790
      CALL ACCEL(X,XD,T)                                                00001800
C                                                                       00001810
C     CALCULATION OF TIME DERIVATIVES                                   00001820
C                                                                       00001830
  100 CALL PARDT                                                        00001840
      GO TO 999                                                         00001850
C                                                                       00001860
C     CALCULATION OF POSITION AND VELOCITY FOR EQUINOCTIAL ELEMENTS     00001870
C                                                                       00001880
  200 CALL EQUNPV(VPELMT,X,XD,NERR,IND3)                                00001890
      IF(NERR.EQ.13) CALL ERROUT(400)                                   00001900
      IF(IND(90).EQ.2) GO TO 198                                        00001910
      VDX(1)=0.D0                                                       00001920
      VDX(2)=0.D0                                                       00001930
      VDX(3)=0.D0                                                       00001940
      GO TO 199                                                         00001950
  198 CONTINUE                                                          00001960
      R=  DSQRT(X(1)*X(1)+X(2)*X(2)+X(3)*X(3))                          00001970
      R2=R*R                                                            00001980
      R3=R2*R                                                           00001990
      R4=R3*R                                                           00002000
      R5=R4*R                                                           00002010
      R7=R5*R2                                                          00002020
      R5IN=1.0D0/R5                                                     00002030
      R7IN=1.0D0/R7                                                     00002040
      XLAM=1.5D0*XMU*XJ2*RB*RB                                          00002050
      X3SQ=X(3)*X(3)                                                    00002060
      VPOT=XLAM*(X3SQ*R5IN-1.0D0/(3.0D0*R3))                            00002070
      XLTB=XLAM*(-5.0D0*X3SQ*R7IN+R5IN)                                 00002080
      VDX(1)=X(1)*XLTB                                                  00002090
      VDX(2)=X(2)*XLTB                                                  00002100
      VDX(3)=X(3)*XLTB+XLAM*X(3)*2.0D0*R5IN                             00002110
  199 CONTINUE                                                          00002120
C                                                                       00002130
C     COMPUTE PARTIAL DERIVATIVES                                       00002140
C                                                                       00002150
      CALL PARTEQ(IND3)                                                 00002160
      IF(OPSEUD) GO TO 220                                              00002170
C                                                                       00002180
C     CALCULATION OF PERTURBING ACCELERATION                            00002190
C                                                                       00002200
      CALL ACCEL(X,XD,T)                                                00002210
      DO 201 I=1,3                                                      00002220
      ANCFJ2(I)=ANCF(I) +VDX(I)                                         00002230
  201 CONTINUE                                                          00002240
  220 CONTINUE                                                          00002250
      DO 221 I=1,3                                                      00002260
      ANCF(I)=ANCFJ2(I)-VDX(I)                                          00002270
  221 CONTINUE                                                          00002280
```

```
C                                                                       00002290
C     CALCULATION OF TIME DERIVATIVES                                   00002300
      DPARDT(1) = DPARDV(1,1) * ANCF(1) +                               00002310
     2            DPARDV(1,2) * ANCF(2) +                              00002320
     3            DPARDV(1,3) * ANCF(3)                                00002330
      DPARDT(2) = DPARDV(2,1) * ANCF(1) +                               00002340
     2            DPARDV(2,2) * ANCF(2) +                              00002350
     3            DPARDV(2,3) * ANCF(3)                                00002360
      DPARDT(3) = DPARDV(3,1) * ANCF(1) +                               00002370
     2            DPARDV(3,2) * ANCF(2) +                              00002380
     3            DPARDV(3,3) * ANCF(3)                                00002390
      DPARDT(4) = DPARDV(4,1) * ANCF(1) +                               00002400
     2            DPARDV(4,2) * ANCF(2) +                              00002410
     3            DPARDV(4,3) * ANCF(3)                                00002420
      DPARDT(5) = DPARDV(5,1) * ANCF(1) +                               00002430
     2            DPARDV(5,2) * ANCF(2) +                              00002440
     3            DPARDV(5,3) * ANCF(3)                                00002450
      DPARDT(6) = DPARDV(6,1) * ANCF(1) +                               00002460
     2            DPARDV(6,2) * ANCF(2) +                              00002470
     3            DPARDV(6,3) * ANCF(3)                                00002480
      CALL MATMUL(DPARDV,ANCF  ,DPARDT,6,3,1)                           00002490
      IF(IND3.EQ.2) DPARDT(6) = DPARDT(6) + XMEAN                       00002500
      IF(IND3.EQ.3) DPARDT(6) = DPARDT(6) + XMEAN*VPELMT(1)/R           00002510
      GO TO 500                                                         00002520
C                                                                       00002530
C     COMPUTE POSITION AND VELOCITY FROM IDEAL ELEMENTS                 00002540
C                                                                       00002550
  300 CALL IDLPV(VPELMT,X,XD,IND3)                                      00002560
C                                                                       00002570
C     COMPUTE PARTIAL DERIVATIVES                                       00002580
C                                                                       00002590
      CALL IPART(IND3)                                                  00002600
      IF(OPSEUD) GO TO 350                                              00002610
      CALL ACCEL(X,XD,T)                                                00002620
  350 CONTINUE                                                          00002630
C                                                                       00002640
C     FORM THE TIME DERIVATIVES                                         00002650
      DPARDT(1) = DPARDV(1,1) * ANCF(1) +                               00002660
     2            DPARDV(1,2) * ANCF(2) +                              00002670
     3            DPARDV(1,3) * ANCF(3)                                00002680
      DPARDT(2) = DPARDV(2,1) * ANCF(1) +                               00002690
     2            DPARDV(2,2) * ANCF(2) +                              00002700
     3            DPARDV(2,3) * ANCF(3)                                00002710
      DPARDT(3) = DPARDV(3,1) * ANCF(1) +                               00002720
     2            DPARDV(3,2) * ANCF(2) +                              00002730
     3            DPARDV(3,3) * ANCF(3)                                00002740
      DPARDT(4) = DPARDV(4,1) * ANCF(1) +                               00002750
     2            DPARDV(4,2) * ANCF(2) +                              00002760
     3            DPARDV(4,3) * ANCF(3)                                00002770
      DPARDT(5) = DPARDV(5,1) * ANCF(1) +                               00002780
     2            DPARDV(5,2) * ANCF(2) +                              00002790
     3            DPARDV(5,3) * ANCF(3)                                00002800
      DPARDT(6) = DPARDV(6,1) * ANCF(1) +                               00002810
     2            DPARDV(6,2) * ANCF(2) +                              00002820
     3            DPARDV(6,3) * ANCF(3)                                00002830
      CALL MATMUL(DPARDV,ANCF  ,DPARDT,6,3,1)                           00002840
      IF(IND3.EQ.5) GO TO 400                                           00002850
```

```
      DPARDT(7) = 1.000/(GI*R**2)                                       00002860
      GO TO 500                                                         00002870
  400 DPARDT(7) = XMEAN*VPELMT(1)/R                                     00002880
      GO TO 500                                                         00002890
C                                                                       00002900
  425 CONTINUE                                                          00002910
C                                                                       00002920
C     CALCULATION OF POSITION AND VELOCITY FOR KEPLERIAN ELEMENTS       00002930
C                                                                       00002940
      CALL CELEM(VPELMT(1),XMU,X,XD)                                    00002950
C                                                                       00002960
C     CALCULATION OF PARTIAL DERIVATIVES WRT VELOCITY                   00002970
C                                                                       00002980
      CALL KPART(PVSAVE,PSAVE)                                          00002990
      DO 428 I= 1,6                                                     00003000
      DO 428 J= 1,3                                                     00003010
      DPARDV(I,J)= PSAVE(I,J+3)                                         00003020
  428 CONTINUE                                                          00003030
      IF(OPSEUD) GO TO 430                                              00003040
C                                                                       00003050
C     CALCULATION OF PERTURBING ACCELERATIONS                           00003060
C                                                                       00003070
      CALL ACCEL(X,XD,T)                                                00003080
C                                                                       00003090
C     CALCULATION OF TIME DERIVATIVES                                   00003100
C                                                                       00003110
  430 CONTINUE                                                          00003120
      CALL MATMUL(DPARDV,ANCF,DPARDT,6,3,1)                             00003130
C     SINCE IND(3)= 8                                                   00003140
      XMEAN= DSQRT(GM(ICENT)/VPELMT(1)**3)                              00003150
      DPARDT(6)= DPARDT(6)+XMEAN                                        00003160
C                                                                       00003170
C     PLACE TIME DERIVATIVE IN APPROPRIATE ROW OF DERIVATIVE ARRAY      00003180
C                                                                       00003190
  500 DO 600 I=1,NOPARM                                                 00003200
      DPARRY(K,I)= DPARDT(I)                                            00003210
  600 CONTINUE                                                          00003220
      GO TO 999                                                         00003230
C                                                                       00003240
C     BRANCH FOR VOP/NUMERICAL AVERAGING                                00003250
C                                                                       00003260
  700 IF(OPSEUD) IND(24)= 2                                             00003270
      IF(.NOT.OPSEUD) IND(24)= 1                                        00003280
      CALL AVRAGE                                                       00003290
      IF (IND3.EQ.12)                                                   00003300
     *          CALL EQUNPV(VPELMT,X,XD,NERR,IND3)                     00003310
      IF (IND3.EQ.18)                                                   00003320
     *          CALL  CELEM(VPELMT(1),XMU,X,XD)                        00003330
      GO TO 999                                                         00003340
C                                                                       00003350
C     BRANCH FOR KS FORMULATION                                         00003360
C                                                                       00003370
  740 CALL DPEL2X(VPELMT,STREG,U,US,X,XD,SE2,CE2)                       00003380
      TREG(K)=T                                                         00003390
      IF(OPSEUD) GO TO 750                                              00003400
C                                                                       00003410
C      COMPUTE PERTURBING ACCELERATION                                  00003420
```

```
C                                                                       00003430
      CALL USER(X,1)                                                    00003440
C                                                                       00003450
C     FORM THE DERIVATIVES                                              00003460
C                                                                       00003470
  750 CALL FUNCTE(U,US,SE2,CE2)                                         00003480
      GO TO 999                                                         00003490
C                                                                       00003500
C      BRANCH FOR DS FORMULATUION                                       00003510
C                                                                       00003520
  800 CALL DSTOCE(VPELMT,STREG,X,XD)                                    00003530
      TREG(K)=T                                                         00003540
      IF(CPSEUD)  GO TO 850                                             00003550
C                                                                       00003560
C      COMPUTE PERTURBING ACCELERATION                                  00003570
C                                                                       00003580
      CALL PERFOR (X,XD,.FALSE.)                                        00003590
      GO TO 875                                                         00003600
C                                                                       00003610
C      FORM THE DERIVATIVES                                             00003620
C                                                                       00003630
  850 CALL PERFOR (X,XD,.TRUE.)                                         00003640
  875 CALL CAPEFO(X,STREG,VPELMT)                                       00003650
  999 RETURN                                                            00003660
 2000 FORMAT(15X,'ERROR KEPLER SOLUTION DOES NOT  CONVERGE')            00003670
 2001 FORMAT(1H ,25X,'KEPLER SOLUTION FOR ECCENTRIC LONGITUDE DOES NOT  00003680
     *       CONVERGE')                                                 00003690
      END                                                               00003700

******** END OF MEMBER EQUMTN     370 RECORDS ********
```

```
MULSTP              JSS1493.GTDS.UPDATE.FORT

      SUBROUTINE  MULSTP(IERR)                                          00000010
C                                                                       00000020
C     VERSION 2.2                                                       00000030
C             FORTRAN SUBRCUTINE FOR IBM 360                            00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE REQUIRED STARTING VALUES OF TIME DERIVATIVES OF00000060
C         PARAMETERS, AND SUMS REQUIRED BY THE ADAMS  INTEGRATION PRO-  00000070
C         CESS                                                          00000080
C     CALLING SEQUENCE                                                  00000090
C         CALL MULSTP(IERR)                                             00000100
C     CALLING SEQUENCE VARIABLES                                        00000110
C     OUTPUT                                                            00000120
C         IERR  = 1 - NORMAL END                                        00000130
C               = 3 - MAXIMUM NUMBER OF ITERATIONS EXCEEDED             00000140
C INPUT                                                                 00000150
C     VIA COMMON SWITCH                                                 00000160
C         IND(6) = 1 - TWO-BODY MOTION , 2 - NON TWO-BODY MOTION        00000170
C         IND(20)= 1 - USE PICARD-CHEBYSHEV TO INTEGRATE VOP EQUATIONS, 00000180
C                  2 - USE STANDARD MULTISTEP INTEGRATOR                00000190
C     VIA COMMON INTEG                                                  00000200
C         MAXIT(2)=    MAXIMUM NUMBER OF CORRECTOR ITERATIONS           00000210
C         NORDER =     NUMBER OF BACK VALUES USED IN THE INTEGRATOR     00000220
C     VIA COMMON WORKER                                                 00000230
C         VPELMT =     ORBITAL ELEMENTS ARRAY                           00000240
C         H      =     INTEGRATION STEPSIZE                             00000250
C INPUT/OUTPUT                                                          00000260
C     VIA COMMON WORKER                                                 00000270
C         K    =  INDEX OF TIME DERIVATIVE AT TIME T = T(K)             00000280
C     VIA COMMON TIMREG                                                 00000290
C         STREG=  CURRENT VALUE OF INDEPENDENT VARIABLE                 00000300
C OUTPUT                                                                00000310
C     VIA COMMON WORDER                                                 00000320
C         DPARRY   =  TIME DERIVATIVES OF THE ORBITAL ELEMENTS          00000330
C         XXSUM    =  FIRST SUMS CALCULATED FROM TIME DERIVATIVES OF THE00000340
C                     ORBITAL ELEMENTS                                  00000350
C         PARARY   =  ARRAYS OF ORBITAL ELEMENTS                        00000360
C         K        =  INDEX OF TIME DERIVATIVES AT TIME T = T(K)        00000370
C     VIA COMMON TIMER                                                  00000380
C         ITERMS   =  AVERAGE NUMBER OF CORRECTOR ITERATIONS IN STARTER 00000390
C                                                                       00000400
C       *         *         *         *         *         *         *   00000410
C                                                                       00000420
C LOCAL VARIABLES USED                                                  00000430
C                                                                       00000440
C         II     = COUNTER FOR CORRECTION ITERATIONS                    00000450
C         KSAVE  =  SAVED VALUE OF K INDEX                              00000460
C         TS     =  SAVED OF TIME T                                     00000470
C         III    =  LOWER INDEX OF ITERATION                            00000480
C         IUP    =  A NEXT ADVANCED INDEX FROM EPOCH IN INCREASING TIME 00000490
C                   DIRECTION                                           00000500
C         ILOW   =  A NEXT ADVANCED INDEX FROM EPOCH IN DECREASING TIME 00000510
C                   DIRECTION                                           00000520
C         II2    =  UPPER INDEX OF ITERATION                            00000530
C         EST    =  MAX DIFFERENCE BETWEEN STATES OF SUCCESSIVE ITERATNS00000540
C       *         *         *         *         *         *         *   00000550
C     EXTERNAL REFERENCES                                               00000560
C          APPSUM                                                       00000570
```

```
C           PARCOR                                                          00000580
C           INARRY                                                          00000590
C           ISTART                                                          00000600
C           EQUNTN                                                          00000610
C           AVSTRT                                                          00000620
C                                                                           00000630
C                                                                           00000640
C     REFERENCES                                                            00000650
C         GTDS TASK NAME - INTEGRATION MULTISTEP STARTER (MSTART)            00000660
C                                                                           00000670
C     PROGRAMMER                                                            00000680
C         K.S. NIMITZ - COMP. SCI. CORP.                                    00000690
C                                                                           00000700
C     PROGRAM MODIFICATIONS                                                 00000710
C         1.  DATE:    06/01/72                                           00000720
C             NAME:    K. S. NIMITZ, COMPUTER SCIENCES CORPORATION        00000730
C             REASON:  ORIGINAL PROGRAM MULSTP                            00000740
C         2.  DATE:    07/15/73                                           00000750
C             NAME:    A. C. LONG, COMPUTER SCIENCES CORPORATION          00000760
C             REASON:  CONFORM TO GTDS PROGRAMMING STANDARDS              00000770
C         3.  DATE:    01/14/74                                           00000780
C             NAME:    B. DIXON                                           00000790
C             REASON:  ALLOW FOR HIGHER ORDER INTEGRATION                 00000800
C         4.  DATE:    06/15/79                                           00000810
C             NAME:    CAPTAIN JEFFREY S. SHAVER, USAF                    00000820
C             REASON:  ALLOW IMMEDIATE RETURN AFTER CALL TO 'AVRINT' IF   00000830
C                      PICARD-CHEBYSHEV INTEGRATOR IS IN USE              00000840
C                                                                           00000850
C*   *   *   *   *   *   *   *    STARTER    PROGRAM    *   *   *    *00000860
C                                                                           00000870
      IMPLICIT REAL*8(A-H,O-Z)                                              00000880
C                                                                           00000890
C * * * * * * * * * * *        COMMON BLOCKS        * * * * * * * * * * * 00000900
C                                                                           00000910
      COMMON /INTEG / DUMMY(149)    , NORDER          , N2            , 00000920
     *                MAXIT(5)                                             00000930
C                                                                           00000940
      COMMON /SWITCH/ IND(100)      , IDUM(100)                           00000950
C                                                                           00000960
      COMMON /TIMER / D3(51)        , N3(23)                              00000970
C                                                                           00000980
      COMMON /TIMREG/ DTRG(94)                                            00000990
C                                                                           00001000
      COMMON /WORKER/ D4(3264)      , IWORK(13)                           00001010
C                                                                           00001020
      COMMON          DUM(192)      , SVEOP(15,6,20)  , VEVOP1(15,6)  , 00001030
     *                VEVPDS                                               00001040
C                                                                           00001050
C * * * * * * * * * * *    DIMENSION STATEMENTS    * * * * * * * * * * *00001060
C                                                                           00001070
      DIMENSION       VPELMT(10)    , DPARRY(15,10)   , XXSUM(10)     , 00001080
     *                PARARY(15,10) , TOL(6)          , NOSTEP(2)     , 00001090
     *                ITERMS(2)                                            00001100
C                                                                           00001110
      DIMENSION       VEVOP(6,20)   , VEVOPD(15,6,20) ,                   00001120
     *                PVOP(6,6)     , VELPAR(6,20)    ,                   00001130
     *                XXSUM(6,20)                                         00001140
```

```
C                                                                       00001150
C * * * * * * * * * *   EQUIVALENCE STATEMENTS   * * * * * * * * * * *00001160
C                                                                       00001170
C     FROM COMMON/INTEG /:                                              00001180
      EQUIVALENCE    (TOL(1),DUMMY(124))                                00001190
C                                                                       00001200
C     FROM COMMON/SWITCH/:                                              00001210
      EQUIVALENCE    (IND(20),ICHEBY)                                   00001220
C                                                                       00001230
C     FROM COMMON/TIMER /:                                              00001240
      EQUIVALENCE    (NOSTEP(1), N3(7))      , (ITERMS(1), N3(5))       00001250
C                                                                       00001260
C     FROM COMMON/TIMREG/:                                              00001270
      EQUIVALENCE    (STREG,DTRG(92))                                   00001280
C                                                                       00001290
C     FROM COMMON/WORKER/:                                              00001300
      EQUIVALENCE    (DPARRY(1,1),D4(7))     , (VPELMT(1), D4(2829))  ,00001310
     *               (H,D4(3216))            , (K,IWORK(2))           ,00001320
     *               (NOPARM, IWORK(3))      ,                         00001330
     *               (PARARY(1,1),D4(157))   , (XXSUM(1),D4(2819))    ,00001340
     *               (IWORK(5),NEQ)          ,                         00001350
     *               (D4(307),VEVOP(1,1))    ,                         00001360
     *               (D4(427),VEVOPD(1,1,1)) ,                         00001370
     *               (D4(2227),PVOP(1,1))    ,                         00001380
     *               (D4(2263),VELPAR(1,1))  ,                         00001390
     *               (D4(2383),XVSUM(1,1))                             00001400
      DO 4 I=1,15                                                       00001410
      DO 4 J=1,10                                                       00001420
 4    DPARRY(I,J)=0.0D0                                                 00001430
C                                                                       00001440
C                                                                       00001450
C                                                                       00001460
C     INITIALIZE VARIOUS TEMPORARY PARAMETERS                           00001470
C                                                                       00001480
      IERR = 1                                                          00001490
      II = 0                                                            00001500
      IJ = 0                                                            00001510
      IIR  =  0                                                         00001520
C                                                                       00001530
C     COMPUTE AND STORE TIME DERIVATIVES OF EPOCH ORBITAL ELEMENTS      00001540
C     IN THE KTH ROW OF THE DERIVATIVE ARRAY                            00001550
C                                                                       00001560
C                                                                       00001570
      IND3=IND(3)                                                       00001580
      K   =  NORDER/2  +  1                                             00001590
      KSAVE  =  K                                                       00001600
      IF (IND(3).GT.10) CALL AVRINT                                     00001610
C                                                                       00001620
C     CHECK IF PICARD-CHEBYSHEV IS BEING USED FOR VOP EQUATIONS.  IF SO,00001630
C     EXIT AFTER CALL TO 'AVRINT' TO SET UP AVERAGING FORCE MODELS.     00001640
C                                                                       00001650
      IF (ICHEBY.EQ.2)                GO TO 6                           00001660
      K = 1                                                             00001670
      IERR = 1                                                          00001680
      DO  600  KCOMP = 1,NOPARM                                         00001690
  600 PARARY(K,KCOMP) = VPELMT(KCOMP)                                   00001700
      GO TO 999                                                         00001710
```

```
C                                                                       00001720
    6 CALL EQUMTN(.FALSE.)                                              00001730
C                                                                       00001740
C     STORE CURRENT VALUE OF INDEPENDENT VARIABLE IN TS                 00001750
C                                                                       00001760
      TS=STREG                                                          00001770
C                                                                       00001780
C     STORE EPOCH ORBITAL ELEMENTS IN THE KTH ROW OF THE                00001790
C      INITIAL ELEMENT ARRAY                                            00001800
C                                                                       00001810
      DO 5 JST = 1,NOPARM                                               00001820
      PARARY(K,JST) = VPELMT(JST)                                       00001830
    5 CONTINUE                                                          00001840
C                                                                       00001850
C     COMPUTE VARIATIONAL PARTIALS OF VOP KEPLERIAN                     00001860
C                                                                       00001870
      IF(IND(3).NE.8 .OR. NEQ.EQ.0) GO TO 10                            00001880
C                                                                       00001890
C     PARTIAL WRT KEPLERIAN VOP PARAMETERS                              00001900
C                                                                       00001910
      CALL VARFRC                                                       00001920
      CALL VARFRV                                                       00001930
C                                                                       00001940
      DO 7 J=1,NOPARM                                                   00001950
      DO 3 I=1,NOPARM                                                   00001960
      SVEOP(K,J,I) = PVOP(J,I)                                          00001970
    3 CONTINUE                                                          00001980
C                                                                       00001990
C     PARTIAL WRT MODEL PARAMETERS                                      00002000
C                                                                       00002010
      DO 2 I=1,NEQ                                                      00002020
      SVEOP(K,J,I+NOPARM) = VELPAR(J,I)                                 00002030
    2 CONTINUE                                                          00002040
C                                                                       00002050
    7 CONTINUE                                                          00002060
   10 CONTINUE                                                          00002070
C                                                                       00002080
C                                                                       00002090
C     START COMPUTATION OF INITIAL ELEMENT ARRAY                        00002100
C                                                                       00002110
C                                                                       00002120
   15 IJ  =  IJ  +  1                                                   00002130
      IIR  =  IIR  +  II                                                00002140
      II  =  0                                                          00002150
C                                                                       00002160
C     BRANCH TO APPROPRIATE SUBROUTINE FOR COMPUTATION OF FIRST         00002170
C       APPROXIMATION                                                   00002180
C                                                                       00002190
      IF(IND(3).GT.10) GO TO 18                                         00002200
      GO TO(16,16,16,17,17,16,16),IND3                                  00002210
   16 CALL INARRY                                                       00002220
      GO TO 20                                                          00002230
   17 CALL ISTART(IND3)                                                 00002240
      GO TO 20                                                          00002250
   18 CALL AVSTRT                                                       00002260
C                                                                       00002270
C     START NEXT CORRECTION ITERATION                                   00002280
```

```
C                                                                       00002290
   20 II  =  II +  1                                                    00002300
      DK  =  K                                                          00002310
      STREG  =  TS  -  DK*H                                             00002320
C                                                                       00002330
C     START LOOP FOR COMPUTING TIME DERIVATIVES OF PARAMETERS           00002340
C                                                                       00002350
      III  =  1                                                         00002360
      IUP  =  K  +  1                                                   00002370
      ILOW  = K-  1                                                     00002380
      II2  =  ILOW                                                      00002390
      DO 45 II3 = 1,2                                                   00002400
      DO 40 KK  =  III,II2                                              00002410
      K  =  KK                                                          00002420
      STREG  =  STREG  +  H                                             00002430
C                                                                       00002440
C     STORE PARAMETERS AT T = 0 IN THE WORKING ARRAYS OF VPELMT(10) FOR  00002450
C     REFERENCE POINTS                                                  00002460
C                                                                       00002470
      DO 25 JPARM = 1,NOPARM                                            00002480
      VPELMT(JPARM)  =  PARARY(K,JPARM)                                 00002490
   25 CONTINUE                                                          00002500
C                                                                       00002510
C     COMPUTE THE TIME DERIVATIVES OF THE ORBITAL ELEMENTS              00002520
C                                                                       00002530
      CALL EQUMTN(.FALSE.)                                              00002540
      IF (IND(3).NE.8 .OR. NEQ.EQ.0) GO TO 40                           00002550
C                                                                       00002560
C     CALL FOR VARIATIONAL PARTIALS                                     00002570
C                                                                       00002580
      CALL VARFRC                                                       00002590
      CALL VARFRV                                                       00002600
C                                                                       00002610
C     STORE PARTIALS WRT VOP PARAMETERS                                 00002620
C                                                                       00002630
      DO 35 J=1,NOPARM                                                  00002640
      DO 30 I=1,NOPARM                                                  00002650
      SVEOP(K,J,I) = PVOP(J,I)                                          00002660
   30 CONTINUE                                                          00002670
C                                                                       00002680
C     STORE PARTIALS WRT MODEL PARAMETERS                               00002690
C                                                                       00002700
      DO 32 I=1,NEQ                                                     00002710
      SVEOP(K,J,I+NOPARM) = VELPAR(J,I)                                 00002720
   32 CONTINUE                                                          00002730
   35 CONTINUE                                                          00002740
C                                                                       00002750
   40 CONTINUE                                                          00002760
      STREG  =  STREG  +  H                                             00002770
      III  =  IUP                                                       00002780
      II2  =  NORDER + 1                                                00002790
   45 CONTINUE                                                          00002800
      STREG  =  STREG  -  H                                             00002810
      K  =  KSAVE                                                       00002820
C                                                                       00002830
C     COMPUTE THE FIRST SUMS OF THE ORBITAL ELEMENTS                    00002840
C                                                                       00002850
```

```
      CALL  APPSUM(PARARY,DPARRY,XXSUM)                                 00002860
C                                                                       00002870
C     TEST FOR TWO BODY MOTION                                          00002880
C                                                                       00002890
      IF(IND(6) .EQ. 1) GO TO 100                                       00002900
C                                                                       00002910
C     START THE CORRECTION PROCESS                                      00002920
C                                                                       00002930
      CALL PARCOR(PARARY,DPARRY,XXSUM,EST)                              00002940
C                                                                       00002950
C     TEST FOR CONVERGENCE                                              00002960
C                                                                       00002970
      IF (II.EQ.1) WRITE (6,8887)                                       00002980
 8887 FORMAT('1'///)                                                    00002990
      WRITE(6,8888) EST                                                 00003000
 8888 FORMAT(' CORR.TOL.= ',D23.16)                                     00003010
C                                                                       00003020
      IF(EST .GT. TOL(5)) GO TO 70                                      00003030
      GO TO 100                                                         00003040
C                                                                       00003050
C     TEST FOR MAX. NO. OF CORRECTION ITERATIONS                        00003060
C                                                                       00003070
   70 IF(II .LT. MAXIT(2)) GO TO 20                                     00003080
C                                                                       00003090
C     EXIT BECAUSE MAX. NO. OF CORRECTION ITERATIONS WAS EXCEEDED       00003100
C                                                                       00003110
      IERR = 3                                                          00003120
      GO TO 999                                                         00003130
C                                                                       00003140
C     STORE NUMBER OF STARTING ITERATIONS AND AVERAGE NUMBER OF         00003150
C     CORRECTOR ITERATIONS                                              00003160
C                                                                       00003170
  100 IIR =  IIR +  II                                                  00003180
      NOSTEP(1)  =  IJ                                                  00003190
      XIR  =  IIR                                                       00003200
      XJ  =  IJ                                                         00003210
      XIR  =  (XIR/XJ)   +  0.5D0                                       00003220
      ITERMS(1)  =  XIR                                                 00003230
      K = II2                                                           00003240
C                                                                       00003250
C     STORE THE LATEST VALUES OF PARAMETERS BACK IN VPELMT(10)          00003260
C                                                                       00003270
      DO  110 JP  =  1,NOPARM                                           00003280
      VPELMT(JP)  =  PARARY(NORDER + 1,JP)                              00003290
  110 CONTINUE                                                          00003300
C                                                                       00003310
C     TEST IF THERE ARE VARIATIONAL EQUATIONS NEEDING STARTING VALUES   00003320
C                                                                       00003330
      IF (IND(3).NE.8 .OR. NEQ.EQ.0) GO TO 999                          00003340
      IIR=0                                                             00003350
      II=0                                                              00003360
C                                                                       00003370
C     START LOOP OVER VARIATIONAL EQUATIONS                             00003380
C                                                                       00003390
      DO 240 I=1,NEQ                                                    00003400
C                                                                       00003410
C     STORE TOTAL NO. OF CORRECTOR ITERATIONS OVER ALL EQNS.            00003420
```

```
C                                                                       00003430
      IIR = IIR+II                                                      00003440
      II=0                                                              00003450
C                                                                       00003460
      DO 160 J=1,NOPARM                                                 00003470
      VEVOPT = VEVOP(J,I)                                               00003480
C                                                                       00003490
C     STORE THE FIRST APPROXIMATION TO NORDER STARTING PARTIALS          00003500
C                                                                       00003510
      II4 = NORDER+1                                                    00003520
      DO 160 L=1,II4                                                    00003530
      VEVOP1(L,J) = VEVOPT                                              00003540
  160 CONTINUE                                                          00003550
C                                                                       00003560
C     STORE VELOCITY PARTIALS AT K=NORDER/2+1                           00003570
C                                                                       00003580
      K=NORDER/2+1                                                      00003590
      DO 168 J=1,NOPARM                                                 00003600
      VEVPDS = SVEOP(K,J,I+NOPARM)                                      00003610
      DO 162 L=1,NOPARM                                                 00003620
  162 VEVPDS = VEVPDS + SVEOP(K,J,L)*VEVOP1(K,L)                        00003630
C                                                                       00003640
  166 VEVOPD(K,J,I) = VEVPDS                                            00003650
C                                                                       00003660
  168 CONTINUE                                                          00003670
C                                                                       00003680
C     START NEXT CORRECTOR ITERATION                                    00003690
C                                                                       00003700
  170 II = II+1                                                         00003710
      II1 = 1                                                           00003720
      IUP = K+1                                                         00003730
      ILOW= K-1                                                         00003740
      II2 = ILOW                                                        00003750
C                                                                       00003760
      DO 195 II3=1,2                                                    00003770
C                                                                       00003780
C     START VECTOR COMPONENT LOOP                                       00003790
C                                                                       00003800
      DO 190 J=1,NOPARM                                                 00003810
C                                                                       00003820
C     START TIME LOOP K                                                 00003830
C                                                                       00003840
      DO 190 KK=II1,II2                                                 00003850
C                                                                       00003860
C     INITIALIZE VELOCITY PARTIALS                                      00003870
C                                                                       00003880
      VEVPDS = SVEOP(KK,J,I+NOPARM)                                     00003890
C                                                                       00003900
C     START MATRIX MULTIPLICATION LOOP                                  00003910
C                                                                       00003920
      DO 180 L=1,NOPARM                                                 00003930
      VEVPDS = VEVPDS + SVEOP(KK,J,L)*VEVOP1(KK,L)                      00003940
  180 CONTINUE                                                          00003950
C                                                                       00003960
C     STORE VELOCITY PARTIALS                                           00003970
C                                                                       00003980
  186 VEVOPD(KK,J,I) = VEVPDS                                           00003990
```

```
MULSTP              JSS1493.GTDS.UPDATE.FORT

C                                                                       00004000
  190 CONTINUE                                                          00004010
      III = IUP                                                         00004020
      II2 = NORDER+1                                                    00004030
  195 CONTINUE                                                          00004040
C                                                                       00004050
C     CALL FOR IITH APPROXIMATION TO VOP 1ST SUMS - SUMMING OVER ALL II 00004060
C     TIMES.                                                            00004070
C                                                                       00004080
      CALL APPSUM (VEVOP1,VEVOPD(1,1,I),XVSUM(1,I))                     00004090
C                                                                       00004100
C     CALL FOR (II+1)TH APPROX OF THE STARTING POSITION SUMMING OVER    00004110
C     ALL TIMES                                                         00004120
C                                                                       00004130
      CALL PARCOR (VEVOP1,VEVOPD(1,1,I),XVSUM(1,I),EST)                 00004140
C                                                                       00004150
C     TEST FOR CONVERGENCE                                              00004160
C                                                                       00004170
      WRITE(6,8889) EST                                                 00004180
 8889 FORMAT (' V.E.CORR.TOL. = ',D23.16)                               00004190
      IF(EST.LE.TOL(6)) GO TO 220                                       00004200
C                                                                       00004210
C     TEST IF MAX. NO. OF CORRECTOR ITERATIONS WILL BE EXCEEDED         00004220
C                                                                       00004230
      IF(II.LT.MAXIT(2)) GO TO 170                                      00004240
C                                                                       00004250
C     EXIT BECAUSE MAX. NO. OF CORRECTOR ITERATIONS WILL BE EXCEEDED.   00004260
C                                                                       00004270
      IERR=4                                                            00004280
      GO TO 999                                                         00004290
C                                                                       00004300
C     STORE LATEST VALUES OF POSITION PARTIALS                          00004310
C                                                                       00004320
  220 DO 230 J=1,NOPARM                                                 00004330
      VEVOP(J,I) = VEVOP1(NORDER+1,J)                                   00004340
  230 CONTINUE                                                          00004350
  240 CONTINUE                                                          00004360
C                                                                       00004370
C     STORE AVERAGE NO. OF CORRECTOR ITERATIONS OVER ALL EQNS.          00004380
C                                                                       00004390
      IIR = IIR+II                                                      00004400
      XJ = NEQ                                                          00004410
      XIR= IIR                                                          00004420
      XIR= (XIR/XJ)+.5D0                                                00004430
      ITERMS(2)=XIR                                                     00004440
      K = II2                                                           00004450
  999 RETURN                                                            00004460
      END                                                               00004470

******** END OF MEMBER MULSTP     447 RECORDS ********
```

```
      SUBROUTINE ORBITT (INTFLG,MSGERR,*)                                00000010
      IMPLICIT REAL*8 (A-H,O-Z)                                          00000020
C     VERSION 3.0                                                        00000030
C                                                                        00000040
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00000050
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00000060
C                                                                        00000070
C  ORBITT SUBPROGRAM IS DESIGNED TO INTEGRATE OR TO INTERPOLATE THE      00000080
C  POSITION AND VELOCITY OF THE SPACECRAFT AT REQUEST TIME (TTO) USING   00000090
C  A CHEBYSHEV SERIES OF ORTHOGONAL POLYNOMIALS.                         00000100
C                                                                        00000110
C                                                                        00000120
C                                                                        00000130
C                                                                        00000140
C  NOTE THAT THE NUMBER OF EQUATIONS, NEQ, IS LIMITED TO 0 FOR NOW.      00000150
C                                                                        00000160
C  CALLING SEQUENCE IS:      CALL  ORBITT  (INTFLG,*)                    00000170
C    WHERE:                                                              00000180
C    INTFLG = 1  MEANS THAT ONLY PARTIALS ARE COMPUTED BY INTERPOLATION  00000190
C    INTFLG = 2  MEANS THAT OUTPUT IS OBTAINED BY CHEBYSHEV INTEGRATION  00000200
C                OR CHEBYSHEV INTERPOLATION                              00000210
C                                                                        00000220
C                                                                        00000230
C  AUTHORED BY T. FEAGIN                                                 00000240
C                                                                        00000250
C  RESEARCH WAS ACCOMPLISHED WHILE THE AUTHOR WAS A NATIONAL RESEARCH    00000260
C  COUNCIL RESIDENT RESEARCH ASSOCIATE AT NASA'S GSFC                    00000270
C                                                                        00000280
C  DIRECTED BY C. E. VELEZ                                               00000290
C                                                                        00000300
C  FEATURING THE ORTHOGONAL POLYNOMIAL SERIES DERIVED BY THE RUSSIAN     00000310
C  MATHEMATICIAN PAVNUTII LVOVICH CHEBYSHEV OR TSCHEBYCHEFF              00000320
C                                                                        00000330
C                                                                        00000340
C                                                                        00000350
C  MODIFICATIONS                                                         00000360
C                                                                        00000370
C  05/01/74 BY R.PAJERSKI TO ADD CALL TO CHETO TO END FILE WHEN          00000380
C                  INTEGRATOR FINISHED                                   00000390
C                                                                        00000400
C  05/16/79 BY JEFFREY S. SHAVER, CAPTAIN, USAF TO USE THE MODIFIED      00000410
C                  SUBROUTINES 'PCHEBY' AND 'CSRSTA'.                    00000420
C                                                                        00000430
C                                                                        00000440
C * * * * * * * * *        COMMON BLOCKS          * * * * * * * * * *   00000450
C                                                                        00000460
      COMMON/NPCNTL/ MORD, NSUB, IEQSET, ITERM                           00000470
C                                                                        00000480
      COMMON/COVAL / A(70,6), BC(70,6), FORC(70,6)                       00000490
C                                                                        00000500
      COMMON/SECTN / DPSEC(675) ,INSEC(325)                              00000510
C                                                                        00000520
      COMMON/WORKER/ DPWOR(3264),INWOR(13)                               00000530
C                                                                        00000540
      COMMON/DCINT / DPDCI(432) ,INDCI(26)                               00000550
C                                                                        00000560
      COMMON/SWITCH/ INSWI(200)                                          00000570
C                                                                        00000580
      COMMON/SATPOS/ DPSAT(100) ,INSAT(6)                                00000590
C                                                                        00000600
```

```
ORBITT              JSS1493.GTDS.UPDATE.FORT

C                                                                       00000580
      COMMON/SATPOS/ DPSAT(100) ,INSAT(6)                               00000590
C                                                                       00000600
      COMMON/SATMAN/ DDPSAT(164) ,IINSAT(10)                            00000610
C                                                                       00000620
      COMMON/CHVTP / DPCHV(297),IEDFP                                   00000630
C                                                                       00000640
C * * * * * * * * *      DIMENSION STATEMENTS        * * * * * * * * * *00000650
C                                                                       00000660
      DIMENSION      CV(1,1,1),CVD(1,1,1)                               00000670
C                                                                       00000680
      DIMENSION      TEMPPX(6,20)          ,PARMAN(6,20)               00000690
C                                                                       00000700
      DIMENSION      IND(100)   ,XTO(3)   ,XDTO(3)     ,PXTXO(6,20)    00000710
C                                                                       00000720
      DIMENSION      XDUM(6)    ,STA(6)                                 00000730
C                                                                       00000740
C * * * * * * * * *     EQUIVALENCE STATEMENTS       * * * * * * * * * *00000750
C                                                                       00000760
C     FROM COMMON/SATMAN/:                                              00000770
      EQUIVALENCE    (DDPSAT(45),PARMAN(1,1))                           00000780
C                                                                       00000790
C     FROM COMMON/SATPOS/:                                              00000800
      EQUIVALENCE    (DPSAT(27),TSTRT)    ,(DPSAT(28),TEND)            00000810
C                                                                       00000820
C     FROM COMMON/DCINT /:                                              00000830
      EQUIVALENCE    (DPDCI(1),XTO(1))    ,(DPDCI(4),XDTO(1))   ,      00000840
     1               (DPDCI(7),PXTXO(1,1)),(DPDCI(127),TTO)     ,      00000850
     2               (INDCI(1),IV)        ,(INDCI(2),NPART)     ,      00000860
     3               (INDCI(4),IVPAR)     ,(DPDCI(1),STA(1))           00000870
C                                                                       00000880
C     FROM COMMON/WORKER/:                                              00000890
      EQUIVALENCE    (DPWOR(3215),T)      ,(DPWOR(3216),H)      ,      00000900
     1               (INWOR(2),K)         ,(INWOR(13),NSTR)     ,      00000910
     2               (INWOR(5),NEQ)                                     00000920
C                                                                       00000930
C     FROM COMMON/SWITCH/:                                              00000940
      EQUIVALENCE    (INSWI(1),IND(1))    ,(INSWI(154),NSTATE) ,       00000950
     1               (INSWI(170),IELEVN)                                00000960
C                                                                       00000970
C     FROM COMMON/SECTN /:                                              00000980
      EQUIVALENCE    (INSEC(313),NCONDT)  ,(INSEC(315),MANDON)         00000990
C                                                                       00001000
C  START PROGRAM                                                        00001010
C                                                                       00001020
      IF(INTFLG.EQ.1)GO TO 160                                          00001030
C                                                                       00001040
C  TEST FOR TERMINAL SECTION                                            00001050
C                       NO-     CONTINUE                              00001060
   20 IF(NCONDT.LE.100) GO TO 30                                        00001070
C                                                                       00001080
C  CHEBY TAPE DESIRED                                                   00001090
C                       NO-       RETURN                              00001100
      IF(INSWI(87).EQ.0) GO TO 999                                      00001110
      IEDFP= 3                                                          00001120
      CALL CHETO(XDUM)                                                  00001130
      GO TO 999                                                         00001140
```

```
   30 CONTINUE                                                          00001150
C                                                                       00001160
C  TEST FOR SECTIONING WITH INTERVENING PRINT                           00001170
C                                                                       00001180
      IF (NCONDT)  150,40,40                                            00001190
  150 CALL SECHEK  (&160,&170,&160)                                     00001200
C                                                                       00001210
C  IS REQUEST TIME LESS THAN CURRENT TIME  IF SO, USE SERIES TO INTERP  00001220
C  IF REQUEST TIME IS EQUAL TO CURRENT TIME THEN NO INTERPOLATION NEEDED00001230
C                                                                       00001240
   40 IF (DABS(TTO)-DABS(T))  170,45,50                                 00001250
   45 DO 47 I=1,3                                                       00001260
      XTO(I)=DFWDR(I)                                                   00001270
   47 XDTO(I)=DFWDR(I+3)                                                00001280
      GO TO 180                                                         00001290
C                                                                       00001300
C  OTHERWISE,  IT'S TIME TO TAKE ANOTHER STEP                           00001310
C                                                                       00001320
   50 T=T+H                                                             00001330
      CALL PCHEBY(&165)                                                 00001340
C                                                                       00001350
C  TEST FOR RUN WITH FIXED STEPSIZE                                     00001360
C                                                                       00001370
C                              YES                                      00001380
  120 IF (IND(26).EQ.1)  GO TO 150                                      00001390
C                                                                       00001400
C  SOME DAY WE MAY WANT TO VARY STEPSIZE HERE, BUT FOR THE PRESENT JUST 00001410
      GO TO 150                                                         00001420
C                                                                       00001430
C  SECTIONING HAS OCCURRED                                              00001440
C                                                                       00001450
  160 CALL CROSST (MSGERR,&165)                                         00001460
      GO TO 20                                                          00001470
  165 RETURN1                                                           00001480
C                                                                       00001490
C  INTERPOLATE USING THE CURRENT CHEBYSHEV SERIES OF POLYNOMIALS        00001500
C                                                                       00001510
  170 CALL CSRSTA(STA,T,TTO,H,IV)                                       00001520
C                                                                       00001530
C  THE REST OF THIS SUBPROGRAM IS LIKE ORBITC                           00001540
C                                                                       00001550
C  TEST FOR COMPUTING THE PARTIALS                                      00001560
C                                                                       00001570
  180 IF (NPART.NE.0)  GO TO 999                                        00001580
      IF (IND(4).EQ.1)  GO TO 195                                       00001590
      IF (NSTATE.EQ.0)  GO TO 190                                       00001600
C                                                                       00001610
C  CALCULATE THE STATE PARTIALS ANALYTICALLY                            00001620
C                                                                       00001630
      CALL ANPART (XTO,XDTO,TTO,2,PXTXO)                                00001640
      IF(NEQ .LE. 0) GO TO 999                                          00001650
C                                                                       00001660
C  SET-UP TO COMPUTE THE DYNAMIC PARTIALS FOR NON-STATE PARAMETERS WHEN 00001670
C  ANALYTIC OPTION IS TURNED ON                                         00001680
C                                                                       00001690
  190 N=NSTATE                                                          00001700
      GO TO 245                                                         00001710
```

```
C                                                                       00001720
C  TEST FOR ANY MANEUVERS                                               00001730
C                                                                       00001740
  195 N=0                                                               00001750
      IF (MANDON.EQ.0)  GO TO 245                                       00001760
      DO 200 I=1,NEQ                                                    00001770
      CALL CHIRP (TEMPPX(1,I),TEMPPX(4,I),CV(1,1,I),CVD(1,1,I),T,TTO,H, 00001780
     *     IVPAR)                                                       00001790
  200 CONTINUE                                                          00001800
C                                                                       00001810
C  TRANSFORM THE PARTIALS TO EPOCH                                      00001820
C                                                                       00001830
      IF (IVPAR-2)  211,212,213                                         00001840
  211 L1=1                                                              00001850
      L2=6                                                              00001860
      GO TO 214                                                         00001870
  212 L1=1                                                              00001880
      L2=3                                                              00001890
      GO TO 214                                                         00001900
  213 L1=4                                                              00001910
      L2=6                                                              00001920
  214 DO 220 I=1,NEQ                                                    00001930
      DO 220 L=L1,L2                                                    00001940
      PR=0.D0                                                           00001950
      DO 215 J=1,6                                                      00001960
  215 PR=PR+TEMPPX(L,J)*PARMAN(J,I)                                     00001970
  220 PXTXO(L,I)=PR                                                     00001980
      IF (NEQ.LE.6)  GO TO 999                                          00001990
      DO 225 I=7,NEQ                                                    00002000
  225 PXTXO(L,I)=PXTXO(L,I)+TEMPPX(L,I)                                 00002010
      GO TO 999                                                         00002020
C                                                                       00002030
C  SET-UP THE PARTIAL'S  OUTPUT FOR NON-MANEUVER CASE                   00002040
C                                                                       00002050
  245 DO 250 I=1,NEQ                                                    00002060
      CALL CHIRP (PXTXO(1,N+I),PXTXO(4,N+I),CV(1,1,I),CVD(1,1,I),T,TTO, 00002070
     *     H,IVPAR)                                                     00002080
  250 CONTINUE                                                          00002090
  999 RETURN                                                            00002100
      END                                                               00002110

******** END OF MEMBER ORBITT     211 RECORDS ********
```

```
      SUBROUTINE ORBITV(INTFLG,*)                                       00000010
C                                                                       00000020
C     VERSION 2.2                                                       00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INTEGRATE FOR POSITION AND VELOCITY AT THE REQUEST TIME    00000060
C         USING THE VARIATION-OF-PARAMETERS ORBIT GENERATOR             00000070
C                                                                       00000080
C     CALLING SEQUENCE                                                  00000090
C         CALL ORBITV(INTFLG,*)                                         00000100
C             INTFLG=1 RETURN                                           00000110
C             INTFLG=2 PERFORM VOP INTEGRATION                          00000120
C                                                                       00000130
C     COMMON BLOCK VARIABLES                                            00000140
C         INPUT                                                         00000150
C             H       STEPSIZE                                          00000160
C             IND     ARRAY OF CONTROL SWITCHES                         00000170
C             NPART   NUMBER OF PARTIALS REQUIRED                       00000180
C             NSTR    NUMBER OF BACK POINTS RECOPIED                    00000190
C             NOPARM  NUMBER OF ORBITAL ELEMENTS INTEGRATED             00000200
C             NORDER  NUMBER OF BACK POINTS USED IN INTEGRATOR          00000210
C             STREG   VALUE OF INDEPENDENT VARIABLE AT T=T(K)           00000220
C                                                                       00000230
C                                                                       00000240
C         INPUT-OUTPUT                                                  00000250
C             NCONDT  SECTIONING CONDITION FLAG                         00000260
C             XXSUM   FIRST SUM VECTOR OF DERIVATIVES                   00000270
C             TTO     OUTPUT REQUEST TIME                               00000280
C             DPARRY  DERIVATIVE ARRAY                                  00000290
C                                                                       00000300
C         OUTPUT                                                        00000310
C             K       DERIVATIVE ARRAY INDICATOR OF VALUE AT T=T(K)     00000320
C             PXTXO   VARIATIONAL PARTIAL DERIVATIVES                   00000330
C             T       TIME FROM EPOCH OF K TH  DERIVATIVE               00000340
C             XDTO    S/C VELOCITY VECTOR AT TIME TTO                   00000350
C             XTO     S/C POSITION VECTOR AT TIME TTO                   00000360
C                                                                       00000370
C                                                                       00000380
C     EXTERNAL REFERENCES                                               00000390
C         CSTEPX                                                        00000400
C         SECHEK                                                        00000410
C         CROSSV                                                        00000420
C         HEMITR                                                        00000430
C         INTPAR                                                        00000440
C         DALLPV                                                        00000450
C         EQUNPV                                                        00000460
C         IDLPV                                                         00000470
C         DPEL2X                                                        00000480
C         ANPART                                                        00000490
C         CELEM                                                         00000500
C                                                                       00000510
C     PROGRAMMER                                                        00000520
C         R. W. NELSON      (PROGRAMMED ORIGINAL ORBIT SUBROUTINE)      00000530
C                                                                       00000540
C     PROGRAM MODIFICATIONS                                             00000550
C         1.  DATE:     04/01/72                                        00000560
C             NAME:     A. C. LONG                                      00000570
```

```
C             REASON:  CODE ORIGINAL SUBROUTINE ORBITV                        00000580
C         2.  DATE:    07/15/73                                             00000590
C             NAME:    A. C. LONG                                           00000600
C             REASON:  INCLUDE MORE COMMENTS                                00000610
C         3.  DATE:    06/15/74                                             00000620
C             NAME:    R. PAJERSKI                                          00000630
C             REASON:  ADDED NEW ELEMENT SET (KEPLERIAN)                    00000640
C         4.  DATE:    01/10/75                                             00000650
C             NAME:    R. PAJERSKI                                          00000660
C             REASON:  DELETED ABSOLUTE VALUE TEST                          00000670
C         5.  DATE:    06/15/79                                             00000680
C             NAME:    CAPTAIN JEFFREY S. SHAVER, USAF                      00000690
C             REASON:  ADDED CAPABILITY TO INTEGRATE VOP EQUATIONS WITH     00000700
C                      PICARD-CHEBYSHEV INTEGRATOR                          00000710
C                                                                           00000720
C**********************     START PROGRAM     **************************00000730
C                                                                           00000740
      IMPLICIT REAL*8(A-H,O-Z)                                              00000750
C                                                                           00000760
C * * * * * * * * * * *      COMMON BLOCKS      * * * * * * * * * * * * * 00000770
C                                                                           00000780
      COMMON/DCINT /  DPDCI(432)     ,     INDCI(26)                      00000790
C                                                                           00000800
      COMMON/ FRC  /  DPFRC(1300)    ,     INTFRC(47)                     00000810
C                                                                           00000820
      COMMON/INTEG /  DPINT(149)     ,     INT(1)                         00000830
C                                                                           00000840
      COMMON/SECTN /  DPSEC(675)     ,     INSEC(325)                     00000850
C                                                                           00000860
      COMMON/SWITCH/  INSWI(200)                                          00000870
C                                                                           00000880
      COMMON/TIMREG/  DTRG(94)                                            00000890
C                                                                           00000900
      COMMON/WORKER/  DPWOR(3264)    ,     INWOR(13)                      00000910
C                                                                           00000920
      COMMON          TEMPPK(6,20)   ,     PKPC(6,6)                      00000930
C                                                                           00000940
C * * * * * * * * * * *    DIMENSION STATEMENTS    * * * * * * * * * * * *00000950
C                                                                           00000960
      DIMENSION       VPELMT(10)     ,   DPARRY(15,10)   ,   XXSUM(10),00000970
     1                IND(100)       ,   XTO(3)          ,   XDTO(3)  ,00000980
     2                PXTXO(6,20)                                        00000990
C                                                                           00001000
      DIMENSION       TREG(40)       ,   U(4)            ,   US(4)      00001010
C                                                                           00001020
      DIMENSION       GM(11)                                             00001030
C                                                                           00001040
      DIMENSION       PARARY(15,10)                                      00001050
C                                                                           00001060
      DIMENSION       VEVOP(6,20)    ,   VEVOPD(15,6,20) ,              00001070
     1                XVSUM(6,20)    ,   VOPTMP(6)                      00001080
C                                                                           00001090
C * * * * * * * * * * *   EQUIVALENCE STATEMENTS   * * * * * * * * * * * *00001100
C                                                                           00001110
C     FROM COMMON/DCINT /:                                                  00001120
      EQUIVALENCE    (DPDCI(1),XTO(1))     , (DPDCI(4),XDTO(1))    ,     00001130
     1               (DPDCI(7),PXTXO(1,1)) , (DPDCI(127),TTO)      ,     00001140
```

```
     2               (INDCI(2),NPART)                                          00001150
C                                                                              00001160
C     FROM COMMON/ FRC  /:                                                     00001170
      EQUIVALENCE    (DPFRC(2),GM(1))      , (INTFRC(1),ICENT)                 00001180
C                                                                              00001190
C     FROM COMMON/INTEG /:                                                     00001200
      EQUIVALENCE    (INT(1),NORDER)                                           00001210
C                                                                              00001220
C     FROM COMMON/SECTN /:                                                     00001230
      EQUIVALENCE    (INSEC(313),NCONDT)   , (INSEC(315),MANDON)               00001240
C                                                                              00001250
C     FROM COMMON/SWITCH/:                                                     00001260
      EQUIVALENCE    (INSWI(1),IND(1))     , (INSWI(54),NSTATE)               ,00001270
     1               (INSWI(20),ICHEBY)                                        00001280
C                                                                              00001290
C     FROM COMMON/TIMREG/:                                                     00001300
      EQUIVALENCE    (STREG,DTRG(92))      , (TREG(1),DTRG(1))                 00001310
C                                                                              00001320
C     FROM COMMON/WORKER/:                                                     00001330
      EQUIVALENCE    (DPWOR(7),DPARRY(1,1))    ,(DPWOR(2819),XXSUM(1)) ,00001340
     1               (DPWOR(3215),T)           ,                            00001350
     2               (DPWOR(3216),H)           ,(INWOR(2),K)               ,00001360
     3               (INWOR(3),NOPARM)         ,(INWOR(13),NSTR)           ,00001370
     4               (DPWOR(157),PARARY(1,1))  ,                            00001380
     5               (DPWOR(307),VEVOP(1,1))   ,                            00001390
     6               (DPWOR(427),VEVOPD(1,1,1)),                            00001400
     7               (DPWOR(2383),XVSUM(1,1))  ,                            00001410
     8               (INWOR(5),NEQ)                                         00001420
      NX=NORDER+1                                                            00001430
      STIME=TTO                                                              00001440
      IND3=IND(3)                                                            00001450
      IF(INTFLG.EQ.1)GO TO 180                                               00001460
C                                                                              00001470
C     TEST FOR TERMINAL SECTION                                                00001480
C                                                                              00001490
   20 IF(NCONDT.GT.100) GO TO 999                                            00001500
C                                                                              00001510
C     TEST FOR SECTIONING WITH INTERVENING PRINT                               00001520
C                                                                              00001530
      IF(NCONDT.LT.0) GO TO 150                                              00001540
C                                                                              00001550
C     IS REQUEST TIME LESS THEN CURRENT TIME, IF SO INTERPOLATE                00001560
C                                                                              00001570
      IF(IND3.EQ.7)   T=TREG(K-2)                                            00001580
   40 IF(DABS(TTO).LE.DABS(T + 1.D-02)) GO TO 170                            00001590
C                                                                              00001600
C     SET PARAMETERS TO INTEGRATE ANOTHER STEP                                 00001610
C     **  CHECK ICHEBY TO SEE IF PICARD-CHEBYSHEV INTEGRATION IS IN USE        00001620
C                                                                              00001630
   50 IF (ICHEBY.EQ.1)                    GO TO 400                          00001640
      K=K+1                                                                  00001650
      STREG=STREG+H                                                          00001660
C                                                                              00001670
C     IS THE DERIVATIVE ARRAY FULL, IF SO RECOPY THE LAST NSTR ROWS            00001680
C                                                                              00001690
      IF(K.LE.15) GO TO 110                                                  00001700
      DO 100 I=1,NSTR                                                        00001710
```

```
      NBCK = 15 - NSTR + I                                                    00001720
      IF(IND3.NE.6.AND.IND3.NE.7)   GO TO 90                                  00001730
      TREG(I)=TREG(NBCK)                                                      00001740
   90 CONTINUE                                                                00001750
      DO 100 J=1,NOPARM                                                       00001760
      DPARRY(I,J)=DPARRY(NBCK,J)                                              00001770
      IF(IND3.EQ.7.AND.IND(26).NE.1)  PARARY(I,J)=PARARY(NBCK,J)              00001780
C                                                                             00001790
      IF (NEQ.LE.0) GO TO 100                                                 00001800
      DO 95 L=1,NEQ                                                           00001810
      VEVOPD(I,J,L) = VEVOPD(15-NSTR+I,J,L)                                   00001820
   95 CONTINUE                                                                00001830
  100 CONTINUE                                                                00001840
      K=NSTR+1                                                                00001850
C                                                                             00001860
C     INTEGRATE TO NEXT STEP                                                  00001870
C                                                                             00001880
  110 CALL CSTEPX(IERR)                                                       00001890
C                                                                             00001900
C     CHECK FOR SECTIONING                                                    00001910
C                                                                             00001920
  150 CALL SECHEK(&160,&170,&160)                                             00001930
      GO TO 40                                                                00001940
C                                                                             00001950
C     SECTIONING HAS OCCURRED                                                 00001960
C                                                                             00001970
  160 CALL CROSSV(MSGERR,&165)                                                00001980
      GO TO 20                                                                00001990
C                                                                             00002000
C     ERROR RETURN IF MAXIMUM CORRECTOR ITERATIONS WERE VIOLATED OR           00002010
C     ERROR FROM CROSSV                                                       00002020
C                                                                             00002030
  165 RETURN1                                                                 00002040
C                                                                             00002050
C     IS THIS A KS RUN, IF SO CALL THE TIME INTERPOLATOR                      00002060
C                                                                             00002070
  170 IF(IND3.EQ.6) CALL HEMITR(STIME,TTO,K,H,STREG,TREG)                     00002080
C                                                                             00002090
C      IS THIS A DS RUN,IF SO CALL THE INTERPOLATOR                           00002100
C                                                                             00002110
      IF(IND3.EQ.7)  CALL HEMIDS(STIME,TTO,K,H,STREG,TREG,NX,XTO,XDTO)        00002120
      IF(IND3.EQ.7)  GO TO 180                                                00002130
C                                                                             00002140
C     INTERPOLATE FOR ORBITAL ELEMENTS AT THE REQUEST TIME:                   00002150
C     **CHECK ICHEBY TO SEE IF PICARD-CHEBYSHEV INTEGRATION IS IN USE.        00002160
C     **IF SO, USE CSRSTA TO INTERPOLATE FOR ELEMENTS AT REQUEST TIME.        00002170
C                                                                             00002180
      IF (ICHEBY.EQ.1)                  GO TO 1700                            00002190
      CALL INTPAR(VPELMT,DPARRY,XXSUM,STREG,STIME,K,.FALSE.,NX,NOPARM,H)      00002200
      GO TO 1750                                                              00002210
 1700 IF (DABS(TTO).LT.DABS(T))         GO TO 1720                           00002220
      DO 1710  INDX = 1,10                                                    00002230
 1710 VPELMT(INDX) = PARARY(1,INDX)                                           00002240
      GO TO 1750                                                              00002250
 1720 CALL CSRSTA(VOPTMP,STREG,STIME,H,1)                                     00002260
      DO 1730  INDX = 1,6                                                     00002270
 1730 VPELMT(INDX) = VOPTMP(INDX)                                             00002280
```

```
      DO 1740  INDX = 7,10                                                    00002290
 1740 VPELMT(INDX) = 0.D0                                                     00002300
C                                                                             00002310
C     BRANCH TO THE APPROPRIATE SUBROUTINE FOR COMPUTATION OF POSITION        00002320
C     AND VELOCITY FROM THE ORBITAL ELEMENTS                                  00002330
C                                                                             00002340
 1750 IF(IND3.GT.10) IND3=IND3-10                                             00002350
      GO TO (171,172,172,173,173,174,171,176),IND3                            00002360
  171 CALL DALLPV(VPELMT,XTO,XDTO,JERROR)                                     00002370
      GO TO 180                                                               00002380
  172 CALL EGUNPV(VPELMT,XTO,XDTO,JERROR,IND3)                                00002390
      GO TO 180                                                               00002400
  173 CALL IDLPV(VPELMT,XTO,XDTO,IND3)                                        00002410
      GO TO 180                                                               00002420
  174 CALL DPEL2X(VPELMT,STIME,U,US,XTO,XDTO,SE2,CE2)                         00002430
      GO TO 180                                                               00002440
  176 CALL CELEM(VPELMT,GM(ICENT),XTO,XDTO)                                   00002450
C                                                                             00002460
C     ARE ANALYTIC PARTIALS REQUIRED                                          00002470
C                                                                             00002480
  180 IF(NPART.NE.0) GO TO 999                                                00002490
      IF (IND(4).EQ.1) GO TO 195                                              00002500
      IF (NSTATE.EQ.0) GO TO 190                                              00002510
C                                                                             00002520
C     CALCULATE ANALYTIC STATE PARTIALS                                       00002530
C                                                                             00002540
      CALL ANPART(XTO,XDTO,TTO,2,PXTXO)                                       00002550
C                                                                             00002560
  188 IF (NEQ.LE.0) GO TO 999                                                 00002570
C                                                                             00002580
C     SET UP TO COMPUTE DYNAMIC PARTIALS FOR NON-STATE PARAMETERS             00002590
C     WHEN ANALYTIC OPTION IS ON.                                             00002600
C                                                                             00002610
  190 N = NSTATE                                                              00002620
      GO TO 245                                                               00002630
C                                                                             00002640
C     TEST FOR MANEUVERS                                                      00002650
C                                                                             00002660
  195 N = 0                                                                   00002670
      IF (MANDON.EQ.0) GO TO 245                                              00002680
C                                                                             00002690
C     SET UP PARTIAL OUTPUT FOR NON-MANEUVER CASE                             00002700
C                                                                             00002710
  245 DO 250 I=1,NEQ                                                          00002720
      CALL INTPAR (TEMPPK(1,N+I),VEVOPD(1,1,I),XVSUM(1,I),STREG,STIME,        00002730
     *               K,.TRUE.,NX,NOPARM,H)                                    00002740
  250 CONTINUE                                                                00002750
C                                                                             00002760
C     COMPUTE CARTESIAN VARIATIONAL PARTIALS FROM KEPLERIAN                   00002770
C     VOP VARIATIONSL PARTIALS.                                               00002780
C                                                                             00002790
  300 CALL KPART (XTO,PKPC)                                                   00002800
      CALL INV2 (6)                                                           00002810
      CALL MATMUL (PKPC,TEMPPK(1,N+1),PXTXO(1,N+1),6,6,20-N)                  00002820
      GO TO 999                                                               00002830
C                                                                             00002840
C     THIS IS THE JUMP IF PICARD-CHEBYSHEV INTEGRATION IS TO BE USED          00002850
```

```
ORBITV              JSS1493.GTDS.UPDATE.FORT

C     TO INTEGRATE THE VOP EQUATIONS INSTEAD OF THE MULTI-STEP INTE-    00002860
C     GRATOR.  ICHEBY = 1 KEYS THIS JUMP.  AFTER INTEGRATION, CONTROL   00002870
C     RETURNS TO 150 TO CHECK FOR SECTIONING.                           00002880
C                                                                       00002890
  400 K = 1                                                             00002900
      STREG = STREG + H                                                 00002910
      T = STREG                                                         00002920
      CALL PCHEBY(&165)                                                 00002930
      GO TO 150                                                         00002940
C                                                                       00002950
C     NORMAL RETURN                                                     00002960
C                                                                       00002970
  999 RETURN                                                            00002980
      END                                                               00002990

******** END OF MEMBER ORBITV      299 RECORDS ********
```

OUTGEN JSS1493.GTDS.UPDATE.FORT

```
      SUBROUTINE OUTGEN                                                      00000010
C                                                                            00000020
C     VERSION OF JUNE 1979                                                   00000030
C                                                                            00000040
C        FORTRAN SUBROUTINE FOR THE IBM 360                                  00000050
C                                                                            00000060
C     PURPOSE                                                                00000070
C        GENERATE THE ORBIT GENERATOR OPTIONS IN THE INITIAL CONDITIONS      00000080
C        REPORT TO THE ON-LINE PRINTER                                       00000090
C                                                                            00000100
C     USAGE                                                                  00000110
C        CALL OUTGEN                                                         00000120
C                                                                            00000130
C     INPUT                                                                  00000140
C        FROM LABELED COMMONS  /FILES /, /FRC   /, /SECTN /, /SWITCH/,       00000150
C                              /WORKER/, /XLABEL/                            00000160
C                                                                            00000170
C     OUTPUT                                                                 00000180
C        ORBIT GENERATOR TYPE                                                00000190
C        STATE PARTIALS TYPE                                                 00000200
C        NON-STATE PARTIALS                                                  00000210
C                                                                            00000220
C     REMARKS                                                                00000230
C        SUBROUTINE OUTGEN WAS AT ONE TIME PART OF SUBROUTINE OUTOG1.        00000240
C        THIS CODING HAS BEEN REMOVED TO SHORTEN OUTOG1.                     00000250
C                                                                            00000260
C     FUNCTIONS AND SUBROUTINES REQUIRED                                     00000270
C        NONE                                                                00000280
C                                                                            00000290
C     METHOD                                                                 00000300
C        SUBROUTINE OUTGEN OUTPUTS ALL INFORMATION PERTAINING TO THE         00000310
C        ORBIT GENERATOR OPTIONS FROM VARIABLES LOCATED IN BLANK COMMON      00000320
C                                                                            00000330
C     PROGRAMMER                                                             00000340
C        J. TODD     GSFC     CODE 552                                       00000350
C                                                                            00000360
C     PROGRAM MODIFICATIONS                                                  00000370
C        1.  DATE  :  05/17/74                                               00000380
C            NAME  :  J. M. DECICCO                                          00000390
C            REASON:  PROPERLY REPORT ANALYTIC AND ATTITUDE PARTIALS         00000400
C        2.  DATE  :  08/01/74                                               00000410
C            NAME  :  G. SPEARS (CSC)                                        00000420
C            REASON:  CORRECT PRINTOUT OF LINE 'PREGENERATED ORBIT FILE'     00000430
C        3.  DATE  :  01/11/77                                               00000440
C            NAME  :  S. WATSON, R. GORDON (GSFC)                            00000450
C            REASON:  INCLUDE PROPER COMMENTS AS TO THE FOLLOWING:           00000460
C                     (1) WHETHER RESIDUAL 4TH HARMONIC IS INCLUDED IN       00000470
C                     VINTI ORBIT GENERATOR                                  00000480
C                     (2) WHETHER AVERAGED OR SECULAR RATES ARE USED FOR     00000490
C                     PARTIALS FOR VINTI ORBIT GENERATOR                     00000500
C                     (3) WHICH OSCULATING TO MEAN CONVERSION IS USED        00000510
C        4.  DATE  :  02/01/78                                               00000520
C            NAME  :  D. G. SOSKEY (CSC)                                     00000530
C            REASON:  CORRECT NPQ LOGIC                                      00000540
C        5.  DATE  :  02/01/78                                               00000550
C            NAME  :  K. AARONSON (CSC)                                      00000560
C            REASON:  CORRECT PRINT ALIGNMENT OF ORBIT GENERATOR TYPE        00000570
```

```
C        6.  DATE   :  05/15/78                                              00000580
C            NAME   :  K. AARONSON (CSC)                                     00000590
C            REASON:   INCLUDE ORBIT GENERATOR TYPE FOR FILTER               00000600
C        7.  DATE   :  06/21/79                                              00000610
C            NAME   :  CAPTAIN JEFFREY S. SHAVER, USAF                       00000620
C            REASON:   (1) CORRECT LOGIC FOR PRINTOUT OF TYPE OF VOP         00000630
C                      PARAMETER FOR THE CASE OF AVERAGED KEPLERIAN          00000640
C                      ELEMENTS                                              00000650
C                      (2) INCLUDE OPTION TO PRINT 'CHEBYSHEV INTEG-         00000660
C                      RATION' WHEN THE PICARD-CHEBYSHEV INTEGRATOR          00000670
C                      IS USED IN CONJUNCTION WITH THE VOP ORBIT             00000680
C                      GENERATOR OPTION                                      00000690
C                                                                            00000700
C*************************   START PROGRAM   *************************00000710
C                                                                            00000720
      IMPLICIT REAL*8(A-H,O-Z)                                               00000730
C                                                                            00000740
C * * * * * * * * *          COMMON BLOCKS             * * * * * * * * *00000750
C                                                                            00000760
      COMMON/BRWOPT/ DUMR1       , IDSEC       , IDOSM                     00000770
C                                                                            00000780
      COMMON/DCINPT/ DFDCIN(300) , INDCIN(200)                               00000790
C                                                                            00000800
      COMMON/FILES / INFILE(100)                                             00000810
C                                                                            00000820
      COMMON/FRC   / DPFRC(1300) , INFRC(38)                                 00000830
C                                                                            00000840
      COMMON/NPQBLK/ TQ2(14)     , TNPQ2(14)   , NPQQV(2,14) ,               00000850
     *               INDPRT(14)  , NPQP2       , NPQP3       ,              00000860
     *               NPQCT       , NPQ2S       , NPQ3S       ,              00000870
     *               NPQSLV                                                  00000880
C                                                                            00000890
      COMMON/NUMSEC/ DUM1        , DUM2        , DUM3        ,              00000900
     *               DUM4        , IMNRTE                                    00000910
C                                                                            00000920
      COMMON/R4COM / DPR4(20)    , INR4(5)                                   00000930
C                                                                            00000940
      COMMON/SECTN / DPSECT(675) , INSECT(325)                               00000950
C                                                                            00000960
      COMMON/SWITCH/ INSW(200)                                               00000970
C                                                                            00000980
      COMMON/WORKER/ DPWORK(3264), INWORK(13)                                00000990
C                                                                            00001000
      COMMON/XLABEL/ DPLABL(200) , INLABL(150)                               00001010
C                                                                            00001020
C * * * * * * * * *      DIMENSION STATEMENTS      * * * * * * * * * *00001030
C                                                                            00001040
      DIMENSION      TNPQ(14)    , ANPQ(2,14)  , SNPQ(2,14)   ,             00001050
     *               TQ(14)                                                  00001060
C                                                                            00001070
      DIMENSION      R4DATA(12)  , VNRATE(2)   , CHBINT(4)                  00001080
C                                                                            00001090
      DIMENSION      CRDTYP(10)  , IND(100)    , INDSEC(30,10),             00001100
     *               ISPHER(6)   , ISTATE(6)   , KSTATE(6)    ,             00001110
     *               LEMENT(15)  , PART(2)     , STPART(6)    ,             00001120
     *               TYPEPH(104) , TYPVAR(60)  , NDEG(20)     ,             00001130
     *               NORD(20)    , PLANET(11)  , OSNIOR(5)    ,             00001140
```

```
     *               OSMEAN(6)   , OSMKEP(6)   , MBOD(3)                00001150
C                                                                        00001160
C * * * * * * * * *     EQUIVALENCE STATEMENTS     * * * * * * * * * *00001170
C                                                                        00001180
C     FROM COMMON/DCINPT/:                                               00001190
      EQUIVALENCE (INDCIN(187),MBOD(1)  ), (INDCIN(190) ,MAXBOD  )       00001200
C                                                                        00001210
C     FROM COMMON/FILES /:                                               00001220
      EQUIVALENCE (INFILE(22) ,NOUT     )                                00001230
C                                                                        00001240
C     FROM COMMON/FRC   /:                                               00001250
      EQUIVALENCE (DPFRC(1092),TNPQ(1)  ), (DPFRC(1106),ANPQ(1,1)),      00001260
     *            (DPFRC(1134),SNPQ(1,1)), (DPFRC(1162),TQ(1)    ),      00001270
     *            (INFRC(5)   ,NVMAX    ), (INFRC(6)   ,MVMAX    )      00001280
C                                                                        00001290
C     FROM COMMON/R4COM /:                                               00001300
      EQUIVALENCE (INR4(1)    ,IRES4    )                                00001310
C                                                                        00001320
C     FROM COMMON/SECTN /:                                               00001330
      EQUIVALENCE (INSECT(11) ,INDSEC(1,1))                              00001340
C                                                                        00001350
C     FROM COMMON/SWITCH/:                                               00001360
      EQUIVALENCE (INSW(1)    ,IND(1)   ), (INSW(105)  ,NCNM     ),      00001370
     *            (INSW(106)  ,NSNM     ), (INSW(107)  ,NDEG(1)  ),      00001380
     *            (INSW(127)  ,MORD(1)  ), (INSW(154)  ,NSTATE   ),      00001390
     *            (INSW(155)  ,KSTATE(1)), (INSW(20)   ,ICHEBY   )      00001400
C                                                                        00001410
C     FROM COMMON/WORKER/:                                               00001420
      EQUIVALENCE (INWORK(5)  ,NEQ      )                                00001430
C                                                                        00001440
C     FROM COMMON/XLABEL/:                                               00001450
      EQUIVALENCE (INLABL(13) ,ISTATE(1)), (INLABL(19) ,LEMENT(1)),      00001460
     *            (INLABL(34) ,ISPHER(1)), (DPLABL(1)  ,PLANET(1))       00001470
C                                                                        00001480
C * * * * * * * * *       INPUT DATA BLOCKS        * * * * * * * * * * 00001490
C                                                                        00001500
      DATA CRDTYP/'CARTESIA','KEPLERIA','SPHERICA','DODS','N','N','L',   00001510
     * ' ','NONE OUT','PUT'/                                            00001520
C                                                                        00001530
      DATA STPART / 'NUMERICA','L INTEGR','ATION',                       00001540
     *              'ANALYTIC',' COMPUTA','TION ' /                      00001550
C                                                                        00001560
      DATA TYPEPH/'NUMERICA','L INTEGR','ATION OF',' REGULAR','IZED COW'00001570
     *, 'ELL EQUA','TIONS OF',' MOTION',                                 00001580
     *  'NUMERICA','L INTEGR','ATION OF',' COWELL ','EQUATION',         00001590
     *  'S OF MOT','ION',' ',                                           00001600
     *  'REGULAR ','BROUWER',' ',' ',' ',' ',' ',' ',                   00001610
     *  'BROUWER-','LYDANNE',' ',' ',' ',' ',' ',' ',                   00001620
     *  'PREGENER','ATED  ''O','RBIT'' FI','LE',' ',' ',' ',' ',        00001630
     *  'NUMERICA','L INTEGR','ATION OF',' EQUATIO','NS OF MO',         00001640
     *  'TION USI','NG K-S V','ARIABLES',                               00001650
     *  'GENERALI','ZED INTE','GRATOR',' ',' ',' ',' ',' ',            00001660
     *  'CHEBYSHE','V SERIES',' INTEGRA','TOR',' ',' ',' ',' ',        00001670
     *  '        ',',        ','ATION OF',' VARIATI','ON OF PA',       00001680
     *  'RAMETERS',' EQUATIO','NS OF MO','TION USI','NG ',             00001690
     * ' ', ' ', ' ', ' ', ' ', ' ', 'VINTI 3R', 'D ORDER ', 'ANALYTIC',00001700
     * ' ORBIT T','HEORY   ',' ',' ',' ',                              00001710
```

```
     * 'BROUWER-','GORDON ',' ',' ',' ',' ',' ',' ',                    00001720
     * ' 4 TH OR','DER RUNG','E-KUTTA ','FEHLBERG',' ',' ',' ',' '/     00001730
C                                                                       00001740
      DATA TYPVAR/  ' ''DALLAS',''' VARIAB','LES',' ',' ',' ',          00001750
     *  'EQUINOCT','IAL (MEA','N LONG) ','VARIABLE','S       ',' ',    00001760
     *  'EQUINOCT','IAL (ECC','EN LONG)',' VARIABL','ES      ',' ',    00001770
     *  'IDEAL (T','RUE LONG',') VARIAB','LES',' ',' ',                 00001780
     *  'IDEAL (E','CCEN LON','G) VARIA','BLES',' ',' ',                00001790
     *  'KS VARIA','BLES',' ',' ',' ',' ',                              00001800
     *  'DS VARIA','BLES',' ',' ',' ',' ',                              00001810
     *  'KEPLERIA','N VARIAB','LES     ',' ',' ',' ',                   00001820
     *  'AVERAGED',' EQUINOC','TIAL (ME','AN LONG)',' VARIABL','ES',    00001830
     *  'AVERAGED',' KEPLERI','AN VARIA','BLES',' ',' '/                00001840
C                                                                       00001850
      DATA R4DATA / 'WITH INC', 'LUSION O', 'F THE RE',                 00001860
     *              'SIDUAL 4', 'TH HARMO', 'NIC     ',                 00001870
     *              'WITHOUT ', 'INCLUSIO', 'N OF THE',                 00001880
     *              ' RESIDUA', 'L 4TH HA', 'RMONIC  '  /               00001890
C                                                                       00001900
      DATA R8BLNK /'        ' /                                         00001910
C                                                                       00001920
      DATA R3BLNK /'   ' /                                              00001930
C                                                                       00001940
      DATA VNRATE / 'AVERAGE ', 'SECULAR ' /                            00001950
C                                                                       00001960
      DATA OSMEAN/'CARTESIA','N SPACE ','OSCULATI','NG TO ME','AN CONV',00001970
     *            'ERSION  '/                                           00001980
C                                                                       00001990
      DATA OSMKEP/'KEPLERIA','N SPACE ','OSCULATI','NG TO ME','AN CONV',00002000
     *            'ERSION  '/                                           00002010
C                                                                       00002020
      DATA OSM1OR/'1ST ORDE','R OSCULA','TING TO ','MEAN CON','VERSION'/00002030
C                                                                       00002040
      DATA SECULR/'SECULAR'/                                            00002050
C                                                                       00002060
      DATA PART  /'C' , 'S'/                                            00002070
C                                                                       00002080
      DATA CHBINT/'CHEBYSHE','V INTEGR','NUMERICA','L INTEGR'/          00002090
C                                                                       00002100
      WRITE(NOUT,2023)                                                  00002110
      IF(IND(1).NE.10) GO TO 19                                         00002120
      JEND = 104                                                        00002130
      GO TO 24                                                          00002140
   19 CONTINUE                                                          00002150
      IF(IND(1).EQ.11) GO TO 34                                         00002160
      IF(IND(1)-5) 20,25,23                                             00002170
   20 JEND = 8 * IND(1)                                                 00002180
      GO TO 24                                                          00002190
   23 JEND =  8 * (IND(1)-1)                                            00002200
      IF (IND(1) .NE. 12) GO TO 24                                      00002210
      JSTART = 65                                                       00002220
      JEND = 80                                                         00002230
      DO 80 I = 1 , 8                                                   00002240
      TYPEPH(64+I) = TYPEPH(JEND+I)                                     00002250
   80 CONTINUE                                                          00002260
      TYPEPH(73) = R8BLNK                                               00002270
      TYPEPH(74) = R3BLNK                                               00002280
```

```
      J = 6 * (IRES4 - 1)                                                   00002290
      DO 85 I = 1, 6                                                        00002300
      TYPEPH(74+I) = R4DATA(I+J)                                            00002310
   85 CONTINUE                                                              00002320
      GO TO 36                                                              00002330
   34 CONTINUE                                                              00002340
      JEND = 96                                                             00002350
   24 JSTART = JEND-7                                                       00002360
      GO TO 36                                                              00002370
   25 JSTART = 65                                                           00002380
      JEND = 80                                                             00002390
      IVAR = IND(3)                                                         00002400
      IF(IVAR.GE.12) IVAR = IVAR-3                                          00002410
      IF(IND(3).EQ.18) IVAR = 10                                            00002420
      IVAR = 6*IVAR - 6                                                     00002430
      DO 35 I=1,6                                                           00002440
   35 TYPEPH(74 +I) = TYPVAR(IVAR+I)                                        00002450
      IF(ICHEBY.EQ.2) GO TO 360                                             00002460
      TYPEPH(65) = CHBINT(1)                                                00002470
      TYPEPH(66) = CHBINT(2)                                                00002480
      GO TO 36                                                              00002490
  360 TYPEPH(65) = CHBINT(3)                                                00002500
      TYPEPH(66) = CHBINT(4)                                                00002510
   36 WRITE(NOUT,2043) (TYPEPH(I),I=JSTART,JEND)                            00002520
      IF(IND(1).NE.3.AND.IND(1).NE.4.AND.IND(1).NE.11)GO TO 37              00002530
      IF(IDOSM.EQ.0)WRITE(NOUT,2045)OSMEAN                                  00002540
      IF(IDOSM.EQ.1)WRITE(NOUT,2045)OSMKEP                                  00002550
      IF(IDOSM.EQ.2)WRITE(NOUT,2045)OSM1CR                                  00002560
      IF(IDSEC.EQ.1)WRITE(NOUT,2044)SECULR                                  00002570
   37 CONTINUE                                                              00002580
C                                                                           00002590
C     PRINTOUT NPQ CORRECTIONS                                              00002600
C                                                                           00002610
      IF(IND(1).NE.4.OR.NPQCT.EQ.0) GO TO 39                                00002620
   38 WRITE(6,2144)                                                         00002630
      IF(NPQP2.EQ.0) GO TO 61                                               00002640
      DO 160 I=1,NPQP2                                                      00002650
      ROUT=TNPQ(I)                                                          00002660
      IF(IND(48).EQ.2) ROUT=TNPQ2(I)                                        00002670
      IP=NPQQV(1,I)                                                         00002680
      IQ=NPQQV(2,I)                                                         00002690
      R2=SNPQ(1,I)                                                          00002700
      IF(INDPRT(I).EQ.1) WRITE(6,2145) IP,IQ,ROUT,R2                        00002710
      IF(INDPRT(I).EQ.2) WRITE(6,2245) IP,IQ,ROUT,R2                        00002720
 2145 FORMAT(41X,'N',I2,',',I2,2X,'(APPLY ONLY)',4X,2G21.14,/)              00002730
 2245 FORMAT(41X,'N',I2,',',I2,2X,'(SOLVE & APPLY) ',2G21.14,/)             00002740
  160 CONTINUE                                                              00002750
   61 IF(NPQP3.EQ.0) GO TO 39                                               00002760
      DO 62 I=1,NPQP3                                                       00002770
      K=I+NPQP2                                                             00002780
      ROUT=TNPQ(K)                                                          00002790
      IF(IND(48).EQ.2) ROUT=TNPQ2(K)                                        00002800
      IP=NPQQV(1,K)                                                         00002810
      IQ=NPQQV(2,K)                                                         00002820
      R2=SNPQ(2,K)                                                          00002830
      IF(INDPRT(K).EQ.1) WRITE(6,2145) IP,IQ,ROUT,R2                        00002840
      IF(INDPRT(K).EQ.2) WRITE(6,2245) IP,IQ,ROUT,R2                        00002850
```

```
   62 CONTINUE                                                                  00002860
   39 CONTINUE                                                                  00002870
C     CHECK FOR KINDS OF PARTIALS REQUIRED -                                   00002880
C              STATE     DRAG      SOLRAD  HARMONICS   THRUST   ATTITUDE     00002890
      ISUM = IND(40) + IND(41) + IND(42) + IND(43) + IND(44) + IND(19)         00002900
     *  + INSW(102) + INSW(103)                                                00002910
      IF ( ISUM .LT. 16 )  GO TO 33                                           00002920
      WRITE(NOUT,2034)                                                        00002930
      GO TO 999                                                               00002940
   33 IF(IND(40).EQ.1) GO TO 31                                               00002950
      WRITE(NOUT,2049)                                                        00002960
      GO TO 42                                                                00002970
   31 J1 = IND(39)                                                            00002980
      J2 = J1 + 4                                                             00002990
      I1 = IND(4)**2                                                          00003000
      I2 = I1 + 1                                                             00003010
      I3 = I2 + 1                                                             00003020
      WRITE(NOUT,2037) CRDTYP(J1),CRDTYP(J2),STPART(I1),STPART(I2),           00003030
     1  STPART(I3)                                                            00003040
      IF (IND(1) .NE. 12) GO TO 32                                            00003050
      WRITE(NOUT, 2030) VNRATE(IMNRTE)                                        00003060
   32 CONTINUE                                                                00003070
      IF(IND(39).EQ.4) GO TO 29                                               00003080
      IF(NSTATE.EQ.6)   GO TO 40                                              00003090
      KGOTO = IND(39)                                                         00003100
      GO TO (26,27,28   ),KGOTO                                               00003110
   26 WRITE(NOUT,2135) (ISTATE(KSTATE(I)),I=1,NSTATE)                         00003120
      GO TO 40                                                                00003130
   27 WRITE(NOUT,2135) (LEMENT(KSTATE(I)),I=1,NSTATE)                         00003140
      GO TO 40                                                                00003150
   28 WRITE(NOUT,2135) (ISPHER(KSTATE(I)),I=1,NSTATE)                         00003160
      GO TO 40                                                                00003170
   29 IF(KSTATE(NSTATE).EQ.10)GO TO 30                                        00003180
      WRITE(NOUT,2235)(KSTATE(I),I=1,NSTATE)                                  00003190
      GO TO 40                                                                00003200
   30 KSTATE(NSTATE) = 19                                                     00003210
      WRITE(NOUT,2235)(KSTATE(I),I=1,NSTATE)                                  00003220
      KSTATE(NSTATE)= 10                                                      00003230
   40 CONTINUE                                                                00003240
      IF(IND(4).EQ.2) GO TO 42                                                00003250
      WRITE(NOUT,2046) NVMAX,NVMAX                                            00003260
   42 WRITE(NOUT,2047)                                                        00003270
      IF(IND(4).EQ.1.AND.NEQ.GT.NSTATE) GO TO 45                              00003280
      IF(IND(4).EQ.2.AND.NEQ.GT.0) GO TO 45                                   00003290
      WRITE(NOUT,2048)                                                        00003300
      GO TO 999                                                               00003310
   45 IF(IND(41).EQ.2) GO TO 47                                               00003320
      WRITE(NOUT,2035)                                                        00003330
   47 IF(IND(42).EQ.2) GO TO 48                                               00003340
      WRITE(NOUT,2038)                                                        00003350
   48 IF(IND(43).EQ.2) GO TO 50                                               00003360
      NPLUS = NCNM + NSNM                                                     00003370
      WRITE(NOUT,2039) PLANET(INDSEC(1,1))                                    00003380
      I = 0                                                                   00003390
      J = 1                                                                   00003400
   49 I = I + 1                                                               00003410
      IF(I.GT.NPLUS) GO TO  50                                                00003420
```

```
      IF(I.GT.NCNM) J=2                                                 00003430
      WRITE(NOUT,2036) PART(J),NDEG(I),MORD(I)                          00003440
      GO TO 49                                                          00003450
   50 IF(IND(44).EQ.2) GO TO 55                                         00003460
      WRITE(NOUT,2040)                                                  00003470
   55 IF ( IND(19) .EQ. 2 )  GO TO 60                                   00003480
      WRITE(NOUT,2055)                                                  00003490
   60 IF ( INSW(102) .NE. 1 )  GO TO 70                                 00003500
      WRITE( NOUT, 2060) ( MBOD(I), I=1,MAXBOD )                        00003510
   70 IF ( INSW(103) .NE. 1 )  GO TO 999                                00003520
      WRITE( NOUT, 2070 )                                               00003530
  999 RETURN                                                            00003540
 2023 FORMAT('0',/,'0',64('* '))                                        00003550
 2030 FORMAT(1H+, 42X, 'USING ', A8, ' RATES')                          00003560
 2034 FORMAT('0',24('* '),'NO PARTIALS BEING COMPUTED',24('* '),/)      00003570
 2035 FORMAT(55X,'DRAG PARTIALS')                                       00003580
 2135 FORMAT(45X,6(A4,3X))                                              00003590
 2235 FORMAT(45X,6('X',I2))                                             00003600
 2036 FORMAT(T60,A1,'(',I2,',',I2,')')                                  00003610
 2037 FORMAT(/,3X,'STATE PARTIALS TYPE-',/,3X,A8,A1,' USING ',2A8,A5)   00003620
 2038 FORMAT(55X,'RADIATION PARTIALS')                                  00003630
 2039 FORMAT(55X,'POTENTIAL PARTIALS(BODY = ',A8,')')                   00003640
 2040 FORMAT(55X,'THRUST PARTIALS')                                     00003650
 2043 FORMAT(64(' *'),/,26(' *'),'ORBIT GENERATOR OPTIONS',26(' *'),    00003660
     1 //,3X,'ORBIT GENERATOR TYPE-',/,3X,9A8,A3,6A8)                   00003670
 2044 FORMAT(1H+,17X,'--',6A8)                                          00003680
 2045 FORMAT(1H+,27X,'--',6A8)                                          00003690
 2046 FORMAT(35X,'A ',I2,' BY ',I2,' GRAVITY FIELD USED IN COMPUTATION 00003700
     1F PARTIALS')                                                      00003710
 2047 FORMAT(/,3X,'NON-STATE PARTIALS (COMPUTED BY NUMERICAL INTEGRATION00003720
     1)-')                                                              00003730
 2048 FORMAT(50X,'NO NON-STATE PARTIALS CONSIDERED')                    00003740
 2049 FORMAT(10X,'NO STATE PARTIALS CONSIDERED')                        00003750
 2055 FORMAT(55X,'ATTITUDE PARTIALS')                                   00003760
 2060 FORMAT( 45X, 'GM PARTIALS OF BODY ',3( I4, 4X ) )                 00003770
 2070 FORMAT( 45X, 'LUMPED GEOPOTENTIAL PARTIAL' )                      00003780
 2144 FORMAT(/,37X,'N(P,Q) TERMS APPLIED',9X,'APRIORI TIME',9X,         00003790
     *       'APRIORI VALUE')                                           00003800
      END                                                               00003810

******** END OF MEMBER CUTGEN     381 RECORDS ********
```

```
OUTOG1              JSS1493.GTDS.UPDATE.FORT

      SUBROUTINE OUTOG1                                                      00000010
C                                                                            00000020
C     VERSION OF NOV. 1977                                                   00000030
C                                                                            00000040
C         FORTRAN SUBROUTINE FOR THE IBM 360                                 00000050
C                                                                            00000060
C     PURPOSE                                                                00000070
C         GENERATE THE ORBIT GENERATOR INITIAL CONDITIONS REPORT TO          00000080
C         THE ON-LINE PRINTER                                                00000090
C                                                                            00000100
C     USAGE                                                                  00000110
C         CALL OUTOG1                                                        00000120
C                                                                            00000130
C     INPUT                                                                  00000140
C         FROM LABELED COMMONS  /DCINT /, /EAINT /, /FILES /, /FRC    /,     00000150
C                               /INTEG /, /SATPOS/, /SECTN /, /SWITCH/,      00000160
C                               /TITLE /, /WORKER/, /XLABEL/                 00000170
C                                                                            00000180
C     OUTPUT                                                                 00000190
C         HEADING                                                            00000200
C         CURRENT DATE AND TIME                                              00000210
C         SCOPE IN USE OR NOT                                                00000220
C         TYPE OF RUN                                                        00000230
C         SATELLITE NAME                                                     00000240
C         SATELLITE NUMBER                                                   00000250
C         SATELLITE DIMENSIONS                                               00000260
C         START AND END TIME                                                 00000270
C         PRINT INTERVAL                                                     00000280
C         TIME SPAN OF RUN                                                   00000290
C         CENTRAL BODY                                                       00000300
C         TIME AND COORDINATE SYSTEMS                                        00000310
C         CARTESIAN COORDINATES                                              00000320
C         CLASSICAL ELEMENTS                                                 00000330
C         SPHERICAL COORDINATES                                              00000340
C         INTEGRATION ---                                                    00000350
C                       TRUNCATION ERROR BOUNDS                              00000360
C                       CORRECTION TOLERANCES                                00000370
C                       MINIMUM STEPSIZE                                     00000380
C                       STEP MODE                                            00000390
C                       ORDER                                                00000400
C         TOTAL NUMBER OF SECTIONS                                           00000410
C         MODEL NUMBER USED AS BASIC MODEL                                   00000420
C         TOLERANCES ON CROSSOVER CONDITIONS FOR EACH SECTION                00000430
C         NAME OF ATMOSPHERIC DENSITY MODEL                                  00000440
C         NAME OF EARTH POTENTIAL FIELD MODEL                                00000450
C         NAME OF LUNAR POTENTIAL FIELD MODEL                                00000460
C         BY SECTION---                                                      00000470
C                     INTEGRATION STEP MODE                                  00000480
C                     HARMONICS                                              00000490
C                     CENTRAL BODY                                           00000500
C                     FIRST AND SECOND NON-CENTRAL BODIES                    00000510
C                     FORCE OPTIONS CONSIDERED                               00000520
C                     CROSSOVER CONDITIONS                                   00000530
C         TOTAL NUMBER OF MANEUVERS                                          00000540
C         FOR EACH MANEUVER---                                               00000550
C                           TIME                                             00000560
C                           CHANGE IN MASS                                   00000570
```

```
OUTOG1              JSS1493.GTDS.UPDATE.FORT

C                              VELOCITY INCREMENT                          00000580
C         PHYSICAL CONSTANTS(FOR BODIES CONSIDERED)                        00000590
C         PARTIALS CONSIDERED                                              00000600
C         INITIAL COVARIANCE MATRIX                                        00000610
C         OUTPUT OPTIONS                                                   00000620
C                                                                          00000630
C                                                                          00000640
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                        00000650
C         SUBROUTINE CDATE                                                 00000660
C         SUBROUTINE ELEM                                                  00000670
C         SUBROUTINE EVAL                                                  00000680
C         SUBROUTINE HARM                                                  00000690
C         SUBROUTINE CGCROS                                                00000700
C         SUBROUTINE OUTGEN                                                00000705
C         SUBROUTINE RPDAT0                                                00000710
C         SUBROUTINE RYMDI                                                 00000720
C         SUBROUTINE UPPAGE                                                00000730
C         REAL FUNCTION DJUL                                               00000740
C                                                                          00000750
C     METHOD                                                               00000760
C         SUBROUTINE OUTOG1 OUTPUTS ALL INFORMATION PERTAINING TO THE      00000770
C                INITIAL CONDITIONS FROM VARIABLES LOCATED IN LABELED      00000780
C                COMMON.                                                   00000790
C         REAL FUNCTION DJUL IS USED TO CALCULATE THE JULIAN DATE.         00000800
C         SUBROUTINE CDATE IS CALLED TO CONVERT FROM THE JULIAN DATE       00000810
C                TO THE GREGORIAN DATE.                                    00000820
C         SUBROUTINE ELEM IS CALLED TO CALCULATE THE CLASSICAL             00000830
C                ELEMENTS AND THE FLIGHT PARAMETERS.                       00000840
C         SUBROUTINE EVAL CALCULATES THE GHA FOR EARTH                     00000850
C         SUBROUTINE HARM PRINTS OUT THE HARMONICS ARRAY                   00000860
C         SUBROUTINE CGCROS IS CALLED TO OUTPUT THE SECTIONING             00000870
C                CROSSOVER CONDITIONS.                                     00000880
C         SUBROUTINE OUTGEN IS CALLED TO PRINT THE ORBIT GENERATOR         00000882
C                OPTIONS TO THE LINE PRINTER.                              00000884
C         SUBROUTINE RPDAT0 IS CALLED TO RETRIEVE THE CURRENT DATE         00000890
C                AND TIME.                                                 00000900
C         SUBROUTINE RYMDI IS CALLED TO UNPACK ANY 6-DIGIT DATE OR         00000910
C                TIME.                                                     00000920
C         SUBROUTINE UPPAGE IS CALLED TO PRINT HEADING AND PAGE            00000930
C                NUMBER AND TO REGULATE THE NUMBER OF LINES PER PAGE       00000940
C                IN PRINTED OUTPUT.                                        00000950
C                                                                          00000960
C     PROGRAMMER                                                           00000970
C         J. TODD       GSFC        CODE 553                             00000980
C                                                                          00000990
C     PROGRAM MODIFICATIONS                                                00001000
C         1.  DATE:     04/22/71                                         00001010
C             NAME:     J. TODD                                          00001020
C             REASON:   ORIGINAL SUBROUTINE                              00001030
C         2.  DATE:     08/13/71                                         00001040
C             NAME:     J. WORTHY, GSFC, CODE 553                        00001050
C             REASON:   ADDED SUBROUTINE 'UPPAGE'; DEFINED INTEGRATION   00001060
C                       PARAMETERS; REFORMATTED TEST FOR INTEGRATION      00001070
C                       STEPMODE; ADDED DRAG AND HARMONIC COEFFICIENTS    00001080
C                       TO PHYSICAL CONSTANTS                             00001090
C         3.  DATE:     01/03/72                                         00001100
C             NAME:     F. L. PERAZZOLI, GSFC, CODE 552                  00001110
```

```
C            REASON:  UNKNOWN                                                  00001120
C         4. DATE:    03/30/72                                                 00001130
C            NAME:    F. MCGARRY                                               00001140
C            REASON:  PRINT ORBIT THEORY PARAMETERS AS WELL AS PRINT           00001150
C                     START TIME; BLOCK OF INPUT OPTIONS ALSO EXTENDED         00001160
C                     IN PRINTOUT                                              00001170
C         5. DATE:    01/11/72                                                 00001180
C            NAME:    A. WELKER                                                00001190
C            REASON:  ADD LOGIC TO WRITE DATA SIM REPORT                       00001200
C         6. DATE:    01/14/74                                                 00001210
C            NAME:    B. DIXON                                                 00001220
C            REASON:  ALLOW FOR HIGHER ORDER INTEGRATION                       00001230
C         7. DATE:    04/01/74                                                 00001240
C            NAME:    G. SPEARS, CSC                                           00001250
C            REASON:  DO THE FOLLOWING DURING ERROR ANALYSIS RUN:              00001260
C                     CALL SUBROUTINE 'CUTTEA' INSTEAD OF 'CUTTIC'             00001270
C                     SKIP PRINTING INT. PARAMS. AND O.G. OPTIONS              00001280
C                     PRINT INITIAL STATE COV. MATRIX DESPITE IND(99)          00001290
C         8. DATE:    02/01/75                                                 00001300
C            NAME:    M. FAN, CSC                                              00001310
C            REASON:  TO PRINT OVERRIDE INITIAL STATE COVARIANCE MATRIX        00001320
C         9. DATE:    07/01/75                                                 00001330
C            NAME:    S. N. WEI AND R. DESAI, CSC                              00001340
C            REASON:  ADDED INITIAL DATA FOR LANDMARK MODELLING                00001350
C        10. DATE:    07/01/76                                                 00001360
C            NAME:    J. FEIN, CSC                                             00001370
C            REASON:  ADDED DATA FUL LANDMARK + EARTH EDGE OBSERVABLES         00001380
C        11. DATE:    05/19/77                                                 00001390
C            NAME:    UNKNOWN                                                  00001400
C            REASON:  ADD MESSAGES FOR THE KALMAN FILTER                       00001410
C        12. DATE:    10/24/77                                                 00001420
C            NAME:    K. AARONSON, CSC                                         00001430
C            REASON:  INSERTED CALL TO 'GVCVL' TO INITIALIZE TITLE             00001440
C                     ARRAY FOR PRINTING COVARIANCE MATRIX                     00001450
C        13. DATE:    01/11/77                                                 00001460
C            NAME:    R. GORDON AND S. WATSON, GSFC                            00001470
C            REASON:  AVOID PRINTOUT OF INTEGRATION PARAMETERS FOR             00001480
C                     THE ANALYTIC THEORIES                                    00001490
C        14. DATE:    06/21/79                                                 00001500
C            NAME:    CAPTAIN JEFFREY S. SHAVER, USAF                          00001510
C            REASON:  CORRECT COMPILATION ERROR DUE TO MISDIMENSION            00001520
C                     OF FBIAS ARRAY; CLEAN UP COMMENTS.                       00001530
C                                                                              00001540
C                                                                              00001550
C*********************       START PROGRAM       ************************00001560
C                                                                              00001570
      IMPLICIT REAL*8(A-H,O-Z)                                                 00001580
C                                                                              00001590
C * * * * * * * * *        REAL VARIABLES         * * * * * * * * * * *00001600
C                                                                              00001610
      REAL*4 SEC                                                               00001620
C                                                                              00001630
C * * * * * * * * *       LOGICAL VARIABLES       * * * * * * * * * * 00001640
C                                                                              00001650
      LOGICAL*1 LEAP  ,  LOVWT  ,  LDCIN  ,  ATTDAT  ,  AEARTH                 00001660
C                                                                              00001670
C * * * * * * * * *     DIMENSION STATEMENTS      * * * * * * * * * * *00001680
```

```
C                                                                       00001690
      DIMENSION  COVMAT(21) , IND(100)  , INDSEC(30,10) , TOL(10)    ,  00001700
     *           HEAD(9)    , IYMDHM(5) , SCOPE(7)      , HEAD1(30)  ,  00001710
     *           TYPVEQ(25) , MAXIT(5)  , MONTHS(12)    , TYPSTP(30) ,  00001720
     *           RATH(5)    , DCTH(5)   , ROLL(5)       , PBIAS(8)      00001730
C                                                                       00001740
      DIMENSION  SAE(3)     , SCON(3,3)                                 00001750
C                                                                       00001760
      DIMENSION  VCVTTL(50)                                             00001770
C                                                                       00001780
C * * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00001790
C                                                                       00001800
      COMMON/CONST / TWOPI        , PI        , DUM(2)       , RTD      00001810
C                                                                       00001820
      COMMON/DCINPT/ DPI(300)     , INTI(200) , LDCIN(72)               00001830
C                                                                       00001840
      COMMON/DCINT / DPDCT(432)   , INDCT(26)                           00001850
C                                                                       00001860
      COMMON/EAINT / DPEAP(60)    , INEAP(20) , LEAP(12)                00001870
C                                                                       00001880
      COMMON/FILES / IFILE(50)                                          00001890
C                                                                       00001900
      COMMON/FRC   / DPFRC(1300)  , INFRC(38)                           00001910
C                                                                       00001920
      COMMON/INTEG / DPITG(149)   , INITG(7)                            00001930
C                                                                       00001940
      COMMON/SATPOS/ DPSAT(100)   , INSAT(5)                            00001950
C                                                                       00001960
      COMMON/SECTN / DPSECT(675)  , INSECT(325)                         00001970
C                                                                       00001980
      COMMON/SWITCH/ ISWTCH(200)                                        00001990
C                                                                       00002000
      COMMON/THRUST/ DPTHR(150)   , INTHR(50)                           00002010
C                                                                       00002020
      COMMON/TITLE / HEADER(9,6)                                        00002030
C                                                                       00002040
      COMMON/WORKER/ DPWORK(3264) , INWORK(13)                          00002050
C                                                                       00002060
      COMMON/XLABEL/ DPLABS(200)  , INLABS(150)                         00002070
C                                                                       00002080
C * * * * * * * * * *    EQUIVALENCE STATEMENTS    * * * * * * * * * * * 00002090
C                                                                       00002100
C     FROM COMMON/DCINPT/:                                              00002110
      EQUIVALENCE    (LDCIN(59) , ATTDAT    ), (LDCIN(60)  , AEARTH  ), 00002120
     *               (DPI(239)  , SENSOR    ), (DPI(297)   , TESTEP  ), 00002130
     *               (INTI(182) , ILEVEL    )                           00002140
C                                                                       00002150
C     FROM COMMON/DCINT /:                                              00002160
      EQUIVALENCE    (DPDCT(178), COVMAT(1) )                          00002170
C                                                                       00002180
C     FROM COMMON/EAINT /:                                              00002190
      EQUIVALENCE    (LCVHT     , LEAP(12)  )                          00002200
C                                                                       00002210
C     FROM COMMON/FILES /:                                              00002220
      EQUIVALENCE    (IFILE(22) , NOUT      ), (IFILE(28)  , NSCO    )  00002230
C                                                                       00002240
C     FROM COMMON/FRC   /:                                              00002250
```

```
OUTOG1              JSS1493.GTDS.UPDATE.FORT

      EQUIVALENCE    (DPFRC(151), AREA      ), (DPFRC(152) , SCMASS  ), 00002260
     *               (DPFRC(154), CSUBDZ    ), (DPFRC(155) , RHO1    ), 00002270
     *               (DPFRC(159), CSUBR     )                           00002280
C                                                                        00002290
C     FROM COMMON/INTEG /:                                               00002300
      EQUIVALENCE    (DPITG(124), TOL(1)    ), (DPITG(123) , STPMIN  ), 00002310
     *               (INITG(1)  , N1        ),                          00002320
     *               (INITG(3)  , MAXIT(1)  ), (INITG(2)   , N2      ) 00002330
C                                                                        00002340
C     FROM COMMON/SATPOS/:                                               00002350
      EQUIVALENCE    (DPSAT(72) , REFJUL    ), (DPSAT(73)  , SATNAM  ), 00002360
     *               (INSAT(5)  , IDSAT     )                           00002370
C                                                                        00002380
C     FROM COMMON/SECTN /:                                               00002390
      EQUIVALENCE    (INSECT(11),INDSEC(1,1)), (INSECT(312), NTSECS  ) 00002400
C                                                                        00002410
C     FROM COMMON/SWITCH/:                                               00002420
      EQUIVALENCE    (ISWTCH(1) , IND(1)    ), (ISWTCH(148), LINES   ) 00002430
C                                                                        00002440
C     FROM COMMON/THRUST/:                                               00002450
      EQUIVALENCE    (DPTHR(16) , RATH(1)   ), (DPTHR(21)  , DCTH(1) ), 00002460
     *               (DPTHR(26) , ROLL(1)   ), (DPTHR(73)  , PBIAS(1)), 00002470
     *               (DPTHR(61) , SAE(1)    ), (DPTHR(64)  , SIP     ), 00002480
     *               (INTHR(28) , MODATT    ), (INTHR(30)  , LANDTY  ), 00002490
     *               (INTHR(16) , IPRY      ), (INTHR(10)  , KONFIG  ) 00002500
C                                                                        00002510
C     FROM COMMON/WORKER/:                                               00002520
      EQUIVALENCE    (INWORK(5) , NEQ       )                           00002530
C                                                                        00002540
C     FROM COMMON/XLABEL/:                                               00002550
      EQUIVALENCE    (INLABS(1) , MONTHS(1) ), (DPLABS(21) , VCVTTL(1)) 00002560
C                                                                        00002570
C * * * * * * * * *        INPUT DATA BLOCKS        * * * * * * * * * * 00002580
C                                                                        00002590
      DATA SCON  /'SPHERICA','L       ',','        ',                   00002600
     *            'CYLINDRI','CAL     ',','        ',                   00002610
     *            'CYLINDRI','CAL WITH',' PADDLES'/                    00002620
C                                                                        00002630
      DATA HEAD1 /'DIFFEREN','TIAL COR','RECTION',                       00002640
     *            'ORBIT GE','NERATION',' ',                           00002650
     *            'DATA SIM','ULATION ',' ',                           00002660
     *            'ORBIT CO','MPARE',' ',                              00002670
     *            'DATA MAN','AGEMENT',' ',                            00002680
     *            'PERMANEN','T FILE R','EPORT',                       00002690
     *            'ERROR AN','ALYSIS',' ',                             00002700
     *            'EARLY OR','BIT',' ',                                00002710
     *            ' ',' ',' ',                                         00002720
     *            'KALMAN F','ILTER',' '/                              00002730
C                                                                        00002740
      DATA TYPSTP/                                                       00002750
     *            'SINGLE P','OINT OFF',' GRID',' ',' ',' ','SUMMED C', 00002760
     *            'CWELL/AD','AMS FRED','ICT-PART','IAL CORR','ECT',   00002770
     *            'SUMMED A','DAMS FRE','DICTOR-C','ORRECTOR',' ',' ', 00002780
     *            'RUNGE-KU','TTA     ',' ',' ',' ',' ',               00002790
     *            'RUNGE-KU','TTA FEHL','BERG    ',' ',' ',' '/        00002800
C                                                                        00002810
      DATA TYPVEQ/                                                       00002820
```

```
     *            'SINGLE P','OINT OFF',' GRID',' ',' ',                  00002830
     *            'SUMMED C','OWELL/AD','AMS CORR','ECTOR-ON','LY',       00002840
     *            'SUMMED A','DAMS COR','RECTOR O','NLY',' ',             00002850
     *            'RUNGE-KU','TTA     ',' ',' ',' ',                      00002860
     *            'RUNGE-KU','TTA FEHL','BERG    ',' ',' '/               00002870
C                                                                         00002880
      DATA HEAD  /' ',' ',' ','GTDS INI','TIAL CON','DITIONS ',           00002890
     *            'REPORT',' ',' '/                                       00002900
C                                                                         00002910
      DATA SCOPE /'2250 ', ' '       ,'SCOPE'   ,'S  NOT  ','2250 '    , 00002920
     *            'AND 2260',' IN USE'/                                   00002930
C                                                                         00002940
C   WRITE HEADING                                                         00002950
C                                                                         00002960
      IF (LOVWT) GO TO 780                                                00002970
      CALL UPPAGE(60,1,HEAD)                                              00002980
      LINES = 0                                                           00002990
      CALL UPPAGE(13,1,HEAD)                                              00003000
      J3 = 3*IND(48)                                                      00003010
      J2 = J3 - 1                                                         00003020
      J1 = J2 - 1                                                         00003030
      WRITE(NOUT,2101) HEAD1(J1),HEAD1(J2),HEAD1(J3)                      00003040
      WRITE(NOUT,2000) (HEADER(I,1),I=1,9)                                00003050
C                                                                         00003060
C   RETRIEVE AND PRINT CURRENT DATE                                       00003070
C                                                                         00003080
      CALL RPDAT0(2,IYMDHM,SEC)                                           00003090
      IYMDHM(1)  = IYMDHM(1) +1900                                        00003100
      WRITE(NOUT,2001) MONTHS(IYMDHM(2)),IYMDHM(3),IYMDHM(1),             00003110
     1      IYMDHM(4),IYMDHM(5),SEC                                       00003120
C                                                                         00003130
C   WRITE SCOPE INDICATION                                                00003140
C                                                                         00003150
      J2=2*IND(45)                                                        00003160
      J1=J2-1                                                             00003170
      WRITE(NOUT,2003) SCOPE(J1),SCOPE(J2),SCOPE(7)                       00003180
C                                                                         00003190
C     WRITE THE LEVEL OF OUTPUT                                           00003200
C                                                                         00003210
      IF(IND(48).EQ.1)WRITE(NOUT,2002)ILEVEL                              00003220
C                                                                         00003230
C   WRITE SAT. NAME, ID, AREA AND MASS                                    00003240
C                                                                         00003250
C    CONVERT AREA TO  M**2 AND MASS TO GRAMS FOR THIS OUTPUT ONLY         00003260
C                                                                         00003270
      AREI = AREA*1.D6                                                    00003280
      WRITE(NOUT,2004) SATNAM                                             00003290
      WRITE(NOUT,2041) IDSAT                                              00003300
      K = KONFIG + 1                                                      00003310
      SCM = .5D0*SCMASS/(1.D0 + RHO1)                                     00003320
      WRITE (NOUT,2042)  (SCON(I,K),I=1,3)                                00003330
C                                                                         00003340
C   THE VALUE OF S DEPENDS ON THE DEFINITION OF SAE(1) IN AEROPR          00003350
C                                                                         00003360
      S=PI                                                                00003370
      IF(SIP.EQ.0.D0) S = S*CSUBDZ*.5                                     00003380
      S= 1.D3*DSQRT(2.D0*SAE(1)*SCM/S)                                    00003390
```

```
      J = 6                                                              00003400
      IF (K .GT. 1)  GO TO 2                                             00003410
      S = S*2.D0                                                         00003420
      WRITE (NOUT,2043)  S                                               00003430
      IF (SIP .EQ. 0.D0)  WRITE (NOUT,2047)  CSUBDZ                      00003440
      GO TO 8                                                            00003450
    2 CONTINUE                                                           00003460
      A = SAE(2)*SCM*1.5D3/S                                             00003470
      J = J + 1                                                          00003480
      WRITE (NOUT,2044)  S,A                                             00003490
      IF (K .LT. 3)  GO TO 8                                             00003500
      J = J + 1                                                          00003510
      A = SAE(3)*1.D6                                                    00003520
      S = DARSIN(SIP)*RTD                                                00003530
      WRITE (NOUT,2045)  A,S                                             00003540
    8 CONTINUE                                                           00003550
      WRITE (NOUT,2046)  SCMASS,RHO1,CSUBR,ARE1                          00003560
      CALL UPPAGE (J,1,HEAD)                                             00003570
C                                                                        00003580
C     WRITE LANDMARK RELATED INFORMATION                                 00003590
C                                                                        00003600
      IF (.NOT.ATTDAT ) GO TO 9                                          00003610
      WRITE(NOUT,3000)                                                   00003620
      A1=DPTHR(54)*RTD                                                   00003630
      A2=DPTHR(55)*RTD                                                   00003640
      IF(LANDTY.EQ.1) WRITE(NOUT,3100) A1,A2                             00003650
      A1 =SENSOR * RTD                                                   00003660
      IF(AEARTH)            WRITE (NOUT,3105) A1, TESTEP                 00003670
      IF(LANDTY.EQ.1.OR.LANDTY.EQ.2.AND.MODATT.EQ.2)WRITE(NOUT,3125)     00003680
      IF(LANDTY.EQ.2.AND.MODATT.EQ.1) WRITE(NOUT,3175)                   00003690
      IF(LANDTY.EQ.2.AND.IPRY.EQ.1)WRITE(NOUT,3185)                      00003700
      IF(LANDTY.EQ.2.AND.IPRY.EQ.2)WRITE(NOUT,3190)                      00003710
      WRITE(NOUT,3200)                                                   00003720
      DO 130 I=1,5                                                       00003730
      A1=RATH(I)*RTD                                                     00003740
      A2=DCTH(I)*RTD                                                     00003750
      A3=ROLL(I)*RTD                                                     00003760
      A4=PBIAS(I)*RTD                                                    00003770
      IF(I.EQ.5) A4=PBIAS(5)                                             00003780
      WRITE (NOUT,3225) A1,A2,A3,A4                                      00003790
  130 CONTINUE                                                           00003800
      A1 = PBIAS(6) * RTD                                                00003810
      A2 = PBIAS(7) * RTD                                                00003820
      A3 = PBIAS(8) * RTD                                                00003830
      IF(       AEARTH ) WRITE(NOUT,3227) A1,A2,A3                       00003840
    9 CONTINUE                                                           00003850
C                                                                        00003860
      IF(IND(48).EQ.3) GO TO 10                                          00003870
      CALL OUTGEN                                                        00003880
   10 IF(IND(48).EQ.7) CALL OUTTEA                                       00003890
      IF(IND(48).NE.7) CALL OUTTIC                                       00003900
      CALL OUTCRD                                                        00003910
      IF(IND(48).EQ.3) RETURN                                            00003920
      IF(IND(48).EQ.7) GO TO 300                                         00003930
      IF(IND(1).EQ.3.OR.IND(1).EQ.4.OR.IND(1).EQ.6.OR.IND(1).EQ.11       00003940
     *.OR.IND(1).EQ.12)GO TO 300                                         00003950
      CALL UPPAGE(21,1,HEAD)                                             00003960
```

```
C                                                                       00003970
C    WRITE INTEGRATION PARAMETERS                                       00003980
C          TOLERANCES                                                   00003990
C          MINIMUM STEP SIZE                                            00004000
C          INTEGRATION STEP MODE                                        00004010
C          INTEGRATION ORDER(N1 + 2)                                    00004020
C                                                                       00004030
      WRITE(NOUT,2023)                                                  00004040
      INTORD = N1 + 2                                                   00004050
      IF(IND(2).EQ.4) INTORD = 8                                        00004060
      IF(IND(2).EQ.5) INTORD = 4                                        00004070
      J1STAT = 6*IND(2)                                                 00004080
      J2STAT = J1STAT - 5                                               00004090
      J1VAR = 5*IND(2)                                                  00004100
      J2VAR = J1VAR - 4                                                 00004110
      WRITE(NOUT,2076) INTORD,(TYPSTP(I),I=J2STAT,J1STAT)               00004120
      IF(NEQ.EQ.0) GO TO 210                                            00004130
      INTORD = N2 + 2                                                   00004140
      IF(IND(2).EQ.4) INTORD = 8                                        00004150
      IF(IND(2).EQ.5) INTORD = 4                                        00004160
      WRITE(NOUT,2077) INTORD,(TYPVEQ(I),I=J2VAR,J1VAR)                 00004170
  210 CONTINUE                                                          00004180
      IF(IND(2).EQ.5) GO TO 225                                         00004190
      WRITE(NOUT,2008)                                                  00004200
      DO 205 I=1,NTSECS                                                 00004210
      IF(INDSEC(5,I).NE.1) GO TO 207                                    00004220
  205 CONTINUE                                                          00004230
      GO TO 209                                                         00004240
  207 WRITE(NOUT,2019) TOL(1),TOL(2),TOL(3),TOL(7),MAXIT(3),STPMIN      00004250
      CALL UPPAGE(8,1,HEAD)                                             00004260
  209 WRITE(NOUT,2021) TOL(4),MAXIT(1)                                  00004270
      DO 215 I=1,NTSECS                                                 00004280
      IF(INDSEC(28,I).EQ.1) GO TO 220                                   00004290
  215 CONTINUE                                                          00004300
      GO TO 225                                                         00004310
  220 WRITE(NOUT,2022) TOL(5),TOL(6),MAXIT(2)                           00004320
      CALL UPPAGE(3,1,HEAD)                                             00004330
  225 CONTINUE                                                          00004340
  300 CONTINUE                                                          00004350
      WRITE(NOUT,2023)                                                  00004360
C                                                                       00004370
C***********************************************************************00004380
C                                                                       00004390
C   WRITE SECTIONING INFORMATION                                        00004400
C                                                                       00004410
      CALL OUTSEC                                                       00004420
C                                                                       00004430
C                                                                       00004440
C        WRITE CROSSOVER CONDITIONS WITH A CALL TO SUBROUTINE OGCROS    00004450
C                                                                       00004460
      IF(LINES.LT.68) GO TO 480                                         00004470
      CALL UPPAGE(79,1,HEAD)                                            00004480
      LINES=0                                                           00004490
  480 IF(IND(48).EQ.1.AND.ILEVEL.LE.2)GO TO 310                         00004500
      CALL OGCROS                                                       00004510
C   END OF SECTIONING DEPENDENT OUTPUT                                  00004520
C                                                                       00004530
```

```
C*************************************************************************00004540
C                                                                         00004550
C                                                                         00004560
C   WRITE MANEUVERS DATA                                                  00004570
C                                                                         00004580
      CALL OUTPHC                                                         00004590
C                                                                         00004600
C                    HARMONIC COEFFICIENTS                                00004610
C                                                                         00004620
  310 IEORD=0                                                             00004630
      IEDEG=0                                                             00004640
      IMORD=0                                                             00004650
      IMDEG=0                                                             00004660
      DO 330 J=1,NTSECS                                                   00004670
      IF (INDSEC(4,J).EQ.1) GO TO 330                                     00004680
      IF (INDSEC(1,J).NE.1) GO TO 320                                     00004690
      IF (INDSEC(11,J).GT.IEORD) IEORD=INDSEC(11,J)                       00004700
      IF (INDSEC(12,J).GT.IEDEG) IEDEG=INDSEC(12,J)                       00004710
      GO TO 330                                                           00004720
  320 IF (INDSEC(1,J).NE.2) GO TO 330                                     00004730
      IF (INDSEC(11,J).GT.IMORD) IMORD=INDSEC(11,J)                       00004740
      IF (INDSEC(12,J).GT.IMDEG) IMDEG=INDSEC(12,J)                       00004750
  330 CONTINUE                                                            00004760
      IF (IEDEG.NE.0) GO TO 680                                           00004770
      IF (IMDEG.NE.0) GO TO 690                                           00004780
      CALL UPPAGE(2,1,HEAD)                                               00004790
      WRITE(NOUT,2058)                                                    00004800
      GO TO 770                                                           00004810
C                                                                         00004820
C   WRITE EARTH HARMONICS                                                 00004830
C                                                                         00004840
  680 IF(IND(48).EQ.1.AND.ILEVEL.LE.2)GO TO 805                           00004850
      CALL HARM(IEORD,IEDEG,1,1,HEAD)                                     00004860
      IF (IMDEG.EQ.0) GO TO 770                                           00004870
C                                                                         00004880
C   WRITE MOON HARMONICS                                                  00004890
C                                                                         00004900
  690 CALL HARM(IMORD,IMDEG,2,1,HEAD)                                     00004910
C                                                                         00004920
C                                                                         00004930
C                                                                         00004940
C                                                                         00004950
C   WRITE COVARIANCE MATRIX                                               00004960
C                                                                         00004970
  770 IF(IND(99).NE.1.AND.IND(48).NE.7) GO TO 800                         00004980
      IF(IND(40).NE.0.OR.IND(41).NE.0) GO TO 780                          00004990
      IF(IND(42).NE.0.OR.IND(43).NE.0) GO TO 780                          00005000
      GO TO 800                                                           00005010
  780 CALL UPPAGE(20,1,HEAD)                                              00005020
      IF ( LOVWT )  WRITE( NOUT, 2200 )                                   00005030
      CALL GVCVL                                                          00005040
      IF ( .NOT. LOVWT )  WRITE( NOUT, 2032 )  ( VCVTTL(I), I = 1,6 )     00005050
      WRITE(NOUT,2033) VCVTTL(1), ( COVMAT(I), I=1, 6 )                   00005060
      WRITE(NOUT,2034) VCVTTL(2), ( COVMAT(I), I=7, 11)                   00005070
      WRITE(NOUT,2035) VCVTTL(3), ( COVMAT(I), I=12, 15)                  00005080
      WRITE(NOUT,2036) VCVTTL(4), ( COVMAT(I), I=16, 18)                  00005090
      WRITE(NOUT,2037) VCVTTL(5), ( COVMAT(I), I=19, 20)                  00005100
```

```
      WRITE(NOUT,2038) VCVTTL(6), COVMAT(21)                                00005110
  800 IF(IND(48).EQ.7) GO TO 999                                            00005120
      CALL UPPAGE(5,1,HEAD)                                                 00005130
      WRITE(NOUT,2023)                                                      00005140
C                                                                           00005150
C   WRITE OUTPUT OPTIONS                                                    00005160
C                                                                           00005170
      IF(IND(48).EQ.1.AND.ILEVEL.LE.2)GO TO 805                             00005180
      CALL OUTOUT                                                           00005190
C                                                                           00005200
C   WRITE END MESSAGE                                                       00005210
C                                                                           00005220
  805 IF(IND(48).EQ.1)GO TO 810                                             00005230
      CALL UPPAGE(3,1,HEAD)                                                 00005240
      WRITE(NOUT,2009)                                                      00005250
      LINES = 100                                                           00005260
C                                                                           00005270
C   INITIALIZE NSCO FOR OUTOG2 USE                                          00005280
C                                                                           00005290
  810 IF (IND(45).NE.2) REWIND NSCO                                         00005300
C                                                                           00005310
  999 RETURN                                                                00005320
C                                                                           00005330
C                                                                           00005340
 2000 FORMAT(T34,9A8)                                                       00005350
 2101 FORMAT(54X,2A8,A7)                                                    00005360
 2001 FORMAT('-',5X,'CURRENT TIME ',A4,I2,', ',I4,', ',I2,' HRS,',I2,      00005370
     1  ' MIN, ', F7.3,' SEC')                                             00005380
 2002 FORMAT(T7,'LEVEL OF OUTPUT IS',T26,I2)                                00005390
 2003 FORMAT('+',T92,A5,A8,A7)                                              00005400
 2004 FORMAT(T7,'SATELLITE NAME',T30,A8)                                    00005410
 2008 FORMAT('0',27('* '),'INTEGRATION PARAMETERS',27('* '))                00005420
 2009 FORMAT(24(' *'),'END OF GTDS INITIAL CONDITIONS REPORT',23(' *'))     00005430
 2019 FORMAT(//,T20,'RELATIVE UPPER TRUNCATION ERROR BOUND FOR STEPSIZE     00005440
     1CONTROL',T83,' = ',D11.4,/,                                           00005450
     2  T20,'RELATIVE LOWER TRUNCATION ERROR BOUND FOR STEPSIZE CONTROL'    00005460
     3,T83,' = ',D11.4,/,                                                   00005470
     4  T20,'RELATIVE NOMINAL TRUNCATION ERROR BOUND FOR STEPSIZE CONTRO    00005480
     6L',T83,' = ',D11.4,/,                                                 00005490
     7 T20,'INCREASE FACTOR FOR STEPSIZE COMPUTATION',T83,' = ',D11.4,/,    00005500
     1 T20,'MAXIMUM NUMBER OF STEPSIZE-CHANGE RESTARTS FOR STARTER',        00005510
     9 T83,' = ',I5,/,                                                      00005520
     * T20,'MINIMUM STEPSIZE FOR VARY STEP',T83,' = ',F8.1,' SECONDS')     00005530
 2021 FORMAT(T20,'CONVERGENCE TOLERANCE FOR PREDICTOR-CORRECTOR ITERATIO    00005540
     1NS',T83,' = ',D11.4,/,   T20,'MAXIMUM NUMBER OF CORRECTOR ITERATIO    00005550
     2NS',T83,' = ',I5)                                                     00005560
 2022 FORMAT(T20,'CORRECTOR TOLERANCE FOR EQUATIONS OF MOTIONS MULTISTEP    00005570
     1 STARTER',T83,' = ',D11.4,/,                                          00005580
     2  T20,'CORRECTOR TOLERANCE FOR VARIATIONAL EQUATIONS STARTER',        00005590
     4  T83,' = ',D11.4,/,                                                 00005600
     5  T20,'MAXIMUM NUMBER OF CORRECTOR ITERATIONS IN STARTER',            00005610
     6   T83,' = ',I5)                                                     00005620
 2023 FORMAT('0',/,'0',64('* '))                                            00005630
 2032 FORMAT('-',T53,'INITIAL STATE COVARIANCE MATRIX',/                    00005640
     *      '0',19X,5(A8,11X), A8)                                         00005650
 2033 FORMAT('0',5X,A8,4X,5(D16.8,3X),D16.8)                                00005660
 2034 FORMAT('0',5X,A8,4X,T38,4(D16.8,3X),D16.8)                            00005670
```

```
 2035 FORMAT('0',5X,A8,4X,T57,3(D16.8,3X),D16.8)                                00005680
 2036 FORMAT('0',5X,A8,4X,T76,2(D16.8,3X),D16.8)                                00005690
 2037 FORMAT('0',5X,A8,4X,T95,D16.8,3X,D16.8)                                   00005700
 2038 FORMAT('0',5X,A8,4X,T114,D16.8)                                           00005710
 2041 FORMAT(T7,'SATELLITE NUMBER',T30,I8)                                      00005720
 2042 FORMAT(T7,'SATELLITE CONFIGURATION:'/T12,'SATELLITE IS ',3A8)             00005730
 2043 FORMAT(T12,'DIAMETER                =',D16.8,'  METERS')                  00005740
 2044 FORMAT(T12,'DIAMETER                =',D16.8,'  METERS'/                  00005750
     *       T12,'LENGTH                  =',D16.8,'  METERS')                  00005760
 2045 FORMAT(T12,'PADDLE AREA             =',D16.8,'  M**2  ','ANGLE BET00005770
     *WEEN PADDLES AND AXIS = ',F10.5,' DEGREES')                                00005780
 2046 FORMAT(T12,'MASS                    =',D16.8,'  KG'/                      00005790
     *       T12,'AERO VARIATION (RHO1)   =',D16.8/                            00005800
     *       T12,'REFLECTIVITY CONSTANT   =',D16.8,'    AREA FOR RADIATI00005810
     *ON PRESSURE =',D16.8,'  M**2')                                             00005820
 2047 FORMAT ('+',T68,'COEFFICIENT OF DRAG USED FOR THE RUN =',D16.8)           00005830
 2058 FORMAT(/,49X,'HARMONIC COEFFICIENTS NOT REQUIRED')                        00005840
 2076 FORMAT(/,21(' *'),' NUMERICAL INTEGRATION OPTIONS AND PARAMETERS '00005850
     1,21(' *'),//,3X,'NUMERICAL INTEGRATOR TYPE FOR STATE PROPAGATION-'00005860
     2,/,10X,I2,' TH ORDER ', 5A8,A3)                                           00005870
 2077 FORMAT(3X,'NUMERICAL INTEGRATOR TYPE VARIATIONAL EQUATIONS-',             00005880
     1 /,10X,I2,' TH ORDER ',4A8,A2)                                            00005890
 2200 FORMAT( '0', T53, 'OVERRIDE WEIGHTING MATRIX' / )                         00005900
 3000 FORMAT (T7,'SPACECRAFT PARAMETERS USED WITH LANDMARK AND PICTURE E00005910
     *ARTH-EDGE DATA - UNITS ARE DEGREES,SECONDS OR A COMBINATION OF THE00005920
     1SE' )                                                                     00005930
 3100 FORMAT(T12,'SUN SENSOR TO CAMERA ANGLE =',D16.8,' DEGREE',T68,            00005940
     *  'SATELLITE SPIN RATE =',D16.8,' DEG/SEC')                               00005950
 3105 FORMAT(T12,' SENSOR MOUNTING ANGLE =',D16.8,' DEGREE , DETERMINE T00005960
     *RUE EARTH HORIZON CROSSING EVERY ',F5.0,'MINUTES' )                        00005970
 3125 FORMAT(T9,' A POLYNOMIAL-PLUS-TRIGONOMETRIC MODEL WILL BE ASSUMED 00005980
     * FOR EACH ATTITUDE ANGLE')                                                00005990
 3175 FORMAT(T9,'A POLYNOMIAL MODEL IS ASSUMED FOR EACH ATTITUDE ANGLE')00006000
 3185 FORMAT(T9,' A GEODETIC REFERENCE FRAME IS USED TO DEFINE YAW,PITCH00006010
     * AND ROLL ATTITUDE')                                                      00006020
 3190 FORMAT(T9,' A GEOCENTRIC REFERENCE FRAME IS USED TO DEFINE YAW,PIT00006030
     *CH AND ROLL ATTITUDE')                                                   00006040
 3200 FORMAT(T6,'RT.ASC.OR YAW',T40,'DECL.OR PITCH',T74,'ROLL(3-AXIS)',         00006050
     *       T108,'PICTURE BIAS')                                              00006060
 3225 FORMAT(T6,D16.8,T40,D16.8,T74,D16.8,T108,D16.8)                           00006070
 3227 FORMAT (T108,D16.8)                                                       00006080
      END                                                                       00006090
```

******** END OF MEMBER OUTCG1 612 RECORDS ********

```
      SUBROUTINE PCHEBY(*)                                                  00000010
C                                                                           00000020
C     VERSION OF 06/30/79     **** CHECKED OUT ****                         00000030
C                                                                           00000040
C     GENEOLOGY:  PROGRAM ORIGINALLY CALLED 'COPS'.                         00000050
C         THIS PROGRAM WAS INITIALLY WRITTEN BY DR. TERRY FEAGIN.           00000060
C         WHEN GTDS WAS TRANSFERRED TO THE CSDL TO BECOME OUR PRIMARY       00000070
C         ORBIT TEST BED, I COULD NOT FULLY UNDERSTAND THE PROGRAM AS       00000080
C         CODED IN GTDS.  I THEREFORE REDEVELOPED THE EQUATIONS AND         00000090
C         IMPLEMENTED THIS VERSION OF SUBROUTINE PCHEBY.  THE CAPABILI-     00000100
C         TIES OF THIS PROGRAM HAVE BEEN EXPANDED TO ALLOW INTEGRATION      00000110
C         OF THE VARIATION OF PARAMETER EQUATIONS OF MOTION (BOTH           00000120
C         AVERAGED AND HIGH PRECISION), AS WELL AS THE ORIGINAL COWELL      00000130
C         EQUATIONS OF MOTION.                                              00000140
C                                                                           00000150
C         THE INTENT OF THIS DEVELOPMENT WAS TO FORMULATE A METHOD OF       00000160
C         NUMERICAL INTEGRATION THAT WOULD BE AMENABLE TO PARALLEL          00000170
C         PROCESSING ON A SINGLE INSTRUCTION STREAM/MULTIPLE DATA           00000180
C         STREAM HARDWARE ARCHITECTURE.                                     00000190
C                                                                           00000200
C                                                                           00000210
C     PURPOSE                                                               00000220
C         TO INTEGRATE VIA THE METHOD OF PICARD ITERATION A VECTOR          00000230
C         OF FIRST ORDER DIFFERENTIAL EQUATIONS OF THE FORM:                00000240
C                                                                           00000250
C                 DY/DX = F(X,Y(X));  Y(X ) = Y                             00000260
C                                       0     0                             00000270
C                                                                           00000280
C                                  OR                                       00000290
C                                                                           00000300
C               DYDOT/DX = F(X,Y(X),YDOT(X));  YDOT(X ) = YDOT              00000310
C                                                    0        0             00000320
C                                                                           00000330
C                  DY/DX = YDOT(X);  Y(X ) = Y                              00000340
C                                    0     0                                00000350
C                                                                           00000360
C         WHERE THE RIGHT HAND SIDES OF THESE EQUATIONS ARE APPROXIMATED    00000370
C         BY APPROPRIATE SERIES OF CHEBYSHEV POLYNOMIALS.                   00000380
C                                                                           00000390
C     METHOD                                                                00000400
C         ONCE TRANSFORMED TO INTEGRAL FORM, THESE EQUATIONS ARE SOLVED     00000410
C         BY SUCCESSIVE APPROXIMATION FOR Y(X) IN THE FOLLOWING MANNER:     00000420
C                                                                           00000430
C                                      X                                    00000440
C             Y   (X) = Y(X ) + INT/  F(S,Y (S))DS                          00000450
C              K+1          0       X       K                             00000460
C                                    0                                      00000470
C                     OR                                                    00000480
C                                                                           00000490
C                                      X                                    00000500
C           YDOT   (X) = YDOT(X ) + INT/  F(S,Y (S),YDOT (S))DS             00000510
C               K+1             0       X       K        K                 00000520
C                                       0                                   00000530
C                                                                           00000540
C                                      X                                    00000550
C             Y   (X) = Y(X ) + INT/  YDOT   (S)DS                          00000560
C              K+1          0       X      K+1                             00000570
```

```
C                                   0                                          00000580
C                                                                              00000590
C                         WHERE S IS A DUMMY VARIABLE OF INTEGRATION           00000600
C                         AND THE SUBSCRIPT ON Y AND YDOT REPRESENTS           00000610
C                         THE ITERATION NUMBER.                                00000620
C                                                                              00000630
C          THE INTEGRANDS ABOVE ARE APPROXIMATED BY SERIES OF                  00000640
C          CHEBYSHEV POLYNOMIALS SUCH THAT:                                    00000650
C                                                                              00000660
C                                                 M                            00000670
C                         F(X,Y(X)) = A /2 + SUM/  A T (X)                     00000680
C                                     0       J=1  J J                         00000690
C                                                                              00000700
C                         WHERE THE A'S ARE COEFFICIENTS AND THE T'S           00000710
C                         ARE CHEBYSHEV POLYNOMIALS.                           00000720
C                                                                              00000730
C          THE COEFFICIENTS ARE FOUND BY APPLYING THE PROPERTY OF              00000740
C          ORTHOGONALITY OVER SUMMATION POSSESSED BY CHEBYSHEV                 00000750
C          POLYNOMIALS AT THE N+1 POINTS X(0) TO X(N) GIVEN BY:                00000760
C                                                                              00000770
C                         X  = COS (I*PI/N);    I = 0,1,2,...,N              00000780
C                          I                                                   00000790
C                                                                              00000800
C          THE COEFFICIENTS ARE GIVEN BY THE FOLLOWING:                        00000810
C                              N                                               00000820
C               A  = 2/N * SUM"/  F(X ,Y(X )) * T (X );  L = 0,1,...,M         00000830
C                L            I=0    I    I      L  I                         00000840
C                                                                              00000850
C                         WHERE THE DOUBLE PRIME (") NOTATION ON THE           00000860
C                         SUMMATION INDICATES THAT THE FIRST (I=0) AND         00000870
C                         THE LAST (I=N) TERMS ARE TO BE FACTORED BY           00000880
C                         ONE HALF.                                            00000890
C                                                                              00000900
C     CALLING SEQUENCE                                                         00000910
C         CALL PCHEBY(&NNN)                                                    00000920
C                                                                              00000930
C     CALLING SEQUENCE PARAMETERS                                              00000940
C         &NNN            STATEMENT NUMBER IN THE CALLING PROGRAM TO           00000950
C                         WHICH CONTROL IS TO BE TRANSFERRED IF AN ERROR       00000960
C                         IS DETECTED IN 'PCHEBY'.                             00000970
C                                                                              00000980
C     COMMON/CHBINT/ T(70), X(70), TX(70,70)                                   00000990
C         T(70)           ARRAY (70X1) CONTAINING THE ORTHOGONALITY            00001000
C                         POINTS IN THE TRUE INDEPENDENT VARIABLE T.           00001010
C         X(70)           ARRAY (70X1) CONTAINING THE ORTHOGONALITY            00001020
C                         POINTS IN  NORMALIZED INDEPENDENT VARIABLE X.        00001030
C         TX(70,70)       ARRAY (70X70) CONTAINING THE M+1 CHEBYSHEV           00001040
C                         POLYNOMIALS EVALUATED AT THE N+1 POINTS              00001050
C                         OF ORTHOGONALITY.                                    00001060
C                                                                              00001070
C     COMMON/COVAL/ A(70,6), BC(70,6), FORC(70,6)                              00001080
C         A(70,6)         ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS        00001090
C                         THE M+1 COEFFICIENTS OF THE CHEBYSHEV SERIES         00001100
C                         APPROXIMATING THE INTEGRAND OF THE KTH               00001110
C                         EQUATION TO BE INTEGRATED (AS SPECIFIED BY THE       00001120
C                         VALUE OF THE FLAG IEQSET).                           00001130
C         BC(70,6)        ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS        00001140
```

```
C                       THE COEFFICIENTS OF THE CHEBYSHEV SERIES       00001150
C                       APPROXIMATING THE INTEGRAL OF THE KTH INTEGRAND00001160
C                       WHOSE COEFFICIENTS WERE STORED IN THE KTH      00001170
C                       COLUMN OF MATRIX A.                            00001180
C         FORC(70,6)    ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS  00001190
C                       THE INTEGRAND OF THE KTH EQUATION TO BE        00001200
C                       INTEGRATED, EVALUATED AT THE N+1 POINTS OF     00001210
C                       ORTHOGONALITY.                                 00001220
C                                                                      00001230
C     COMMON/NPCNTL/ MORD, NSUB, IEQSET, ITERM                         00001240
C         MORD          MAXIMUM ORDER OF CHEBYSHEV POLYNOMIAL IN THE   00001250
C                       SERIES REPRESENTATION OF THE INTEGRAND (M<68). 00001260
C         NSUB          SUBSCRIPT OF FINAL ORTHOGONALITY POINT (N<70). 00001270
C         IEQSET        FLAG INDICATING THE SET OF FIRST ORDER         00001280
C                       DIFFERENTIAL EQUATIONS TO BE INTEGRATED        00001290
C                       BY THE PICARD-CHEBYSHEV INTEGRATOR.            00001300
C                       IEQSET = 1:  COWELL EQUATIONS                  00001310
C                       IEQSET = 2:  KEPLERIAN VOP EQUATIONS           00001320
C                                    (MEAN ANOMALY)                    00001330
C                       IEQSET = 3:  EQUINOCTIAL VOP EQUATIONS         00001340
C                                    (MEAN LONGITUDE)                  00001350
C                       IEQSET = 4:  EQUINOCTIAL VOP EQUATIONS         00001360
C                                    (ECCENTRIC LONGITUDE)             00001370
C                       IEQSET = 5:  AVERAGED EQUINOCTIAL VOP EQUATIONS00001380
C                                    (MEAN MEAN LONGITUDE)             00001390
C                       IEQSET = 6:  AVERAGED KEPLERIAN VOP EQUATIONS  00001400
C                                    (MEAN MEAN ANOMALY)               00001410
C         ITERM         TERMINAL ITERATION, SPECIFYING THE MAXIMUM     00001420
C                       NUMBER OF ITERATIONS ALLOWED BEFORE THE        00001430
C                       INTEGRATION PROCESS WILL BE TERMINATED.        00001440
C                                                                      00001450
C     COMMON/STATE/ S0(6), STK(70,6), STK1(70,6)                       00001460
C         S0(6)         ARRAY (6X1) CONTAINING THE INITIAL CONDITIONS  00001470
C                       FOR EACH OF THE COMPONENTS OF THE STATE VECTOR 00001480
C                       TO BE INTEGRATED.                              00001490
C         STK(70,6)     ARRAY (70X6) IN WHICH THE ITH COLUMN CONTAINS  00001500
C                       THE ITH STATE VARIABLE, EVALUATED AT THE N+1   00001510
C                       POINTS OF ORTHOGONALITY, FOR THE KTH ITERATION 00001520
C                       OF THE PICARD METHOD.                          00001530
C         STK1(70,6)    ARRAY (70X6) IN WHICH THE ITH COLUMN CONTAINS  00001540
C                       THE ITH STATE VARIABLE, EVALUATED AT THE N+1   00001550
C                       POINTS OF ORTHOGONALITY, FOR ITERATION K+1     00001560
C                       OF THE PICARD METHOD.                          00001570
C                                                                      00001580
C     COMMON/CNVRG/ CON(6), CONKEP                                     00001590
C         CON(6)        ARRAY (6X1) CONTAINING THE CONVERGENCE         00001600
C                       CRITERION FOR EACH OF THE STATE EQUATIONS TO BE00001610
C                       INTEGRATED.                                    00001620
C         CONKEP        CONVERGENCE CRITERION TO BE USED BY THE        00001630
C                       SUBROUTINE KEPLR IN SOLVING FOR ECCENTRIC      00001640
C                       (OR TRUE) ANOMALY BY NEWTON'S METHOD.          00001650
C                                                                      00001660
C     EXTERNAL REFERENCES                                              00001670
C         VORTPT,VMATTX,VCOVAL,VCOINT,F2BDY,FORCES,EPELMS,CSER1        00001680
C         KEPLR,LGRNGE                                                 00001690
C                                                                      00001700
C     ANALYSIS                                                         00001710
```

```
PCHEBY              JSS1493.GTDS.UPDATE.FORT

C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00001720
C                                                                            00001730
C     PROGRAMMER                                                             00001740
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00001750
C                                                                            00001760
      IMPLICIT REAL*8(A-H,O-Z)                                               00001770
C                                                                            00001780
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * *00001790
C                                                                            00001800
      COMMON/CHBINT/ T(70)         , X(70)         , TX(70,70)             00001810
C                                                                            00001820
      COMMON/COVAL / A(70,6)       , BC(70,6)      , FORC(70,6)            00001830
C                                                                            00001840
      COMMON/CNVRG / CON(6)        , CONKEP                                  00001850
C                                                                            00001860
      COMMON/  FRC / FCOM(1300)    , IFCOM(50)                               00001870
C                                                                            00001880
      COMMON/NPCNTL/ MORD          , NSUB          , IEQSET       , ITERM 00001890
C                                                                            00001900
      COMMON/STATE / S0(6)         , STK(70,6)     , STK1(70,6)            00001910
C                                                                            00001920
      COMMON/TIMREG/ DTRG(94)                                                00001930
C                                                                            00001940
      COMMON/WORKER/ DPWOR(3264)   , INWOR(13)                               00001950
C                                                                            00001960
C * * * * * * * * *       DIMENSION STATEMENTS       * * * * * * * * * *00001970
C                                                                            00001980
      DIMENSION      STA(6),RHS(6),EL(26),DPARRY(15,10)                      00001990
C                                                                            00002000
      DIMENSION      GRAV(11),VPELMT(10),PARARY(15,10)                       00002010
C                                                                            00002020
C * * * * * * * * *      EQUIVALENCE STATEMENTS      * * * * * * * * * *00002030
C                                                                            00002040
C     FROM COMMON/  FRC /:                                                   00002050
      EQUIVALENCE   ( FCOM(2)    , GRAV(1) ) , ( IFCOM(1)   , ICENT   ) 00002060
C                                                                            00002070
C     FROM COMMON/NPCNTL/:                                                   00002080
      EQUIVALENCE   ( MORD       ,     M   ) , ( NSUB       ,   N     ) 00002090
C                                                                            00002100
C     FROM COMMON/TIMREG/:                                                   00002110
      EQUIVALENCE   ( DTRG(92)   ,  STREG  )                                 00002120
C                                                                            00002130
C     FROM COMMON/WORKER/:                                                   00002140
      EQUIVALENCE   ( DPWOR(1)   , STA(1)  ) , ( DPWOR(7)   ,   XACC  ),00002150
     1              ( DPWOR(47)  ,   YACC  ) , ( DPWOR(87)  ,   ZACC  ),00002160
     2              ( DPWOR(3215),   TFE   ) , ( DPWOR(3216),    H    ),00002170
     3              ( INWOR(2)   ,   KACC  ) , ( DPWOR(2829),VPELMT(1)),00002180
     4              ( INWOR(3)   ,  NOPARM ) ,                          00002190
     5              ( DPWOR(7)   ,  DPARRY(1,1)  ) ,                    00002200
     6              ( DPWOR(157) ,  PARARY(1,1)  )                      00002210
C                                                                            00002220
C * * * * * * * * *        LOGICAL VARIABLES         * * * * * * * * * *00002230
C                                                                            00002240
      LOGICAL        CONK,CONK1,CNV(6)                                       00002250
C                                                                            00002260
C     ERROR CHECK INPUT PARAMETERS                                           00002270
C                                                                            00002280
```

```
      IF (IEQSET.LT.1.OR.IEQSET.GT.6)   GO TO 900                       00002290
      IF (ITERM.LT.2)                   GO TO 905                       00002300
      IF (M.GE.68)                      GO TO 910                       00002310
      IF (N.GE.70)                      GO TO 915                       00002320
C                                                                       00002330
C     INITIALIZE VARIABLES TL, TU, M1, M2, M3, N1, GM, KACC,            00002340
C     ICOUNT, ISET, CONK AND CONK1.                                     00002350
C                                                                       00002360
      TU = TFE                                                          00002370
      TL = TU - H                                                       00002380
      M1 = M + 1                                                        00002390
      M2 = M + 2                                                        00002400
      M3 = M + 3                                                        00002410
      N1 = N + 1                                                        00002420
      GM = GRAV(ICENT)                                                  00002430
      KACC = 1                                                          00002440
      ICOUNT = 1                                                        00002450
      ISET   = 1                                                        00002460
      CONK   = .FALSE.                                                  00002470
      CONK1  = .FALSE.                                                  00002480
C                                                                       00002490
C     COMPUTE THE MATRIX OF CHEBYSHEV POLYNOMIALS T(0) THROUGH T(M) AT  00002500
C     THE N+1 POINTS OF ORTHOGONALITY X(0) THROUGH X(N).                00002510
C                                                                       00002520
      CALL VORTPT(N,TL,TU)                                              00002530
      CALL VMATTX(M,N)                                                  00002540
C                                                                       00002550
C     BEGIN COMPUTATION FOR COWELL EQUATIONS:  DETERMINE EPOCH ELEMENTS 00002560
C                                                                       00002570
      IF (IEQSET.GT.1)                  GO TO 200                       00002580
      PRINT    665                                                      00002590
      DO   5    KCOMP = 1,6                                             00002600
      S0(KCOMP) = STA(KCOMP)                                            00002610
    5 CONTINUE                                                          00002620
      CALL EPELMS (S0,TL,GM,ISET,EL)                                    00002630
C                                                                       00002640
C     EVALUATE RIGHT-HAND MEMBERS FOR THE ZEROTH ITERATION              00002650
C                                                                       00002660
      DO  20   I = 1,N1                                                 00002670
      AM = EL(7) * (T(I) - EL(6))                                       00002680
      CALL KEPLR(AM,EL(2),CONKEP,AE,AF)                                 00002690
      CALL LGRNGE(S0,EL(10),GM,AF,EL(8),EL(13),STA)                     00002700
      CALL F2BDY(STA,GM,RHS)                                            00002710
      DO  10   KCOMP = 1,6                                              00002720
      STK(I,KCOMP)  = STA(KCOMP)                                        00002730
   10 FORC(I,KCOMP) = RHS(KCOMP)                                        00002740
   20 CONTINUE                                                          00002750
C                                                                       00002760
C     COMPUTE COEFFICIENTS FOR THE CHEBYSHEV SERIES FOR KTH             00002770
C     ITERATION OF THE RIGHT-HAND MEMBERS.                              00002780
C                                                                       00002790
   30 CALL VCOVAL(M,N,IEQSET)                                           00002800
C                                                                       00002810
C     COMPUTE COEFFICIENTS FOR THE INTEGRATED SERIES                    00002820
C                                                                       00002830
      ICOUNT = ICOUNT + 1                                               00002840
      DO  40   KCOMP = 1,6                                              00002850
```

```
   40 CNV(KCOMP) = .TRUE.                                                   00002860
      CALL VCOINT(M,IEQSET,TL,TU)                                           00002870
      IF (IEQSET.GT.1)                  GO TO 270                           00002880
C                                                                           00002890
C     EVALUATE THE POSITION AND VELOCITY SERIES AT THE CHEBYSHEV POINTS     00002900
C     FOR ITERATION K+1; COMPUTE THE RELATIVE ERROR AT EACH POINT.          00002910
C                                                                           00002920
      DO  70   KCOMP = 1,3                                                  00002930
      DO  60   I = 1,N1                                                     00002940
      CALL CSER1(M2,X(I),BC(1,KCOMP+3),STK1(I,KCOMP))                       00002950
      RELERR = DABS((STK(I,KCOMP) - STK1(I,KCOMP))/STK(I,KCOMP))           00002960
      IF (RELERR.GT.CON(KCOMP)/2.D0) CNV(KCOMP) = .FALSE.                   00002970
      CALL CSER1(M2,X(I),BC(1,KCOMP),STK1(I,KCOMP+3))                       00002980
      RELERR = DABS((STK(I,KCOMP+3) - STK1(I,KCOMP+3))/STK(I,KCOMP+3))     00002990
   60 IF (RELERR.GT.CON(KCOMP+3)/2.D0) CNV(KCOMP+3) = .FALSE.               00003000
   70 CONTINUE                                                              00003010
C                                                                           00003020
C     USE NEW VALUES OF POSITION AND VELOCITY TO DETERMINE NEW              00003030
C     VALUES OF ACCELERATION FOR NEXT ITERATION.                            00003040
C                                                                           00003050
      DO 100   I = 1,N1                                                    00003060
      TFE = T(I)                                                            00003070
      DO  80   KCOMP = 1,6                                                  00003080
   80 STA(KCOMP) = STK1(I,KCOMP)                                            00003090
      CALL FORCES                                                           00003100
      FORC(I,1) = XACC                                                      00003110
      FORC(I,2) = YACC                                                      00003120
      FORC(I,3) = ZACC                                                      00003130
      DO  90   KCOMP = 1,3                                                  00003140
   90 FORC(I,KCOMP+3) = 0.D0                                                00003150
  100 CONTINUE                                                              00003160
C                                                                           00003170
C     CHECK FOR CONVERGENCE ON THIS ITERATION                               00003180
C                                                                           00003190
  110 CONK1 = CNV(1).AND.CNV(2).AND.CNV(3).AND.                             00003200
     1        CNV(4).AND.CNV(5).AND.CNV(6)                                 00003210
C                                                                           00003220
C     IF CONVERGED, OUTPUT PICARD CONVERGENCE REPORT.                       00003230
C                                                                           00003240
      IF (CONK.AND.CONK1.AND.IEQSET.EQ.1)         GO TO 150                 00003250
      IF (CONK.AND.CONK1.AND.IEQSET.GT.1)         GO TO 330                 00003260
C                                                                           00003270
C     OTHERWISE, ITERATION K+1 RESULTS BECOME ITERATION K RESULTS.          00003280
C                                                                           00003290
      CONK = CONK1                                                          00003300
      DO 140  KCOMP = 1,6                                                   00003310
      DO 130  I = 1,N1                                                      00003320
  130 STK(I,KCOMP) = STK1(I,KCOMP)                                          00003330
  140 CONTINUE                                                              00003340
C                                                                           00003350
C     IF THE TERMINAL ITERATION NUMBER HAS NOT BEEN EXCEEDED, RETURN TO     00003360
C     30 AND ITERATE AGAIN; OTHERWISE OUTPUT NON-CONVERGENCE REPORT.        00003370
C                                                                           00003380
      IF (ICOUNT.LT.ITERM)              GO TO 30                            00003390
      GO TO 500                                                             00003400
C                                                                           00003410
C     PREPARE AND OUTPUT CONELL CONVERGENCE REPORT                          00003420
```

```
C                                                                       00003430
  150 TFE = TU                                                          00003440
      DO  155    KCOMP = 1,6                                            00003450
      STA(KCOMP) = STK1(N1,KCOMP)                                       00003460
  155 CONTINUE                                                          00003470
      PRINT    600                                                      00003480
      PRINT    650                                                      00003490
      PRINT    605,M                                                    00003500
      PRINT    610,N1                                                   00003510
      PRINT    670,H                                                    00003520
      PRINT    640,CON(4),CON(5),CON(6),CON(1),CON(2),CON(3)            00003530
      PRINT    645,ICOUNT,ITERM                                         00003540
      DO 170  KCOMP = 1,3                                               00003550
      PRINT    625,KCOMP                                                00003560
      DO 160   K = 1,M3                                                 00003570
      K1 = K - 1                                                        00003580
  160 PRINT    630,K1,KCOMP,A(K,KCOMP),K1,KCOMP,BC(K,KCOMP),            00003590
     1             K1,KCOMP,BC(K,KCOMP+3)                             00003600
  170 CONTINUE                                                          00003610
      PRINT    615                                                      00003620
      RETURN                                                            00003630
C                                                                       00003640
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00003650
C                                                                       00003660
C     BEGIN COMPUTATION FOR THE VOP EQUATIONS:  DETERMINE EPOCH ELEMENTS00003670
C                                                                       00003680
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00003690
C                                                                       00003700
  200 IF (IEQSET.GT.2.AND.IEQSET.LT.6)  ISET = 2                        00003710
      PRINT    665                                                      00003720
      DO  210  KCOMP = 1,6                                              00003730
      S0(KCOMP) = PARARY(KACC,KCOMP)                                    00003740
  210 CONTINUE                                                          00003750
      CALL EPELMS(STA,TL,GM,ISET,EL)                                    00003760
C                                                                       00003770
C     EVALUATE RIGHT-HAND MEMBERS FOR THE ZEROTH ITERATION.             00003780
C                                                                       00003790
      DO  260  I = 1,N1                                                 00003800
      AM = EL(23) + EL(7) * (T(I) - TL)                                 00003810
      IF (ISET.EQ.2)                    GO TO 220                       00003820
      FSTVAR = AM                                                       00003830
      GO TO 240                                                         00003840
  220 IF (IEQSET.EQ.4)                  GO TO 230                       00003850
      FSTVAR = AM + EL(11) + EL(13) * EL(12)                            00003860
      GO TO 240                                                         00003870
  230 CALL KEPLR(AM,EL(9),CONKEP,AE,AF)                                 00003880
      FSTVAR = AE + EL(11) + EL(13) * EL(12)                            00003890
      XRPOS  = EL(1)*(1.D0-EL(3)*DCOS(FSTVAR)-EL(2)*DSIN(FSTVAR))       00003900
  240 DO  250  KCOMP = 1,5                                              00003910
      STK(I,KCOMP)  = S0(KCOMP)                                         00003920
      FORC(I,KCOMP) = 0.D0                                              00003930
  250 CONTINUE                                                          00003940
      STK(I,6)  = FSTVAR                                                00003950
      IF (IEQSET.EQ.4)        FORC(I,6) = (EL(7) * EL(1))/XRPOS         00003960
      IF (IEQSET.NE.4)        FORC(I,6) =  EL(7)                        00003970
  260 CONTINUE                                                          00003980
C                                                                       00003990
```

```
C     COMPUTE COEFFICIENTS FOR THE CHEBYSHEV SERIES FOR THE KTH          00004000
C     ITERATION OF THE ELEMENT RATES; INTEGRATE SERIES.                  00004010
C                                                                        00004020
      GO TO 30                                                           00004030
C                                                                        00004040
C     EVALUATE THE VOP ELEMENT SERIES AT THE CHEBYSHEV POINTS FOR        00004050
C     ITERATION K+1; COMPUTE RELATIVE ERROR.                             00004060
C                                                                        00004070
  270 DO  290  KCOMP = 1,6                                               00004080
      DO  280  I = 1,N1                                                  00004090
      CALL CSER1(M1,X(I),BC(1,KCOMP),STK1(I,KCOMP))                      00004100
      RELERR = DABS((STK(I,KCOMP) - STK1(I,KCOMP))/STK(I,KCOMP))        00004110
      IF (RELERR.GT.CON(KCOMP)/2.D0)    CNV(KCOMP) = .FALSE.             00004120
  280 CONTINUE                                                           00004130
  290 CONTINUE                                                           00004140
C                                                                        00004150
C     USE NEW VALUES OF THE ELEMENTS TO EVALUATE THE ELEMENT RATES FOR   00004160
C     THE NEXT ITERATION.                                                00004170
C                                                                        00004180
      DO  320  I = 1,N1                                                  00004190
      STREG = T(I)                                                       00004200
      TFE   = T(I)                                                       00004210
      DO  300  KCOMP = 1,6                                               00004220
  300 VPELMT(KCOMP) = STK1(I,KCOMP)                                      00004230
      CALL EQUMTN(.FALSE.)                                               00004240
      DO  310  KCOMP = 1,6                                               00004250
  310 FORC(I,KCOMP) = DPARRY(KACC,KCOMP)                                 00004260
  320 CONTINUE                                                           00004270
C                                                                        00004280
C     CHECK FOR CONVERGENCE ON THIS ITERATION.                           00004290
C                                                                        00004300
      GO TO 110                                                          00004310
C                                                                        00004320
C     PREPARE AND OUTPUT VOP CONVERGENCE REPORT.                         00004330
C                                                                        00004340
  330 STREG = TU                                                         00004350
      TFE   = TU                                                         00004360
      DO  340  KCOMP = 1,6                                               00004370
      PARARY(KACC,KCOMP) = STK1(N1,KCOMP)                                00004380
  340 CONTINUE                                                           00004390
      PRINT    600                                                       00004400
      IF (IEQSET.EQ.2)                  PRINT 655                        00004410
      IF (IEQSET.EQ.3)                  PRINT 660                        00004420
      IF (IEQSET.EQ.4)                  PRINT 675                        00004430
      IF (IEQSET.EQ.5)                  PRINT 680                        00004440
      IF (IEQSET.EQ.6)                  PRINT 685                        00004450
      PRINT    605,M                                                     00004460
      PRINT    610,N1                                                    00004470
      PRINT    670,H                                                     00004480
      PRINT    640,CON(1),CON(2),CON(3),CON(4),CON(5),CON(6)             00004490
      PRINT    645,ICOUNT,ITERM                                          00004500
      DO  360  KCOMP = 1,6                                               00004510
      PRINT    620,KCOMP                                                 00004520
      DO  350  K = 1,M2                                                  00004530
      K1 = K - 1                                                         00004540
  350 PRINT    690,K1,KCOMP,A(K,KCOMP),K1,KCOMP,BC(K,KCOMP)              00004550
  360 CONTINUE                                                           00004560
```

```
      PRINT    615                                                          00004570
      RETURN                                                                00004580
C                                                                           00004590
C     PREPARE AND OUTPUT NON-CONVERGENCE REPORT                             00004600
C                                                                           00004610
  500 PRINT    635,ICOUNT                                                   00004620
      IF (IEQSET.EQ.1)                  PRINT 650                          00004630
      IF (IEQSET.EQ.2)                  PRINT 655                          00004640
      IF (IEQSET.EQ.3)                  PRINT 660                          00004650
      IF (IEQSET.EQ.4)                  PRINT 675                          00004660
      IF (IEQSET.EQ.5)                  PRINT 680                          00004670
      IF (IEQSET.EQ.6)                  PRINT 685                          00004680
      PRINT    605,M                                                        00004690
      PRINT    610,N1                                                       00004700
      PRINT    670,H                                                        00004710
      PRINT    640,CON(1),CON(2),CON(3),CON(4),CON(5),CON(6)                00004720
      RETURN 1                                                              00004730
C                                                                           00004740
C     ERROR RETURN EXITS                                                    00004750
C                                                                           00004760
  900 PRINT    800                                                          00004770
      RETURN 1                                                              00004780
  905 PRINT    805                                                          00004790
      RETURN 1                                                              00004800
  910 PRINT    810                                                          00004810
      RETURN 1                                                              00004820
  915 PRINT    815                                                          00004830
      RETURN 1                                                              00004840
C                                                                           00004850
C                          FORMAT STATEMENTS                                00004860
C                                                                           00004870
  600 FORMAT('1'/////44X,'* * * * * * * * * * * * * * * * * * * * * * * *'/ 00004880
     1       44X,'*',41X,'*'/44X,'*  PICARD INTEGRATION CONVERGENCE ',      00004890
     2       'REPORT  *'/44X,'*',41X,'*'/44X,'* * * * * * * * * * * * *',  00004900
     3       ' * * * * * * * * * *'///)                                     00004910
C                                                                           00004920
  605 FORMAT(///29X,'THE ORDER OF THE CHEBYSHEV SERIES APPROXIMATING ',     00004930
     1           'THE INTEGRAND IS:  M = ',I2,'.')                          00004940
C                                                                           00004950
  610 FORMAT(///27X,'THE NUMBER OF CHEBYSHEV POINTS AT WHICH THE INTEG',    00004960
     1           'RAND IS EVALUATED IS: N+1 = ',I2,'.')                     00004970
C                                                                           00004980
  615 FORMAT(///1X,131('*')/1X,45('*'),38X,48('*')/1X,45('*'),'   END ',    00004990
     1       'OF PICARD COEFFICIENT REPORT   ',48('*')/1X,45('*'),38X,      00005000
     2       48('*')/1X,131('*')/'1')                                       00005010
C                                                                           00005020
  620 FORMAT('1'///38X,'THE ELEMENT RATE AND ELEMENT SERIES COEFFICIE',     00005030
     1           'NTS FOR VOP ELEMENT ',I1,' ARE:'//28X,'ELEMENT ',         00005040
     2           'RATE SERIES COEFFICIENTS',29X,'ELEMENT SERIES ',          00005050
     3           'COEFFICIENTS')                                            00005060
C                                                                           00005070
  625 FORMAT('1'///15X,'THE SERIES COEFFICIENTS FOR COMPONENT ',I1,         00005080
     1           ' OF THE ACCELERATION, VELOCITY AND POSITION',             00005090
     2           ' VECTORS ARE:'//8X,'ACCELERATION SERIES ',                00005100
     3           'COEFFICIENTS',9X,'VELOCITY SERIES COEFFICIENTS',          00005110
     4           13X,'POSITION SERIES COEFFICIENTS')                        00005120
C                                                                           00005130
```

```
  630 FORMAT(7X,'A(',I2,',',I1,') =',1PD25.15,5X,'B(',I2,',',I1,') =',  00005140
     1          1PD25.15,7X,'C(',I2,',',I1,') =',1PD25.15)              00005150
C                                                                       00005160
  635 FORMAT('1'////28X,'**** THE PICARD INTEGRATION METHOD FAILED TO', 00005170
     1               ' CONVERGE IN ',I2,' ITERATIONS. ****'//)          00005180
C                                                                       00005190
  640 FORMAT(///32X,'THE CONVERGENCE CRITERIA IN THIS CASE WAS THREE ', 00005200
     1               'SUCCESSIVE SOLUTIONS'/42X,'WHICH AGREE TO WITHIN ',00005210
     2               'THE FOLLOWING TOLERANCES:'/48X,'EQUATION 1:',     00005220
     3              1PD25.15/48X,'EQUATION 2:',1PD25.15/48X,'EQUATION ',00005230
     4               '3:',1PD25.15/48X,'EQUATION 4:',1PD25.15/48X,      00005240
     5               'EQUATION 5:',1PD25.15/48X,'EQUATION 6:',1PD25.15) 00005250
C                                                                       00005260
  645 FORMAT(///40X,'THE INTEGRATOR ACHIEVED CONVERGENCE IN ',I2,       00005270
     1               ' ITERATIONS.'/42X,'THE MAXIMUM NUMBER OF ',       00005280
     2               'ITERATIONS ALLOWED WAS ',I2,'.'///39X,            00005290
     3               'THE COEFFICIENTS OF THE INTEGRAND AND INTE',      00005300
     4               'GRATED SERIES'/50X,'ARE SHOWN ON THE FOLLOW',     00005310
     5               'ING PAGES.')                                      00005320
C                                                                       00005330
  650 FORMAT(///33X,'THE EQUATIONS BEING INTEGRATED ARE THE COWELL ',   00005340
     1               'EQUATIONS OF MOTION.')                            00005350
C                                                                       00005360
  655 FORMAT(///18X,'THE EQUATIONS BEING INTEGRATED ARE THE KEPLERIAN', 00005370
     1               ' VARIATION OF PARAMETER EQUATIONS (MEAN ANOMALY).')00005380
C                                                                       00005390
  660 FORMAT(///16X,'THE EQUATIONS BEING INTEGRATED ARE THE EQUINO',    00005400
     1               'CTIAL VARIATION OF PARAMETER EQUATIONS (MEAN ',   00005410
     2               'LONGITUDE).')                                     00005420
C                                                                       00005430
  665 FORMAT('1',131('*')/1X,40('*'),51X,40('*')/1X,40('*'),            00005440
     1       '   BEGIN PICARD INTEGRATION METHOD FOR THIS STEP   ',     00005450
     2       40('*')/1X,40('*'),51X,40('*')/1X,131('*')/////)           00005460
C                                                                       00005470
  670 FORMAT(///27X,'THE ARC LENGTH OF THIS INTEGRATION STEP WAS',      00005480
     1       1PD23.15,' SECONDS.')                                      00005490
C                                                                       00005500
  675 FORMAT(///13X,'THE EQUATIONS BEING INTEGRATED ARE THE EQUINO',    00005510
     1               'CTIAL VARIATION OF PARAMETER EQUATIONS (ECCENTRIC',00005520
     2               ' LONGITUDE).')                                    00005530
C                                                                       00005540
  680 FORMAT(///9X,'THE EQUATIONS BEING INTEGRATED ARE THE AVERAGED ',  00005550
     1               'EQUINOCTIAL VARIATION OF PARAMETER EQUATIONS ',   00005560
     2               '(MEAN MEAN LONGITUDE).')                          00005570
C                                                                       00005580
  685 FORMAT(///11X,'THE EQUATIONS BEING INTEGRATED ARE THE AVERAGED ', 00005590
     1               'KEPLERIAN VARIATION OF PARAMETER EQUATIONS (MEAN ',00005600
     2               'MEAN ANOMALY).')                                  00005610
C                                                                       00005620
  690 FORMAT(27X,'A(',I2,',',I1,') =',1PD25.15,25X,'B(',I2,',',I1,') =',00005630
     1               1PD25.15)                                          00005640
C                                                                       00005650
  800 FORMAT('1'////10X,'THE IEQSET INPUT FLAG MUST BE SET TO 1, 2 OR', 00005660
     1               ' 3.'/10X,'IEQSET = 1 IMPLIES INTEGRATION OF THE ', 00005670
     2               'COWELL EQUATIONS OF MOTION.'/10X,'IEQSET = 2 ',   00005680
     3               'IMPLIES INTEGRATION OF THE KEPLERIAN VOP ',       00005690
     4               'EQUATIONS (MEAN ANOMALY).'/10X,'IEQSET = 3 IMPLIE',00005700
```

```
     5              'S INTEGRATION OF THE EQUINOCTIAL VOP EQUATIONS ',  00005710
     6              '(MEAN LONGITUDE).'/10X,'IEQSET = 4 IMPLIES INTEGR',00005720
     7              'ATION OF THE EQUINOCTIAL VOP EQUATIONS (ECCENTRIC',00005730
     8              ' LONGITUDE).'/10X,'IEQSET = 5 IMPLIES INTEGRATION',00005740
     9              ' OF THE AVERAGED EQUINOCTIAL VOP EQUATIONS (MEAN ',00005750
     *             'MEAN LONGITUDE).'/10X,'IEQSET = 6 IMPLIES INTEGRAT',00005760
     *              'ION OF THE AVERAGED KEPLERIAN VOP EQUATIONS (MEAN',00005770
     *              ' MEAN ANOMALY).'//10X,'MODIFY IEQSET INPUT PARAME',00005780
     *              'TER AND RETRY.')                                   00005790
C                                                                       00005800
  810 FORMAT('1'////10X,'THE MAXIMUM ORDER CHEBYSHEV POLYNOMIAL IN TH', 00005810
     1                  'IS PROGRAM IS RESTRICTED TO BE 67 OR LESS.'/   00005820
     2                  10X,'MODIFY M INPUT PARAMETER AND RETRY.')      00005830
C                                                                       00005840
  805 FORMAT('1'////10X,'THE VALUE OF THE TOTAL NUMBER OF ITERATIONS ', 00005850
     1                  'ALLOWED MUST BE GREATER THAN TWO BECAUSE ',    00005860
     2                  'THE PROGRAM'/10X,'THE PROGRAM NEEDS THREE ',   00005870
     3                  'SUCCESSIVE SOLUTIONS THAT AGREE TO WITHIN ',   00005880
     4                  'THE CONVERGENCE CRITERIA.'/10X,'MODIFY THE ',  00005890
     5                  'INPUT PARAMETER ITERM AND RETRY.')             00005900
C                                                                       00005910
  815 FORMAT('1'////10X,'THE LARGEST SUBSCRIPT N ON THE CHEBYSHEV POI', 00005920
     1                  'NTS OF ORTHOGONALITY IS RESTRICTED TO BE 69 ', 00005930
     2                  'OR LESS.'/10X,'MODIFY N INPUT PARAMETER AND ', 00005940
     3                  'RETRY.')                                       00005950
C                                                                       00005960
      END                                                               00005970
```

SETRUN JSS1493.GTDS.UPDATE.FORT

```
      SUBROUTINE SETRUN(*,*,*,*)                                        00000010
C                                                                       00000020
C     VERSION OF NOV. 1977                                              00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO CONTROL THE OPERATION OF THE INPUT PROCESSOR                00000060
C                                                                       00000070
C     PROGRAMMER                                                        00000080
C        D. A. BUSHI, COMPUTING AND SOFTWARE, INC                       00000090
C                                                                       00000100
C     CALLING SEQUENCE                                                  00000110
C        CALL SETRUN(&NNN,&MMM,&KKK,&LLL)                               00000120
C                                                                       00000130
C     CALLING SEQUENCE PARAMETERS                                       00000140
C           NNN - THE NUMBER OF THE STATMENT IN THE CALLING SUBROUTINE  00000150
C                 THAT IS TO RECEIVE CONTROL IF DATA CARD ERRORS ARE    00000160
C                 ENCOUNTERED IN PROCESSING THE INPUT DECK.             00000170
C           MMM - THE NUMBER OF THE STATMENT IN THE CALLING SUBROUTINE  00000180
C                 THAT IS TO RECEIVE CONTROL IF AN I/O ERROR WAS        00000190
C                 ENCOUNTERED DURING RUN SET-UP.                        00000200
C           KKK - THE NUMBER OF THE STATEMENT IN THE CALLING            00000210
C                 SUBROUTINE THAT IS TO RECEIVE CONTROL IF ALL          00000220
C                 CARD DECKS HAVE BEEN PROCESSED.                       00000230
C           LLL - THE NUMBER OF THE STATEMENT IN THE CALLING            00000240
C                 SUBROUTINE THAT IS TO RECEIVE CONTROL IN THE CRT      00000250
C                 INPUT MODE.                                           00000260
C                                                                       00000270
C     COMMON BLOCK PARAMETERS                                           00000280
C          CONST                                                        00000290
C             DTR     - CONVERSION FACTOR FROM DEGREES TO RADIANS       00000300
C          CNVRG                                                        00000310
C             CON(6)  - CONVERGENCE CRITERIA FOR THE SIX STATE          00000320
C                       EQUATIONS IN THE PICARD-CHEBYSHEV INTEGRATOR    00000330
C             CONKEP  - CONVERGENCE CRITERION FOR THE ITERATIVE          00000340
C                       SOLUTION OF THE KEPLER EQUATION                 00000350
C          DCINPT                                                       00000360
C             CARTES  - IF TRUE THE CARTESIAN OBSERVATION SET           00000370
C             YMDNEW  - YYMMDD OF ADVANCED EPOCH                        00000380
C             HMSNEW  - HHMMSS.SSS OF ADVANCED EPOCH                    00000390
C          DCINT                                                        00000400
C             TTO     - TIME IN A.1 SECONDS FROM EPOCH                  00000410
C          EAINT                                                        00000420
C             LOVWT   - PRINT EAP OVERRIDE WEIGHTS OPTION SWITCH        00000430
C             OVWT    - EAP OVERRIDE WEIGHTS OPTION SWITCH              00000440
C             SMAP    - MAPPING START TIME                              00000450
C                       SMAP(1)=YYMMDD.                                 00000460
C                       SMAP(2)=HHMMSS.SSS                              00000470
C             STRACK  - PARTIAL TRACKING REPORT START TIME              00000480
C                       STRACK(1)=YYMMDD.                               00000490
C                       STRACK(2)=HHMMSS.SSS                            00000500
C          EDIT                                                         00000510
C             ETIME   - LATEST DESIRED OBSERVATION TIME                 00000520
C             IDLTOB  - ARRAY OF OBSERVATION NUMBERS FOR EDITING        00000530
C             NDLT    - NUMBER OF ENTRIES IN IDLTOB ARRAY               00000540
C             STIME   - EARLIEST DESIRED OBSERVATION TIME               00000550
C          FILES                                                        00000560
C             NOIDS   - FRN OF THE COMMON BLOCK NOMINAL VALUES DATA SET 00000570
```

```
C            NOPTS   - FRN OF THE OPTIONS INPUT DATA SET                00000580
C            NOUT    - FRN OF THE PRINTER OUTPUT                        00000590
C          FRC                                                          00000600
C            AB      - MEAN EQUATORIAL RADIUS OF THE BODY (KM)          00000610
C            BSQ     - ONE MINUS ECCENTRICITY SQUARED (1-E**2)          00000620
C            ESQ     - ECCENTRICITY OF CENTRAL BODY SQUARED             00000630
C            FLAT    - INVERSE OF FLATTENING COEFFICIENTS ARRAY         00000640
C            ICENT   - CENTRAL BODY INDICATOR                           00000650
C            I50     - INTEGRATOR COORDINATE SYSTEM REFERENCE           00000660
C            VAB     - RADIUS OF CENTRAL BODY SQUARED                   00000670
C          NPCNTL                                                       00000680
C            MORD    - MAXIMUM ORDER OF THE CHEBYSHEV APPROXIMATING     00000690
C                      SERIES                                           00000700
C            NSUB    - MAXIMUM SUBSCRIPT OF THE CHEBYSHEV ORTHOGONALITY 00000710
C                      POINTS                                           00000720
C            IEQSET  - TYPE EQUATIONS BEING INTEGRATED BY THE PICARD-   00000730
C                      CHEBYSHEV INTEGRATOR                             00000740
C            ITERM   - TERMINAL ITERATION NUMBER                        00000750
C          SATPOS                                                       00000760
C            ELEMS   - KEPLERIAN ELEMENTS ARRAY                         00000770
C            DELTAT  - OUTPUT PRINT INTERVAL                            00000780
C            HMSFN   - END OF OUTPUT HHMMSS.SSSS                        00000790
C            HMSIC   - EPOCH HHMMSS.SSSS                                00000800
C            HMSOUT  - START OF OUTPUT HHMMSS.SSSS                      00000810
C            IAIREF  - YEAR OF TIME REFERENCE                           00000820
C            IBODY   - CENTRAL BODY REFERENCE FOR ELEMENTS              00000830
C            ICORD   - COORDINATE SYSTEM REFERENCE FOR ELEMENTS         00000840
C            ICTYPE  - ELEMENTS COORDINATE SYSTEM TYPE                  00000850
C            IDSAT   - SATELLITE ID NUMBER                              00000860
C            ORBEL   - AUXILIARY ORBITAL ELEMENTS ARRAY                 00000870
C            POSI    - X,Y,Z OF CARTESIAN ELEMENTS                      00000880
C            REFJUL  - JULIAN DATE CORRESPONDING TO YMDREF              00000890
C            SATNAM  - SATELLITE NAME                                   00000900
C            SPHCOR  - SPHERICAL COORDINATES ARRAY                      00000910
C            VELI    - X,Y,Z OF CARTESIAN ELEMENTS                      00000920
C            YMDFN   - END OF OUTPUT YYMMDD.                            00000930
C            YMDIC   - EPOCH YYMMDD.                                    00000940
C            YMDOUT  - START OF OUTPUT YYMMDD.                          00000950
C            YMDREF  - REFERENCED TIME OF TRUE SYSTEM (YYMMDD)          00000960
C          SECTN                                                        00000970
C            INDSEC  - SECTIONING  INDICATORS AND FLAGS ARRAY           00000980
C            NCONDT  - CURRENT CONDITION FLAG                           00000990
C            SECMCD  - RADIAL DISTANCES STEPSIZES,AND ERROR BOUNDS      00001000
C          STAGEO                                                       00001010
C            OBCOR   - OBSERVATION CORRECTION FLAGS ARRAY               00001020
C            NOISTA  - NUMBER OF STATIONS IN THE PROBLEM                00001030
C          SWITCH                                                       00001040
C            IND(4)  - COMPUTATION METHOD FOR PARTIALS                  00001050
C            IND(5)  - EARLY ORBIT COMPUTATION METHOD                   00001060
C            IND(8)  - RESTORE NOMINAL COMMON BLOCK VALUES              00001070
C            IND(20) - PICARD-CHEBYSHEV INTEGRATOR USED TO INTEGRATE    00001080
C                      VOP EQUATIONS                                    00001090
C            IND(31) - EVAL INITIALIZE SWITCH                           00001100
C            IND(40) - STATE PARTIALS REQUIRED                          00001110
C            IND(45) - SCOPE IN USE INDICATOR                           00001120
C            IND(46) - HANDS-ON MODE                                    00001130
C            IND(47) - LANDMARK DATA CARD INPPUT                        00001140
```

```
C             IND(48) - RUN TYPE INDICATOR                                    00001150
C             IND(49) - PCE OBSERVATION INPUT                                 00001160
C             IND(50) - REAL-TIME OBSERVATION INPUT                           00001170
C             IND(51) - OBSERVATION WORKING FILE INDICATOR                    00001180
C             IND(52) - GTDS OBSERVATION TAPE INPUT                           00001190
C             IND(53) - DODS OBSERVATION TAPE INPUT                           00001200
C             IND(54) - GTDS OBSERVATION DISK FILE INPUT                      00001210
C             IND(55) - DODS OBSERVATION DISK FILE INPUT                      00001220
C             IND(56) - OPTICAL ASPECT 9-TRACK TAPE INDICATOR                 00001230
C             IND(57) - RAW OBSERVATIONS (PERMANENT FILES) INPUT              00001240
C             IND(58) - OBSERVATIONS CARD INPUT                               00001250
C             IND(63) - CHANGE TO STANDARD PARAMETERS INDICATOR               00001260
C             IND(68) - IONOSPHERIC REFRACTION WORKING FILE INDICATOR         00001270
C             IND(88) - EPOCH ADVANCE INDICATOR                               00001280
C             IND(95) - EPOCH AND ELEMENTS CARDS READ FROM INPUT DECK         00001290
C             IPRINT  - PRINT INDICATOR                                       00001300
C             KSTATE  - LABEL NUMBER ARRAY FOR STATE UNKNOWNS(1-9,19)         00001310
C             LINES   - CURRENT LINE NUMBER OF PAGE OUTPUT                    00001320
C             NERCCD  - NUMBER OF CONTROL CARDS IN THE INPUT DECK             00001330
C             NERCRD  - CARD COUNTER                                          00001340
C             NCDERR  - CUMULATIVE CARD ERROR COUNTER                         00001350
C             NSTATE  - NUMBER OF STATE UNKNOWNS (MAX=6)                      00001360
C          TITLE                                                              00001370
C             HEADER  - TITLE ARRAY                                           00001380
C             MODWF   - MODEL NUMBERS ARRAY                                   00001390
C                                                                             00001400
C     METHOD                                                                  00001410
C          SETRUN READS-IN AND PROCESSES THE RUN CONTROL CARD. THE            00001420
C          MANDATORY CARDS FOR THE RUN ARE READ-IN AND PROCESSED. IF          00001430
C          THE RUN SET-UP CONTAINS OPTIONS OR DATA DECKS, SETRUN              00001440
C          IDENTIFIES THE DECK AND CALLS THE SUBROUTINE WHICH PROCESSES       00001450
C          THAT DECK.                                                         00001460
C                                                                             00001470
C     CALLED SUBROUTINES                                                      00001480
C          DELTIM                                                             00001490
C          ELEME                                                              00001500
C          ERROUT                                                             00001510
C          GETDSP                                                             00001520
C          IRWF                                                               00001530
C          MSGNTR                                                             00001540
C          SETDC                                                              00001550
C          SETCMP                                                             00001560
C          SETAVR                                                             00001570
C          SETDM                                                              00001580
C          SETORB                                                             00001590
C          UPPAGE                                                             00001600
C          WFCONT                                                             00001610
C                                                                             00001620
C     PROGRAM MODIFICATIONS                                                   00001630
C          1.  DATE  :  05/15/79                                              00001640
C              NAME  :  CAPTAIN JEFFREY S. SHAVER, USAF                       00001650
C              REASON:  EXPANDED 'ORBTYPE' CARD PROCESSING FOR                00001660
C                       PICARD-CHEBYSHEV ORBIT GENERATION CAPABILITY.         00001670
C          2.  DATE  :  06/06/79                                              00001680
C              NAME  :  CAPTAIN JEFFREY S. SHAVER, USAF                       00001690
C              REASON:  EXPANDED 'ORBTYPE' CARD PROCESSING FOR VOP            00001700
C                       ORBIT GENERATOR OPTION TO INCLUDE PICARD-             00001710
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

C                     CHEBYSHEV INTEGRATION OF THE VOP EQUATIONS      00001720
C                     OF MOTION.                                      00001730
C                                                                     00001740
C**********************************START OF PROGRAM*********************00001750
C                                                                     00001760
      IMPLICIT REAL*8(A-H,O-Z)                                        00001770
      REAL*4          R4EDIT                                          00001780
      INTEGER*2       I2EDIT                                          00001790
      LOGICAL         LSSC        ,MULORB     ,IPLOTH                 00001800
      LOGICAL*1       LGDCNP      ,CARTES     ,RETERR     ,METYPE     00001810
      LOGICAL*1       WFCALL      ,OBCOR      ,DCSTRT     ,DCEND      00001820
      LOGICAL*1       LEAP        ,SENSIT     ,DROYET     ,DRCOLD     00001830
      LOGICAL*1       OVHT        ,LOVHT      ,CONSHH     ,LSTA       00001840
      LOGICAL*1       ISEEHA                                          00001850
C                                                                     00001860
      DIMENSION       XTO(3)      ,XDTO(3)                            00001870
      DIMENSION       POSAPR(3)   ,VELAPR(3)                          00001880
      DIMENSION       PVINT(6)    ,AMINT(6)   ,BMELS(6)               00001890
      DIMENSION       POSI(3)     ,VELI(3)    ,ELEMS(6)   ,SPHCOR(6)  00001900
C                                                                     00001910
      DIMENSION       SECIN(5,10)                                     00001920
      DIMENSION       SECMCD(50,10)           ,INDSEC(30,10)          00001930
      DIMENSION       FLAT(11)    ,AB(11)     ,KSTATE(6)              00001940
      DIMENSION       TOL(10)                                         00001950
C                                                                     00001960
      DIMENSION       FREQ(6)     ,SGDP(7,20) ,ISG(9,20)              00001970
      DIMENSION       SGSP(14,20)                                     00001980
      DIMENSION       STRACK(2)   ,SMAP(2)                            00001990
      DIMENSION       OBCOR(4,20)                                     00002000
      DIMENSION       TSTART(20)  ,TEND(20)   ,LEVSTR(2)  ,LEVEND(2)  00002010
      DIMENSION       STIME(2)    ,ETIME(2)   ,SINGLE(30)             00002020
      DIMENSION       INTTYP(2)                                       00002030
C                                                                     00002040
      DIMENSION       FIELD(10)   ,CDSMAN(9)  ,TERMCD(8)              00002050
      DIMENSION       IA(3)       ,RA(3)                              00002060
      DIMENSION       NB(9)       ,XB(6,8)                            00002070
C                                                                     00002080
      COMMON/ANAVIN/  INANAV(87)                                      00002090
      COMMON/BRWOPT/  DOTA        ,IDSECL     ,IDOSM                  00002100
      COMMON/CNVRG /  CON(6)      ,CONKEP                             00002110
      COMMON/CONST /  DPPP(3)     ,DTR                                00002120
      COMMON/CONTRL/  ICNTL(6)                                        00002130
      COMMON/DCINPT/  DPDCNP(300) ,INTDCN(200),LGDCNP(72)             00002140
      COMMON/DCINT /  DPDCI(432)  ,INTDCI(26)                         00002150
      COMMON/EAINT /  DPEAP(60)   ,INEAP(20)  ,LEAP(19)   ,           00002160
     *                SENSIT      ,DROYET     ,DRCOLD                 00002170
      COMMON/EDIT  /  R8EDIT(74)  ,R4EDIT(20) ,I4EDIT(107),I2EDIT(40) ,00002180
     *                METYPE(20,3)                                    00002190
      COMMON/FILES /  INTFI(100)                                      00002200
      COMMON/FRC   /  DPFRC(1300) ,INTFRC(50)                         00002210
      COMMON/HALO  /  PVHALO(6)   ,PVHALI(6)  ,POSILP(3)  ,VELILP(3)  ,00002220
     *                PPLP(6,2)   ,SIGLP(6)   ,CVL(21)    ,EXCCOV(6)  ,00002230
     *                ISOHAL      ,INDAPC     ,INDEXC     ,ISEEHA     00002240
      COMMON/HSTPLT/  HST(31)     ,IHST(33)   ,IPLOTH                 00002250
      COMMON/INTEG /  DPINT(149)  ,INTINT(7)                          00002260
      COMMON/MATRIX/  DPNTR(57)                                       00002270
      COMMON/NPCNTL/  MORD        ,NSUB       ,IEQSET     ,ITERM      00002280
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

      COMMON/NPQBLK/   TQ2(14)      ,TNPQ2(14)   ,NPQQV(2,14),INDPRT(14) ,00002290
     *                 NPQP2        ,NPQP3       ,NPQCT       ,NPQ2S       ,00002300
     *                 NPQ3S        ,NPQSLV                                 00002310
      COMMON/NUMSEC/   DUM1         ,DUM2        ,DUM3        ,DUM4        ,00002320
     *                 INNRTE                                               00002330
      COMMON/R4COM /   DPR4(20)     ,IRES4                                  00002340
      COMMON/SATMAN/   DPSTM(164)   ,INTSTM(10)                             00002350
      COMMON/SATPOS/   DPSS(100)    ,INTSS(6)                               00002360
      COMMON/SECTN /   DPSCT(675)   ,INTSCT(325)                            00002370
      COMMON/SLPOPT/   DPSLP(1)     ,INTSLP(19)                             00002380
      COMMON/SSTCOM/   DPSSC(70)    ,ISSC(18)    ,LSSC(2)                   00002390
      COMMON/STAGEO/   DPSTA(435)   ,INTSTA(182),LSTA(80)                   00002400
      COMMON/SWITCH/   IND(230)                                             00002410
      COMMON/TEMPR1/   ITER01                                               00002420
      COMMON/THRUST/   DPTHR(150)   ,INTHR(50)                              00002430
      COMMON/TITLE /   HEADER(9,6) ,MODNF(6)                                00002440
      COMMON/WEICHG/   WEINEW(10)   ,WTNUM(10)   ,WMIX(2,3)                 00002450
      COMMON/WORKER/   DPWOR(3264) ,INTWOR(13)                              00002460
      COMMON           A1           ,R1          ,R2          ,R3          ,00002470
     *                 I1           ,I2          ,I3          ,IDUM        ,00002480
     *                 AS(7)        ,TR1         ,TR2         ,NUM         ,00002490
     *                 NTP          ,ITM         ,ICARD(8)                  00002500
C                                                                           00002510
      EQUIVALENCE      (AMINT(1)                  ,DPDCI(245)              ) 00002520
      EQUIVALENCE      (TTIME                     ,DPDCNP(295)             ) 00002530
      EQUIVALENCE      (LINES                     ,IND(148)                ) 00002540
      EQUIVALENCE      (INDG                      ,INTDCN(191)             ) 00002550
      EQUIVALENCE      (NPBIA                     ,INTHR(20)               ) 00002560
C                                                                           00002570
      EQUIVALENCE  (STPMIN     ,DPINT(123)  ),(TOL(1)     ,DPINT(124)   ) 00002580
      EQUIVALENCE  (LOVHT      ,LEAP(12)    ),(OVHT       ,LEAP(11)     ) 00002590
      EQUIVALENCE  (NOISTA     ,INTSTA(181) ),(NS         ,INTSTA(182) ),00002600
     *             (OBCOR(1,1) ,LSTA(1)     )                             00002610
      EQUIVALENCE  (TSTART(1)  ,R8EDIT(5)   ),(TEND(1)    ,R8EDIT(25)   ),00002620
     *             (LEVSTR(1)  ,I4EDIT(4)   ),(LEVEND(1)  ,I4EDIT(6)    ) 00002630
      EQUIVALENCE  (STIME(1)   ,R8EDIT(1)   ),(ETIME(1)   ,R8EDIT(3)    ),00002640
     *             (SINGLE(1)  ,R8EDIT(45)  )                             00002650
C                                                                           00002660
      EQUIVALENCE      (DPSTA(141)              ,SGSP(1,1)               ) 00002670
      EQUIVALENCE      (DPDCNP(135)             ,YMDNEW                  ),00002680
     *                 (DPDCNP(136)             ,HMSNEW                  ) 00002690
      EQUIVALENCE      (DPSCT(1)                ,SECIN(1,1)              ),00002700
     *                 (DPSCT(151)              ,SECMCD(1,1)             ),00002710
     *                 (INTSCT(11)              ,INDSEC(1,1)             ),00002720
     *                 (INTSCT(312)             ,NTSECS                  ),00002730
     *                 (INTSCT(313)             ,NCONDT                  ),00002740
     *                 (INTSCT(314)             ,NMANVS                  ),00002750
     *                 (INTSCT(321)             ,IFLAG                   ) 00002760
      EQUIVALENCE      (IND(101)                ,JCTYPE                  ) 00002770
      EQUIVALENCE      (IND(226)                ,INDNEX                  ) 00002780
      EQUIVALENCE      (INTFI(57)               ,NOIDS                   ) 00002790
      EQUIVALENCE      (INTSCT(322)             ,IASDES                  ) 00002800
      EQUIVALENCE      (INTSCT(323)             ,INMAN                   ) 00002810
      EQUIVALENCE      (INTSCT(324)             ,NFREQ                   ) 00002820
      EQUIVALENCE      (INTSCT(325)             ,NODDES                  ) 00002830
      EQUIVALENCE      (IND(216)                ,NDATAB                  ) 00002840
      EQUIVALENCE      (IND(200)                ,INDSW                   ) 00002850
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

     ' EQUIVALENCE      (INTDCN(170)              ,NSTOPT                ) 00002860
      EQUIVALENCE       (INTDCN(59)               ,NINVRS                ) 00002870
      EQUIVALENCE       (LGDCNP(47)               ,CARTES                ) 00002880
      EQUIVALENCE       (LGDCNP(51)               ,CONSWH                ) 00002890
C                                                                          00002900
      EQUIVALENCE  (DPSTA(1)     ,SGDP(1,1) ),(INTSTA(1)    ,ISG(1,1)   ) 00002910
      EQUIVALENCE  (INTDCN(163)  ,IEPN      ),(INTDCN(181) ,MTRACK      ) 00002920
      EQUIVALENCE  (DPDCI(1)     ,XTO(1)    ),(DPDCI(4)    ,XDTO(1)     ) 00002930
      EQUIVALENCE  (ISSC(1)      ,IJK1      ),(ISSC(5)     ,IDSAT2      ),00002940
     *             (DPSSC(60)    ,SSYMD     ),(DPSSC(61)   ,SSHMS       ),00002950
     *             (ISSC(8)      ,INTTYP(1) ),(LSSC(2)     ,MULORB      ) 00002960
      EQUIVALENCE  (DPEAP(1)     ,STRACK(1) ),(DPEAP(5)    ,SMAP(1)     ) 00002970
      EQUIVALENCE  (DPFP(1)      ,TWOPI     ),(DPMTR(37)   ,GHA         ) 00002980
      EQUIVALENCE  (INTFI(18)    ,NWSLP     ),(INTFI(78)   ,NTCD        ),00002990
     *             (INTFI(14)    ,NPSLP     ),(INTFI(71)   ,NSWSLP      ) 00003000
      EQUIVALENCE  (INTFI (21)   ,NOPTS     ),(INTFI(22)   ,NOUT        ) 00003010
      EQUIVALENCE  (DPFRC(165)   ,ESQ       ),(DPFRC(1270) ,BSQ         ),00003020
     *             (DPFRC(1279)  ,VAB       ),(INTFRC(1)   ,ICENT       ),00003030
     *             (DPFRC(68)    ,FLAT(1)   ),(DPFRC(24)   ,AB(1)       ),00003040
     *             (DPFRC(1274)  ,REFDA4    ),(IND(33)     ,IWZZ        ) 00003050
      EQUIVALENCE  (INTFRC(2)    ,I50       ),(DPDCI(251)  ,PVINT(1)    ) 00003060
      EQUIVALENCE  (DPSS(3)      ,FOSI(1)   ),(DPSS(6)     ,VELI(1)     ),00003070
     *             (DPSS(11)     ,ELEMS(1)  ),(DPSS(17)    ,SPHCOR(1)   ),00003080
     *             (DPSS(64)     ,YMDIC     ),(DPSS(65)    ,HMSIC       ),00003090
     *             (DPSS(66)     ,YMDOUT    ),(DPSS(67)    ,HMSOUT      ),00003100
     *             (DPSS(68)     ,YMDFN     ),(DPSS(69)    ,HMSFN       ),00003110
     *             (DPSS(70)     ,DELTAT    ),(DPSS(71)    ,YMDREF      ),00003120
     *             (DPSS(72)     ,REFJUL    ),(DPSS(73)    ,SATNAM      ),00003130
     *             (DPSS(95)     ,EMELS(1)  ),                            00003140
     *             (INTSS(1)     ,IBODY     ),(INTSS(2)    ,ICORD       ),00003150
     *             (INTSS(4)     ,ICTYPE    ),(INTSS(5)    ,IDSAT       ),00003160
     *             (INTSS(6)     ,IA1REF    )                            00003170
      EQUIVALENCE  (IND(165)     ,NERCRD    ),(IND(166)    ,IPRINT      ),00003180
     *             (IND(167)     ,NODERR    )                            00003190
      EQUIVALENCE  (IND(154)     ,NSTATE    ),(IND(155)    ,KSTATE(1)   ) 00003200
      EQUIVALENCE  (IND(191)     ,INDTYP    ),(IND(1)      ,IND1        ) 00003210
C                                                                          00003220
      EQUIVALENCE  (IA(1),I1),     (IA(2),I2),     (IA(3),I3)            00003230
      EQUIVALENCE  (RA(1),R1),     (RA(2),R2),     (RA(3),R3)            00003240
C                                                                          00003250
      DATA CONTRL     /'CONTROL '/                                       00003260
      DATA DUM        /'        '/                                       00003270
      DATA FDSS       /'  GTDS  '/                                       00003280
      DATA OUTPUT     /'OUTPUT  '/                                       00003290
      DATA PRGM       /' PROGRAM'/                                       00003300
      DATA PROMPT     /'PROMPT  '/                                       00003310
      DATA REALTM     /'REALTIME'/                                       00003320
      DATA STOPRT     /'NO-PRINT'/                                       00003330
C                                                                          00003340
      DATA FIELD      /'DC      ',                                       00003350
     1                 'EPHEM   ',                                       00003360
     2                 'DATASIM ',                                       00003370
     3                 'COMPARE ',                                       00003380
     4                 'DATAMGT ',                                       00003390
     5                 'FILERPT ',                                       00003400
     6                 'ANALYSIS',                                       00003410
     7                 'EARLYORB',                                       00003420
```

```
SETRUN              JSS1493.GTDS.UFDATE.FORT

     8                'ATOMAT  ',                                          00003430
     9                'FILTER  '/                                          00003440
C                                                                         00003450
      DATA CDSMAN     /'ELEMENT1',                                        00003460
     1                 'ELEMENT2',                                        00003470
     2                 'EPOCH   ',                                        00003480
     3                 'OUTPUT  ',                                        00003490
     4                 'ORBTYPE ',                                        00003500
     5                 'OBSINPUT',                                        00003510
     6                 'OBSNUME ',                                        00003520
     7                 'TYPE    ',                                        00003530
     8                 'LTYPE   '/                                        00003540
C                                                                         00003550
      DATA TERMCD     /'DMOPT  ',                                        00003560
     1                 'OSOPT  ',                                        00003570
     2                 'FIN    ',                                        00003580
     3                 'DCOPT  ',                                        00003590
     4                 'COMPOPT',                                        00003600
     5                 'PPROPT ',                                        00003610
     6                 'FILOPT ',                                        00003620
     7                 'EAOPT  '/                                        00003630
C                                                                         00003640
C     DEFAULT STEPSIZE FOR INTEGRATION                                    00003650
C                                                                         00003660
      DATA STPSZE     / 24.0D0     /                                      00003670
      DATA TN4        / 1.D4       /                                      00003680
      DATA FREQ       / 148.26D0   ,136.2D0    ,2200.0D0   ,2200.0D0  ,  00003690
     *                  2198.875D0 ,2198.875D0 /                         00003700
C                                                                         00003710
C**** PROCESS CONTROL CARDS ***************************************       00003720
C                                                                         00003730
      CONSWH = .FALSE.                                                    00003740
      ICRTYP=0                                                            00003750
      DO 7 I = 1,10                                                       00003760
    7 WTNUM(I) = 0.D0                                                     00003770
      DO 10 I=1,3                                                         00003780
      WMIX(1,I) = 0.D0                                                    00003790
   10 WMIX(2,I) = 0.D0                                                    00003800
      IPLOTH=.FALSE.                                                      00003810
      LGDCNP(56)=.FALSE.                                                  00003820
      LGDCNP(57)=.FALSE.                                                  00003830
      LGDCNP(58)=.FALSE.                                                  00003840
      NGATE=0                                                             00003850
      IND(76) = 1                                                         00003860
      IASDES=0                                                            00003870
      NCDDES=0                                                            00003880
      NFREQ=9999999                                                       00003890
      INMAN=9999999                                                       00003900
      IND(50)=2                                                           00003910
C     SET ITERATION COUNTER TO ZERO-SOR                                   00003920
      ITER01=0                                                            00003930
C     SET DEFAULT EPHEM CENTRAL BODY IN COORDINATES SYSTEM                00003940
      IND (90) =1                                                         00003950
   35 READ(NOPTS,1830,END=815,ERR=800)A8,IOK,ITM                          00003960
      IF(AS(1).EQ.DUM)   GO TO 35                                         00003970
      IF (AS(1).NE.CONTRL) GO TO 50                                       00003980
      IF(AS(2).EQ.PROMPT)  GO TO 35                                       00003990
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

C                                                                       00004000
C     TEST FOR RESTORING OF NOMINAL COMMON BLOCK VALUE                  00004010
C                                                                       00004020
      IF( AS(6).EQ.DUM )  GO TO 38                                      00004030
      READ(NOIDS,END=810,ERR=800)                                       00004040
     *               DFDCNP   ,INTDCN   ,LGDCNP   ,DPDCI   ,INTDCI   , 00004050
     *               DFFRC    ,INTFRC   ,DPINT    ,INTINT  ,DFSTM    , 00004060
     *               INTSTM   ,DPSS     ,INTSS    ,DFSCT   ,INTSCT   , 00004070
     *               DPSLP    ,INTSLP   ,DPSTA    ,INTSTA  ,LSTA     , 00004080
     *               HEADER   ,MODWF    ,IND      ,INANAV  ,DFEAP    , 00004090
     *               INEAP    ,LEAP     ,SENSIT   ,DROYET  ,DROOLD     00004100
      REWIND NOIDS                                                      00004110
C                                                                       00004120
      IND(183)=2                                                        00004130
C     RESTORE SWITCH FOR TIMING COEFFICIENTS                            00004140
      IND(199)=1                                                        00004150
   38 NERCRD=NERCRD+1                                                   00004160
      IF( AS(3).EQ.REALTM)   IND(50)=1                                  00004170
      IPRINT = 1                                                        00004180
      IF (A8(4).EQ.STOPRT) IPRINT=0                                     00004190
      IND(95) = 2                                                       00004200
C     TEST FOR ELEMENTS TO BE PASSED FROM PREVIOUS RUN                  00004210
      IF(AS(5).EQ.DUM) GO TO 6                                          00004220
      IND(95) = 3                                                       00004230
      IF(AS(5).NE.OUTPUT) GO TO 6                                       00004240
      JCTYPE = ICTYPE                                                   00004250
      ICTYPE = 1                                                        00004260
      ICORD = I50                                                       00004270
      IBODY = ICENT                                                     00004280
C     INDNEX VARIABLE IS SET IN CDSEXEC                                 00004290
      IF ( INDNEX.EQ.2 ) GO TO 71                                       00004300
      POSI(1) = PVINT(1)                                                00004310
      POSI(2) = PVINT(2)                                                00004320
      POSI(3) = PVINT(3)                                                00004330
      VELI(1) = PVINT(4)                                                00004340
      VELI(2) = PVINT(5)                                                00004350
      VELI(3) = PVINT(6)                                                00004360
      IF(INDTYP.NE.2)GO TO 22                                           00004370
      ICORD=3                                                           00004380
      ICTYPE=4                                                          00004390
      DO 22 I=1,6                                                       00004400
      ENELS(I) = AMINT(I)                                               00004410
   22 CONTINUE                                                          00004420
      GO TO 6                                                           00004430
C                                                                       00004440
 71   DO 72 I = 1,3                                                     00004450
      POSI (I) = XTO (I)                                                00004460
 72   VELI (I) = XDTO (I)                                               00004470
C                                                                       00004480
      YMDOUT = YMDFN                                                    00004490
      HMSOUT = HMSFN                                                    00004500
      YMDSUB = YMDFN                                                    00004510
      HMSSUB = HMSFN                                                    00004520
      YMDIC  = YMDFN                                                    00004530
      HMSIC  = HMSFN                                                    00004540
      CALL DELTIM( 1, YMDSUB, HMSSUB, REFDA4, DDQZ, DDQW )              00004550
C                                                                       00004560
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

    6 SATNAM = A8(7)                                                    00004570
      IF(IND(95).EQ.3) GO TO 11                                         00004580
      NPQP2=0                                                           00004590
      NPQP3=0                                                           00004600
      NPQCT=0                                                           00004610
      NPQSLV=0                                                          00004620
      NPQ2S=0                                                           00004630
      NPQ3S=0                                                           00004640
   11 CONTINUE                                                          00004650
C                                                                       00004660
C     STORE SATELLITE NUMBER                                            00004670
C                                                                       00004680
      IDSAT = ITM                                                       00004690
      NSTOPT=0                                                          00004700
C                                                                       00004710
C     INITIALIZATION SECTION                                            00004720
C                                                                       00004730
C     SET CONSIDER SWITCH OFF FOR STACKED INPUT DECKS, SOME OF WHICH    00004740
C     MAY NOT BE CONSIDER RUNS                                          00004750
C                                                                       00004760
      DO 3 I=1,6                                                        00004770
    3 ICNTL(I)=0                                                        00004780
      LGDCNP(51) = .FALSE.                                              00004790
      RETERR = .FALSE.                                                  00004800
      HFCALL = .FALSE.                                                  00004810
      DCSTRT = .FALSE.                                                  00004820
      DCEND = .FALSE.                                                   00004830
C                                                                       00004840
      DO 5 I=1,9                                                        00004850
      HEADER(I,1) = DUM                                                 00004860
    5 CONTINUE                                                          00004870
      HEADER(3,1) = FDSS                                                00004880
      HEADER(5,1) = PRGM                                                00004890
C                                                                       00004900
C     INITIALIZE COUNTERS AND SWITCHES                                  00004910
C                                                                       00004920
      IDCRUN = 0                                                        00004930
      IFLDM = 0                                                         00004940
      IM = 0                                                            00004950
      NDLT = 0                                                          00004960
      IND(2) = 2                                                        00004970
      IND(4) = 1                                                        00004980
      IND(77)=0                                                         00004990
C                                                                       00005000
C     TURN OFF ALL FILE GENERATION SWITCHES                             00005010
      IND(151)=2                                                        00005020
      IND(98)=2                                                         00005030
      IND(85)=2                                                         00005040
      MULORB=.FALSE.                                                    00005050
      IND(217)=0                                                        00005060
C                                                                       00005070
C     ZERO OBS FILE TIMES                                               00005080
      STIME(1)=0.D0                                                     00005090
      STIME(2)=0.D0                                                     00005100
      ETIME(1)=0.D0                                                     00005110
      ETIME(2)=0.D0                                                     00005120
C    ZERO ERROR ANALYSIS PROGRAM TIMES                                  00005130
```

```
      STRACK(1)=0.0D0                                                   00005140
      STRACK(2)=0.0D0                                                   00005150
      SMAP(1)=0.0D0                                                     00005160
      SMAP(2)=0.0D0                                                     00005170
      INEAP( 20 ) = 10                                                  00005180
      OVWT = .FALSE.                                                    00005190
      LOVWT = .FALSE.                                                   00005200
C     TURN OFF PARTIAL SWITCHES                                         00005210
C     SET THE DEFAULT VALUE FOR TRANSFORMATION PARTIALS SWITCH          00005220
      IND(39) = 1                                                       00005230
      IND(40) = 2                                                       00005240
      IND(41) = 2                                                       00005250
      IND(42) = 2                                                       00005260
      IND(43) = 2                                                       00005270
      IND(59) = 0                                                       00005280
      NPBIA=0                                                           00005290
C                                                                       00005300
      IND(68) = 2                                                       00005310
      IND(73) = 0                                                       00005320
      INDSW=2                                                           00005330
      IND(79)=2                                                         00005340
      IND(83) = 2                                                       00005350
      IND(102)=0                                                        00005360
      IND(103)=0                                                        00005370
      INDG = 0                                                          00005380
      NCONDT=0                                                          00005390
      NERCRD = 0                                                        00005400
      IERROR = 0                                                        00005410
      MSGERR = 0                                                        00005420
C                                                                       00005430
C     INITIALIZE SECTIONING INDICATORS                                  00005440
C                                                                       00005450
      INDSEC(14,1)=2                                                    00005460
      INDSEC(15,1)=2                                                    00005470
      INDSEC(16,1)=2                                                    00005480
      INDSEC(19,1)=2                                                    00005490
      INDSEC(20,1)=2                                                    00005500
      INDSEC(29,1)=2                                                    00005510
      INDSEC(30,1)=2                                                    00005520
      NTSECS=1                                                          00005530
C     INITIALIZE EDITING SWITCHES                                       00005540
      DO 14 I=2,8                                                       00005550
   14 I4EDIT(I)=0                                                       00005560
      I4EDIT(1)=2                                                       00005570
      I4EDIT(107)=0                                                     00005580
C     TURN OFF OBSERVATION INPUT SOURCE SWITCHES                        00005590
      IND(47) = 2                                                       00005600
      IND(49) = 2                                                       00005610
      IND(34)=2                                                         00005620
      IND(216)=2                                                        00005630
      IND(219)=2                                                        00005640
      IND( 223 ) = 2                                                    00005650
      DO 15 I=52,58                                                     00005660
   15 IND(I) = 2                                                        00005670
C                                                                       00005680
C     ZERO STATION BIASES                                               00005690
C                                                                       00005700
```

```
      DO 8 I=1,20                                                       00005710
      DO 8 M=1,14                                                       00005720
      SGSP(M,I)=0.D0                                                    00005730
    8 CONTINUE                                                          00005740
C                                                                       00005750
C     TURN OFF SPECIAL MINITRACK BIASES SWITCH                          00005760
C                                                                       00005770
      MTRACK=0                                                          00005780
C                                                                       00005790
C  INITIALIZATION FOR ORBIT FILE RETRIEVAL (ORBTYP=6) START/END TIME    00005800
      IF ( INDNEX.EQ.2 ) GO TO 9                                        00005810
      YMDCUT=0.0D0                                                      00005820
      HMSCUT=0.0D0                                                      00005830
      YMDFN=0.0D0                                                       00005840
      HMSFN=0.0D0                                                       00005850
 9    CONTINUE                                                          00005860
C                                                                       00005870
C     SET DEFAULT FOR STATE PARTIALS                                    00005880
C                                                                       00005890
      DO 20 I=1,6                                                       00005900
   20 KSTATE(I) = I                                                     00005910
      NSTATE = 6                                                        00005920
C                                                                       00005930
C         THE FOLLOWING CONVENTION IS USED FOR MANDATORY CARDS          00005940
C                                                                       00005950
C               ICARD(1) - ELEMENT1 CARD PROCESSED                      00005960
C               ICARD(2) - ELEMENT2 CARD PROCESSED                      00005970
C               ICARD(3) - EPOCH CARD PROCESSED                         00005980
C               ICARD(4) - OUTPUT CARD PROCESSED                        00005990
C               ICARD(5) - ORBTYPE CARD PROCESSED                       00006000
C               ICARD(6) - OBSINPUT CARD PROCESSED                      00006010
C               ICARD(7) - OBSNUME CARD PROCESSED                       00006020
C               ICARD(8) - TYPE CARD PROCESSED                          00006030
C                                                                       00006040
      DO 30 I=1,8                                                       00006050
   30 ICARD(I) = 0                                                      00006060
C                                                                       00006070
C     SET TOF TO ZERO FOR ALL SECTIONS                                  00006080
C                                                                       00006090
      DO 36 I=1,10                                                      00006100
   36 SECIN(1,I) = 0.D0                                                 00006110
      IF(IND(95) .NE. 3)   GO TO 37                                     00006120
      ICARD(1) = 1                                                      00006130
      ICARD(2) = 1                                                      00006140
      ICARD(3) = 1                                                      00006150
      IF ( INDNEX.EQ.2 ) GO TO 37                                       00006160
C                                                                       00006170
      YMDOUT = YMDIC                                                    00006180
      HMSOUT = HMSIC                                                    00006190
C     TEST FOR SATELLITE NUMBER                                         00006200
   37 IF(IDSAT.NE.0) GO TO 39                                           00006210
      IF(A8(2).EQ.FIELD(9)) INDSW=1                                     00006220
      IF(A8(2).EQ.FIELD(9)) GO TO 39                                    00006230
      ASSIGN 39 TO M                                                    00006240
      IERR=421                                                          00006250
      GO TO 845                                                         00006260
C                                                                       00006270
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

C     CHECK KEYWORD ON CONTROL CARD                                      00006280
C                                                                        00006290
   39 DO 40 I=1,10                                                       00006300
      IF (A8(2).NE.FIELD(I)) GO TO 40                                    00006310
      ITEM=I                                                             00006320
      HEADER(4,1) = FIELD(I)                                             00006330
      IND(48) = I                                                        00006340
      INDI48 = I                                                         00006350
      GO TO 60                                                           00006360
   40 CONTINUE                                                           00006370
C                                                                        00006380
C     ERROR EXIT FOR NO CONTROL CARD                                     00006390
C                                                                        00006400
   50 ASSIGN 790 TO M                                                    00006410
      IERR=411                                                           00006420
      WRITE(NOUT,1840) A8,IDSAT                                          00006430
      GO TO 845                                                          00006440
   60 IF (IPRINT.NE.1) GO TO 65                                          00006450
      CALL UPPAGE(0,0,DUM)                                               00006460
      CALL UPPAGE(2,0,DUM)                                               00006470
      WRITE (NOUT,1840)A8,IDSAT                                          00006480
   65 GO TO (80,85,740,660,660,660,740,85,80,75), ITEM                   00006490
C                                                                        00006500
C**** BRANCH FOR THE 'FILTER' PROGRAM                                    00006510
C                                                                        00006520
   75 IEPN = 0                                                           00006530
C                                                                        00006540
C*****BRANCH FOR THE 'DC' PROGRAM*************************************** 00006550
C                                                                        00006560
   80 IND(40) = 1                                                        00006570
C                                                                        00006580
C*****BRANCH FOR THE 'EPHEM' AND 'EARLYORB' PROGRAMS********************* 00006590
C                                                                        00006600
   85 ITIME = 0                                                          00006610
   90 CALL RDCARD(A1,I1,I2,I3,R1,R2,R3,&810,&800)                        00006620
      NERCRD=NERCRD+1                                                    00006630
      IF (IPRINT.EQ.1) CALL MSGWTR (2,DUM)                               00006640
C                                                                        00006650
C     IDENTIFY KEYWORD ON FIRST CARD IN DECK                             00006660
C                                                                        00006670
      DO 100 I=1,9                                                       00006680
      IF(A1 .NE. CDSMAN(I)) GO TO 100                                    00006690
      ITEM = I                                                           00006700
      ICARD(I) = 1                                                       00006710
      GO TO 120                                                          00006720
  100 CONTINUE                                                           00006730
      DO 110 I=1,8                                                       00006740
      IF (A1.NE.TERMCD(I)) GO TO 110                                     00006750
      ITEM=I                                                             00006760
      GO TO 540                                                          00006770
  110 CONTINUE                                                           00006780
C                                                                        00006790
C     ERROR EXIT FOR INCORRECT KEYWORD ON A DATA CARD                    00006800
C                                                                        00006810
  111 IF (IPRINT.NE.1) CALL MSGWTR (2,DUM)                               00006820
      CALL MSGWTR (3,DUM)                                                00006830
      IERROR=IERROR+1                                                    00006840
```

```
      GO TO 90                                                          00006850
C                                                                       00006860
C     BRANCH TO PROCESS APPROPRIATE MANDATORY CARD                      00006870
C                                                                       00006880
  120 GO TO (140,180,220,480,270,240,310,330,340),ITEM                  00006890
C                                                                       00006900
C*****MANDATORY CARD WITH KEYWORD 'ELEMENT1'*****************************00006910
C                                                                       00006920
  140 IF(IND(95).EQ.3) GO TO 90                                         00006930
      IF((I1.LT.1).OR.(I1.GT.5))  GO TO 170                             00006940
      ICORD=I1                                                          00006950
      ICTYPE=I2                                                         00006960
      JCTYPE = ICTYPE                                                   00006970
      IF ((I3.LT.1).OR.(I3.GT.11)) GO TO 170                            00006980
      IBODY=I3                                                          00006990
C                                                                       00007000
      GO TO (145,150,160,165,167,150,150), ICTYPE                       00007010
      GO TO 170                                                         00007020
C                                                                       00007030
C     CARTESIAN COORDINATE SYSTEM                                       00007040
C                                                                       00007050
  145 POSI(1)=R1                                                        00007060
      POSI(2)=R2                                                        00007070
      POSI(3)=R3                                                        00007080
      IF (ICARD(2).EQ.0) GO TO 90                                       00007090
      VELI(1)=R4                                                        00007100
      VELI(2)=R5                                                        00007110
      VELI(3)=R6                                                        00007120
      GO TO 90                                                          00007130
C                                                                       00007140
C     KEPLERIAN COORDINATE SYSTEM                                       00007150
C                                                                       00007160
  150 ELEMS(1)=R1                                                       00007170
      ELEMS(2)=R2                                                       00007180
      ELEMS(3)=R3*DTR                                                   00007190
      IF (ICARD(2).EQ.0) GO TO 90                                       00007200
      ELEMS(4)=R4*DTR                                                   00007210
      ELEMS(5)=R5*DTR                                                   00007220
      ELEMS(6)=R6*DTR                                                   00007230
      GO TO 90                                                          00007240
C                                                                       00007250
C     SPHERICAL COORDINATE SYSTEM                                       00007260
C                                                                       00007270
  160 SPHCOR(1)=R1*DTR                                                  00007280
      SPHCOR(2)=R2*DTR                                                  00007290
      SPHCOR(3)=R3*DTR                                                  00007300
      IF (ICARD(2).EQ.0) GO TO 90                                       00007310
      SPHCOR(4)=R4*DTR                                                  00007320
      SPHCOR(5)=R5                                                      00007330
      SPHCOR(6)=R6                                                      00007340
      GO TO 90                                                          00007350
C                                                                       00007360
C     INITIAL BROUWER-MEAN ELEMENTS                                     00007370
C                                                                       00007380
  165 BMELS(1)=R1                                                       00007390
      BMELS(2)=R2                                                       00007400
      BMELS(3)=R3*DTR                                                   00007410
```

```
      IF(ICARD(2).EQ.0)GO TO 90                                          00007420
      BMELS(4)=R4*DTR                                                    00007430
      BMELS(5)=R5*DTR                                                    00007440
      BMELS(6)=R6*DTR                                                    00007450
      GO TO 90                                                           00007460
C                                                                        00007470
C     DODS FLIGHT PARAMETERS COORDINATE SYSTEM                           00007480
C     CALCULATION OF DODS PARAMETERS 1, 2 AND 3 PERFORMED LATER          00007490
C                                                                        00007500
  167 SPHCOR(1) = R1                                                     00007510
      SPHCOR(2) = R2                                                     00007520
      SPHCOR(3) = R3                                                     00007530
      IF (ICARD(2).EQ.0) GO TO 90                                        00007540
      SPHCOR(4) = R4*DTR                                                 00007550
      SPHCOR(5) = R5                                                     00007560
      SPHCOR(6) = R6                                                     00007570
      GO TO 90                                                           00007580
C                                                                        00007590
C     DATA CARD ERROR EXIT                                               00007600
C                                                                        00007610
  170 ASSIGN 90 TO M                                                     00007620
      IERR=401                                                           00007630
      GO TO 840                                                          00007640
C                                                                        00007650
C*****MANDATORY CARD WITH KEYWORD 'ELEMENT2'*****************************00007660
C                                                                        00007670
  180 IF(IND(95).EQ.3) GO TO 90                                          00007680
      IF (ICARD(1).EQ.0) GO TO 210                                       00007690
      GO TO (185,190,200,205,207,190,190),ICTYPE                         00007700
C                                                                        00007710
C     CARTESIAN COORDINATE SYSTEM                                        00007720
C                                                                        00007730
  185 VELI(1)=R1                                                         00007740
      VELI(2)=R2                                                         00007750
      VELI(3)=R3                                                         00007760
      GO TO 90                                                           00007770
C                                                                        00007780
C     KEPLERIAN COORDINATE SYSTEM                                        00007790
C                                                                        00007800
  190 ELEMS(4)=R1*DTR                                                    00007810
      ELEMS(5)=R2*DTR                                                    00007820
      ELEMS(6)=R3*DTR                                                    00007830
      GO TO 90                                                           00007840
C                                                                        00007850
C     SPHERICAL COORDINATE SYSTEM                                        00007860
C                                                                        00007870
  200 SPHCOR(4)=R1*DTR                                                   00007880
      SPHCOR(5)=R2                                                       00007890
      SPHCOR(6)=R3                                                       00007900
      GO TO 90                                                           00007910
C                                                                        00007920
C     INITIAL BROUWER-MEAN ELEMENTS                                      00007930
C                                                                        00007940
  205 BMELS(4)=R1*DTR                                                    00007950
      BMELS(5)=R2*DTR                                                    00007960
      BMELS(6)=R3*DTR                                                    00007970
      GO TO 90                                                           00007980
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

C                                                                       00007990
C     DODS FLIGHT PARAMETERS                                            00008000
C                                                                       00008010
  207 SPHCOR(4) = R1*DTR                                                00008020
      SPHCOR(5) = R2                                                    00008030
      SPHCOR(6) = R3                                                    00008040
      GO TO 90                                                          00008050
C                                                                       00008060
C     SAVE THE VELOCITY ELEMENTS                                        00008070
C                                                                       00008080
  210 R4=R1                                                             00008090
      R5=R2                                                             00008100
      R6=R3                                                             00008110
      GO TO 90                                                          00008120
C                                                                       00008130
C*****MANDATORY CARD WITH KEYWORD 'EPOCH    '**************************00008140
C                                                                       00008150
  220 IF(IND(95).EQ.3.AND.R3.EQ.0.D0) GO TO 90                          00008160
      IF(R1.LE.0.D0.AND.R3.EQ.0.D0) GO TO 228                           00008170
          IF(R3.EQ.0.D0) GO TO 219                                      00008180
          YMDNEW = DFLOAT(IDINT( R3 / 1.D6 ))                           00008190
      HMSNEW = R3 - YMDNEW * 1.D6                                       00008200
C   OBTAIN JULIAN DATE OF (INPUT EPOCH ADVANCE) EPOCH.                  00008210
      CALL DELTIM(1,YMDNEW,HMSNEW,TTIME,REFJUL,DT)                      00008220
      CALL DELTIM(1,R1,R2,TOJUL,REFJUL,DT)                              00008230
      DT = 2.D0*TTIME - TOJUL                                           00008240
C   OBTAIN NEW EPOCH ADVANCE VARIABLE.                                  00008250
      CALL CDATE(DT,Y, XM,D,HR,TM,SEC,IERR)                             00008260
      TTIME = SEC +TM*1.D02 + HR*1.D04 + D*1.D06 +XM*1.D08 + Y*1.D10    00008270
      IND(88) = 1                                                       00008280
      YMDSUB = YMDNEW                                                   00008290
      HMSSUB = HMSNEW                                                   00008300
      IF(IND(95).EQ.3) GO TO 217                                        00008310
      IF(R1.NE.0.D0) GO TO 218                                          00008320
      ICARD(3) = 0                                                      00008330
      GO TO 217                                                         00008340
  219 YMDSUB = R1                                                       00008350
      HMSSUB = R2                                                       00008360
  218 YMDIC = R1                                                        00008370
      HMSIC = R2                                                        00008380
      SSYMD=R1                                                          00008390
      SSHMS=R2                                                          00008400
      IR1 = YMDIC                                                       00008410
      IA1REF = IR1 / 10000                                              00008420
      YMDREF = YMDIC                                                    00008430
      CALL DELTIM(1,YMDREF,0.D0,REFJUL,REFJUL,DT)                       00008440
  217 CALL DELTIM(1,YMDSUB,HMSSUB,REFDA4,DDQZ,DDQW)                     00008450
      IF(IND(48).NE.2) GO TO 221                                        00008460
      YMDOUT = R1                                                       00008470
      HMSOUT = R2                                                       00008480
      GO TO 90                                                          00008490
  221 IF(ITIME.NE.0.AND.ITIME.NE.2) GO TO 222                           00008500
      STIME(1) = YMDSUB                                                 00008510
      STIME(2) = HMSSUB                                                 00008520
  222 TR1 = IDINT(YMDSUB / TN4) * TN4                                   00008530
      TR2 = (YMDSUB - TR1) +300.D0                                      00008540
      IF(TR2 .LT. 1300.D0) GO TO 225                                    00008550
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

      TR2 = TR2 - 1200.D0                                                    00008560
      TR1 = TR1 + TN4                                                        00008570
C     SET END TIME DEFAULT AS EPOCH + 3 MONTHS                               00008580
  225 IF(ITIME.GT.1) GO TO 90                                                00008590
      ETIME(1) = TR1 + TR2                                                   00008600
      ETIME(2) = HMSSUB                                                      00008610
C     SET END OF SECTION TIME DEFAULT TO ETIME + 1 SECOND                    00008620
      YMDFN = ETIME(1)                                                       00008630
      HMSFN=ETIME(2)                                                         00008640
      GO TO 90                                                               00008650
  228 IF(IND(48).NE.8) GO TO 170                                             00008660
      YMDIC=0.0D0                                                            00008670
      HMSIC=0.0D0                                                            00008680
      YMDREF=0.0D0                                                           00008690
      IA1REF=0                                                               00008700
      GO TO 90                                                               00008710
C                                                                            00008720
C*****MANDATORY CARD WITH KEYWORD 'OBSINPUT'*****************************00008730
C                                                                            00008740
  240 DO 244 I=1,3                                                           00008750
      IF(IA(I) .EQ. 0) GO TO 244                                             00008760
      IF(IA(I).GE.200) GO TO 259                                             00008770
      IF ( ( IA( I ) .LT. 1 ) .OR. ( IA( I ) .GT. 14 ) )  GO TO 170          00008780
      ITM = IA(I)                                                            00008790
      GO TO (247,248,249,250,251,    252,253,254,255,256,                    00008800
     *       257,258,260,261), ITM                                           00008810
C     SET GTDS TAPE FILE SWITCH                                              00008820
  247 IND(52) = 1                                                            00008830
      GO TO 244                                                              00008840
C     SET GTDS DISK FILE SWITCH                                              00008850
  248 IND(54) = 1                                                            00008860
      GO TO 244                                                              00008870
C     SET DODS TAPE FILE SWITCH                                              00008880
  249 IND(53) = 1                                                            00008890
      GO TO 244                                                              00008900
C     SET DODS DISK FILE SWITCH                                              00008910
  250 IND(55) = 1                                                            00008920
      GO TO 244                                                              00008930
C     SET GTDS CARD FILE SWITCH                                              00008940
  251 IND(58) = 1                                                            00008950
      GO TO 244                                                              00008960
C     SET CDS PREPROCESSED OBSERVATIONS FILE                                 00008970
  252 IND(56) = 1                                                            00008980
      GO TO 244                                                              00008990
C      SET OPTICAL ASPECT 9-TRACK TAPE                                       00009000
  253 IND(57) = 1                                                            00009010
      GO TO 244                                                              00009020
C     SET INTERIM OBSERVATIONS DATA BASE SWITCH                              00009030
  254 NDATAB = 1                                                             00009040
      GO TO 244                                                              00009050
C     SET PCE ORB1 TAPE SWITCH                                               00009060
  255 IND(49) = 1                                                            00009070
      CARTES=.TRUE.                                                          00009080
      GO TO 244                                                              00009090
C     SET LANDMARK DATA CARD INPUT INDICATOR                                 00009100
  256 IND(47) = 1                                                            00009110
      GO TO 244                                                              00009120
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

C GENERAL DATA HANDLER FORMAT                                          00009130
  258 IND(34)=1                                                        00009140
      GO TO 244                                                        00009150
C     SET SATELLITE-TO-SATELLITE SOURCE INDICATOR                      00009160
  259 IJK1=IA(I)                                                       00009170
      GO TO 244                                                        00009180
C     SET EARTH-EDGE DATA SOURCE INDICATOR                             00009190
  260 IND(219)=1                                                       00009200
      GO TO 244                                                        00009210
C             SET SWITCH TO INDICATE THAT A GPS DATA TAPE IS THE       00009220
C             SOURCE OF THE DATA                                       00009230
C                                                                      00009240
  261 IND( 223 ) = 1                                                   00009250
  244 CONTINUE                                                         00009260
      IF((IA(1)+IA(2)+IA(3)).EQ.0)GO TO 170                            00009270
      IND(51) = 1                                                      00009280
      GO TO 241                                                        00009290
C     SET SWITCH TO USE WORKING FILE CREATED DURING PREVIOUS RUN       00009300
C     - BYPASS OBSNF                                                   00009310
  257 IND(51)=2                                                        00009320
  241 IF(R1.EQ.0) GO TO 242                                            00009330
      ITM = R1 / 1.D06                                                 00009340
      STIME(1) = ITM                                                   00009350
      STIME(2) = R1 - STIME(1) * 1.D06                                 00009360
      ITIME = 1                                                        00009370
  242 IF(R2.EQ.0) GO TO 243                                            00009380
      ITM = R2 / 1.D06                                                 00009390
      ETIME(1) = ITM                                                   00009400
      ETIME(2) = R2 - ETIME(1) * 1.D06                                 00009410
      YMDFN = ETIME(1)                                                 00009420
      HMSFN = ETIME(2) + 1.D0                                          00009430
      ITIME = ITIME + 2                                                00009440
C     SET FINAL TIME LARGE FOR DC CASE                                 00009450
      YMDFN=991231.D0                                                  00009460
      GO TO 2430                                                       00009470
  243 ETIME(1)=YMDFN                                                   00009480
      ETIME(2)=HMSFN                                                   00009490
 2430 IF(R3 .EQ. 0) GO TO 90                                           00009500
      IDSAT2=R3                                                        00009510
      GO TO 90                                                         00009520
C                                                                      00009530
C*****MANDATORY CARD WITH KEYWORD 'ORBTYPE''***************************00009540
C                                                                      00009550
C*****STORE 11 FOR ALL TYPES                                           00009560
 270  IF(I1.LT.0) GO TO 170                                            00009570
      ICRTYP=1                                                         00009580
      IF(I1.EQ.0) I1=2                                                 00009590
      IND(1)=I1                                                        00009600
      INTTYP(2) = I1                                                   00009610
C                                                                      00009620
C   ORBTYPE OPTIONS IMPLEMENTED ARE:                                   00009630
C      TYPES 1,2,3,4,5,6,9,10,11,12,13,14,15                           00009640
C                                                                      00009650
C     BRANCH TO APPROPRIATE CODE FOR ORBIT GENERATOR TYPE.             00009660
C                                                                      00009670
      GO TO (281,281,282,282,281,290,170,170,281,281,                  00009680
     *    282,282,282,282,282), I1                                     00009690
```

```
      GO TO 170                                                          00009700
C                                                                        00009710
C     FOLLOWING FOR BROUWER ONLY                                         00009720
C                                                                        00009730
  282 CONTINUE                                                           00009740
      IDSECL = I2                                                        00009750
      IDOSM = I3                                                         00009760
      DOTA = R1                                                          00009770
C                                                                        00009780
C     FOLLOWING FOR VINTI ONLY                                           00009790
C                                                                        00009800
      IF (I2 .EQ. 0) I2 = 2                                              00009810
      IRES4 = I2                                                         00009820
      IF (I3 .EQ. 0) I3 = 2                                              00009830
      IMNRTE = I3                                                        00009840
      GO TO 90                                                           00009850
C                                                                        00009860
C     NUMERICAL ORBIT GENERATORS.                                        00009870
C                                                                        00009880
 281  IF(I2.LT.0.OR.I2.GT.4.OR.I3.LT.0.OR.I3.GT.2)GO TO 170              00009890
C*****STORE I2 FOR TYPES 1,2,5,7,9                                       00009900
      IF(I2.EQ.0)I2=1                                                    00009910
      DO 271 I=1,10                                                      00009920
 271  INDSEC(5,I)=I2                                                     00009930
C*****STORE I3 FOR TYPES 1,2,5,7,9                                       00009940
      IF(I3.EQ.0)I3=1                                                    00009950
      I50=I3                                                             00009960
      IF(NWSLP.EQ.NSWSLP) GO TO 272                                      00009970
      NWSLP=NFSLP                                                        00009980
      IF(I50 .EQ.2)NWSLP=NTOD                                            00009990
C*****STORE R1 FOR TYPES 1,2,5,7,9                                       00010000
 272  IF(R1.LT.0.D0)GO TO 170                                            00010010
      TMP=STPSZE                                                         00010020
      IF(I1.EQ.1)TMP=200.                                                00010030
      IF(IND(101).EQ.2.AND.R3.EQ.2.D0) TMP=7200.                         00010040
      IF(I1.EQ.9)TMP=5400.                                               00010050
      IF (R1.GT.0.D0) TMP = R1                                           00010060
      NM=31                                                              00010070
      IF(I1.EQ.1)NM=35                                                   00010080
      DO 273 I=1,10                                                      00010090
  273 SECMOD(NM,I)=0.D0                                                  00010100
      SECMOD(NM,1) = TMP                                                 00010110
      STPMIN = 5.D0                                                      00010120
      GO TO (276,90,170,170,278,170,170,170,275,291), I1                 00010130
C                                                                        00010140
C*****STORE R2 FOR TYPE 9                                                00010150
C                                                                        00010160
C     THIS PORTION OF SETRUN HAS BEEN MODIFIED TO ACCEPT ADDITIONAL DATA 00010170
C     FROM FIELD R2.  SPECIFICALLY, FIELD R2 NOW SHOULD BE ENTERED AS    00010180
C     FOLLOWS:  R2 = MM.NNII  WHERE:                                     00010190
C                                                                        00010200
C     MM = MORD  = ORDER OF CHEBYSHEV APPROXIMATING SERIES (<68).        00010210
C     NN = NSUS  = NUMBER-1 OF CHEBYSHEV ORTHOGONALITY POINTS (<70).     00010220
C     II = ITERM = TERMINAL INTERATION NUMBER FOR ITERATIVE INTEGRATOR.   00010230
C                                                                        00010240
C     THE FOLLOWING CODE WILL DECOMPOSE THE FLOATING POINT NUMBER INTO   00010250
C     THE DESCRIBED INTEGER VALUES FOR USE BY THE PICARD-CHEBYSHEV       00010260
```

```
C     INTEGRATOR.                                                          00010270
C                                                                          00010280
  275 CONTINUE                                                             00010290
      IEQSET = 1                                                           00010300
      IF (R2.NE.0.D0)                    GO TO 2701                        00010310
      MORD   = 36                                                          00010320
      NSUB   = 49                                                          00010330
      ITERM  = 99                                                          00010340
      GO TO 2703                                                           00010350
 2701 MORD   = R2 + 1.D-06                                                 00010360
      R2SUB1 = (R2 - DFLOAT(MORD)) * 1.D02                                 00010370
      NSUB   = R2SUB1 + 1.D-04                                             00010380
      R2SUB2 = (R2SUB1 - DFLOAT(NSUB)) * 1.D02                             00010390
      ITERM  = R2SUB2 + 1.D-02                                             00010400
      IF (MORD.GT.4.AND.MORD.LT.68)      GO TO 2703                        00010410
      MORD   = 67                                                          00010420
      NSUB   = 69                                                          00010430
      ITERM  = 99                                                          00010440
      ASSIGN 2703 TO M                                                     00010450
      IERR=461                                                             00010460
      GO TO 846                                                            00010470
C*****STORE R3 FOR TYPE 9                                                  00010480
 2703 IF (R3.EQ.0.D0)                    R3 = 1.D-10                       00010490
      DO  2704      I = 1,6                                                00010500
      CON(I) = R3                                                          00010510
 2704 CONTINUE                                                             00010520
      CONKEP = R3                                                          00010530
      DPINT(149) = R3                                                      00010540
      GO TO 90                                                             00010550
C                                                                          00010560
C     *  END OF MODIFICATIONS FOR PICARD CHEBYSHEV INTEGRATOR  ********    00010570
C                                                                          00010580
C*****STORE R3 FOR TYPE 1                                                  00010590
 276  IF(R3.LT.0.D0)GO TO 170                                              00010600
      IF(R3.EQ.0.D0)R3=1.5D0                                               00010610
      DO 277 I=2,10                                                        00010620
 277  SECMOD(36,I)=0.                                                      00010630
      SECMOD(36,1)=R3                                                      00010640
      GO TO 90                                                             00010650
C                                                                          00010660
C**** STORE R2 AND R3 FOR VARIATION OF PARAMETERS TYPE ***************     00010670
C                                                                          00010680
C     THIS PORTION OF SETRUN HAS BEEN MODIFIED TO ACCEPT ADDITIONAL DATA   00010690
C     FROM THE R2 AND R3 FIELDS OF THE 'ORBTYPE' CARD FOR VOP ORBIT        00010700
C     GENERATORS.  THIS IS PART OF THE EFFORT TO INCORPORATE THE PICARD-    00010710
C     CHEBYSHEV INTEGRATOR INTO THE VOP ORBIT GENERATOR OPTION.            00010720
C     1.  THE R2 FIELD, WHICH WAS PREVIOUSLY BLANK, NOW HAS THE FOLLOW-    00010730
C     ING MEANINGS:                                                        00010740
C         A.  R2 = BLANK IMPLIES THAT PICARD-CHEBYSHEV IS NOT TO BE USED   00010750
C         AND THE APPROPRIATE FLAG IN COMMON/SWITCH/ ICHEBY = IND(20) IS   00010760
C         SET TO 2.                                                        00010770
C         B.  R2 = 1.JJ IMPLIES THAT PICARD-CHEBYSHEV IS TO BE USED.       00010780
C         ICHEBY IS SET TO 1 AND THE DECIMAL PART BECOMES THE POWER OF     00010790
C         TEN IN THE CONVERGENCE CRITERIA, I.E. 1.D-JJ.                    00010800
C                                                                          00010810
C     2.  THE R3 FIELD, WHICH PREVIOUSLY CONTAINED ONLY THE INDICATOR      00010820
C     FOR THE TYPE OF VOP PARAMETER, NOW CONTAINS THE FOLLOWING DATA:      00010830
```

```
C     R3 = JJ.MMNNII  WHERE:                                                  00010840
C         JJ = IND(3) = TYPE OF VOP PARAMETERS                                00010850
C         MM = MORD   = ORDER OF CHEBYSHEV APPROXIMATING SERIES.              00010860
C         NN = NSUB   = MAX SUBSCRIPT OF CHEBYSHEV ORTHOGONALITY POINTS.      00010870
C         II = ITERM  = TERMINAL ITERATION NUMBER.                            00010880
C                                                                             00010890
C     THE FOLLOWING CODE WILL DECOMPOSE THESE FLOATING POINT NUMBERS          00010900
C     INTO THE DESCRIBED VALUES FOR USE BY THE CHEBYSHEV INTEGRATOR.          00010910
C                                                                             00010920
  278 IND(20) = 2                                                             00010930
      IF (R2.LT.1.D0)                    GO TO 2782                           00010940
      IND(20) = 1                                                             00010950
      JPWR = (R2 + 1.D-10 -1.D0) * 1.D02                                      00010960
      DO  2781 I = 1,6                                                        00010970
 2781 CON(I) = 10.D0 ** (-JPWR)                                               00010980
      CONKEP = CON(1)                                                         00010990
      DPINT(149) = CONKEP                                                     00011000
C                                                                             00011010
 2782 IF (R3.LT.1.D0.OR.R3.GE.21.D0)     GO TO 170                          00011020
      IND(3)  = R3 + 1.D-08                                                   00011030
      IF (IND(3).EQ.0)                   IND(3) = 12                          00011040
      IF (IND(20).EQ.2)                  GO TO 2783                           00011050
      R3SUB1  = (R3 - DFLOAT(IND(3))) * 1.D02                                 00011060
      MORD    = R3SUB1 + 1.D-06                                               00011070
      R3SUB2  = (R3SUB1 - DFLOAT(MORD)) * 1.D02                               00011080
      NSUB    = R3SUB2 + 1.D-04                                               00011090
      R3SUB3  = (R3SUB2 - DFLOAT(NSUB)) * 1.D02                               00011100
      ITERM   = R3SUB3 + 1.D-02                                               00011110
C                                                                             00011120
      IF (IND(3).EQ.2 )                  IEQSET = 3                           00011130
      IF (IND(3).EQ.3 )                  IEQSET = 4                           00011140
      IF (IND(3).EQ.8 )                  IEQSET = 2                           00011150
      IF (IND(3).EQ.12)                  IEQSET = 5                           00011160
      IF (IND(3).EQ.18)                  IEQSET = 6                           00011170
 2783 IF (IND(3).LT.11)                  GO TO 90                             00011180
      CALL SETAVR('        ',IA,RA,&170,&170)                                 00011190
      GO TO 90                                                                00011200
C                                                                             00011210
C     * END OF MODIFICATIONS FOR PICARD-CHEBYSHEV INTEGRATOR  **********      00011220
C                                                                             00011230
C*****STORE I2,I3 FOR TYPE 6                                                  00011240
C                                                                             00011250
 290  IND(81)=I2                                                              00011260
      IND(82)=I3                                                              00011270
      IND(6)=2                                                                00011280
      GO TO 90                                                                00011290
C***** SET IND(2) FOR TYPE 10                                                 00011300
  291 IND(2) = 5                                                              00011310
      GO TO 90                                                                00011320
C                                                                             00011330
C*****MANDATORY CARD WITH KEYWORD 'OBSNUME '*****************************     00011340
C                                                                             00011350
  310 DO 315 I = 1,3                                                          00011360
      IF(RA(I).LE.0.D0)   GO TO 315                                           00011370
      IM = IM + 1                                                             00011380
      TSTART(IM) = RA(I) - .005D0                                             00011390
      TEND(IM) = RA(I) + .005D0                                               00011400
```

```
      LEVSTR(2) = 1                                                    00011410
      LEVEND(2) = IM                                                   00011420
      METYPE(IM,1) = I1                                                00011430
      METYPE(IM,2) = I2                                                00011440
      METYPE(IM,3) = I3                                                00011450
      IF(I2.EQ.0) METYPE(IM,2)=99                                      00011460
      IF(I3.EQ.0) METYPE(IM,3)=99                                      00011470
  315 CONTINUE                                                         00011480
      GO TO 90                                                         00011490
C                                                                      00011500
C*****MANDATORY CARD WITH KEYWORD 'TYPE    '*************************00011510
C                                                                      00011520
  330 IF(IND(48).NE.8) GO TO 535                                       00011530
      IF((I1.LT.1).OR.(I1.GT.5))GO TO 170                              00011540
      IND(5) = I1                                                      00011550
      IF((I2.LT.-1).OR.(I2.GT.1))GO TO 170                             00011560
      INZZ=I2                                                          00011570
      IF(I3.LT.1 .OR. I3.GT.11)  GO TO 170                             00011580
      IBODY = I3                                                       00011590
C     STORE SEMI-MAJOR AXIS AND INCLINATION                            00011600
      ELEMS(1) = R1                                                    00011610
      IF(I3.EQ.4) GO TO 90                                             00011620
      ELEMS(3) = R2                                                    00011630
      ELEMS(6) = R3                                                    00011640
      GO TO 90                                                         00011650
C                                                                      00011660
C*****MANDATORY CARD WITH KEYWORD 'LTYPE   '*************************00011670
C                                                                      00011680
  340 IF(I1.LT.1 .OR. I1.GT.2) GO TO 535                               00011690
      IND(192) = I1                                                    00011700
      IF(I2.LT.1 .OR. I2.GT.2) GO TO 535                               00011710
      IND(193) = I2                                                    00011720
      IF(I3.LT.1 .OR. I3.GT.2) GO TO 535                               00011730
      IND(194) = I3                                                    00011740
      IF(R1.LT.0.D0 .OR. R1.GT.54320.D0) GO TO 535                     00011750
      ELEMS(2) = R1                                                    00011760
      ELEMS(4) = R2                                                    00011770
      ELEMS(5) = R3                                                    00011780
      GO TO 90                                                         00011790
C                                                                      00011800
C*****MANDATORY CARD WITH KEYWORD 'OUTPUT  '*************************00011810
C                                                                      00011820
  480 IF(IND(48).NE.2) GO TO 535                                       00011830
      IF((I1.LT.0).OR.(I1.GT.5)) GO TO 170                             00011840
      IF (I1.EQ.0) I1=2                                                00011850
      DO 490 I=1,10                                                    00011860
  490 INDSEC(26,I)=I1                                                  00011870
      YMDFN=R1                                                         00011880
      HMSFN=R2                                                         00011890
      DELTAT=R3                                                        00011900
      ISEEHA = .FALSE.                                                 00011910
      IF((I2.GE.91).AND.(I2.LE.93))  GO TO 482                         00011920
      GO TO 484                                                        00011930
  482 ISEEHA= .TRUE.                                                   00011940
      I2 = I2 - 90                                                     00011950
  484 CONTINUE                                                         00011960
      IF(I2.LT.10) GO TO 489                                           00011970
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

      IASDES= I2/10                                                          00011980
      I2=I2-IASDES*10                                                        00011990
      DELTAT=1.0D20                                                          00012000
      NFREQ=1                                                                00012010
      IF(R3.NE.0.D0) NFREQ=R3                                                00012020
C                                                                            00012030
C     INSECT(322) =1....IS AN ASCENDING NODAL CROSSING                       00012040
C     INSECT(322) =2....IS A DESCENDING NODAL CROSSING                       00012050
C     INSECT(322) =3....IS BOTH TYPES OF CROSSINGS                           00012060
C     RESET I2 FOR NORMAL PROCESSING AND CONTINUE                            00012070
C                                                                            00012080
489   CONTINUE                                                               00012090
      IF(I2 .EQ. 0) I2 = 2                                                   00012100
      IF(I2.LT.1.OR.I2.GT.3) GO TO 170                                       00012110
      DO 491 I=1,10                                                          00012120
      INDSEC(25,I) = I2                                                      00012130
  491 CONTINUE                                                               00012140
      IF ((I3.LT.0).OR.(I3.GT.15))  GO TO 170                                00012150
      NUM = 2                                                                00012160
      IF  (I3.EQ.15)   NUM = 1                                               00012170
      DO 492 I = 1,10                                                        00012180
      INDSEC(21,I) = NUM                                                     00012190
      INDSEC(22,I) = NUM                                                     00012200
      INDSEC(23,I) = NUM                                                     00012210
  492 INDSEC(24,I) = NUM                                                     00012220
      IF ((I3.EQ.0).OR.(I3.EQ.15))   GO TO 90                                00012230
C     TEST FOR GEOCENTRIC OUTPUT                                             00012240
      IF ((I3.EQ.1).OR.(I3.EQ.3).OR.(I3.EQ.5).OR.(I3.EQ.7).OR.(I3.EQ.9) 00012250
     * .OR.(I3.EQ.11).OR.(I3.EQ.13))  GO TO 496                              00012260
C     TEST FOR HELIOCENTRIC OUTPUT                                           00012270
  493 IF ((I3.EQ.2).OR.(I3.EQ.3).OR.(I3.EQ.6).OR.(I3.EQ.7).OR.(I3.EQ.10)00012280
     * .OR.(I3.EQ.11).OR.(I3.EQ.14))  GO TO 497                              00012290
C     TEST FOR SELENECENTRIC OUTPUT                                          00012300
  494 IF ((I3.EQ.4).OR.(I3.EQ.5).OR.(I3.EQ.6).OR.(I3.EQ.7).OR.(I3.EQ.12)00012310
     * .OR.(I3.EQ.13).OR.(I3.EQ.14))  GO TO 498                              00012320
C     TEST FOR TARGET BODY OUTPUT                                            00012330
  495 IF ( I3.LT.8 )  GO TO 90                                               00012340
      NUM = 24                                                               00012350
      NTP = 4                                                                00012360
      GO TO 485                                                              00012370
  496 NUM = 22                                                               00012380
      NTP = 1                                                                00012390
      GO TO 485                                                              00012400
  497 NUM = 23                                                               00012410
      NTP = 2                                                                00012420
      GO TO 485                                                              00012430
  498 NUM = 21                                                               00012440
      NTP = 3                                                                00012450
  485 DO 499 I = 1,10                                                        00012460
  499 INDSEC(NUM,I) = 1                                                      00012470
      GO TO (493,494,495,90), NTP                                            00012480
      GO TO 90                                                               00012490
C                                                                            00012500
C     ILLEGAL MANDATORY CARD EXIT                                            00012510
C                                                                            00012520
  535 ASSIGN 90 TO M                                                         00012530
      IERR=453                                                               00012540
```

```
      IERROR=IERROR-1                                                   00012550
      GO TO 840                                                         00012560
C                                                                       00012570
C*****BRANCH FOR A TERMINAL CARD*****************************************00012580
C                                                                       00012590
  540 IF(IND(48).NE.1.AND.IND(48).NE.10)GO TO 542                       00012600
      IDCRUN = 1                                                        00012610
      ICARD(4) = ICARD(6)                                               00012620
  542 IF(IND(48).LT.3.OR.IND(48).EQ.10)GO TO 550                        00012630
      IF(IND(48).EQ.3) IDCRUN = 1                                       00012640
      IF(IND(48).EQ.9) GO TO 543                                        00012650
      IF(IND(48).NE.8) GO TO 620                                        00012660
      IDCRUN = 1                                                        00012670
      IF(IND(45).EQ.3) GO TO 780                                        00012680
      IF(ICARD(3).NE.1 .OR. ICARD(6).NE.1 .OR. ICARD(7).NE.1 .OR. ICARD(00012690
     *8) .NE.1)   GO TO 580                                             00012700
      GO TO 620                                                         00012710
C                                                                       00012720
C     BRANCH FOR TERMINAL CARD AFTER DC AND ORBIT GENERATOR             00012730
C     MANDATORY CARDS                                                   00012740
C                                                                       00012750
  550 IF(IND(48).NE.2) GO TO 555                                        00012760
      IF(DELTAT.GT.0.D0) GO TO 555                                      00012770
      IND(83) = 1                                                       00012780
      IF(DELTAT.LT.0.D0) IND(83) = 3                                    00012790
      CALL DELTIM(2,YMDFN,HMSFN,DDQZ,REFDA4,DELTAT)                     00012800
  555 IF(IND(1).EQ.6) GO TO 620                                         00012810
      DO 560 I=1,4                                                      00012820
      IF (ICARD(I).EQ.0) GO TO 570                                      00012830
  560 CONTINUE                                                          00012840
      IND(95)=1                                                         00012850
      GO TO 615                                                         00012860
  543 DO 544 MM=1,3                                                     00012870
  544 IF (FOSI(MM).EQ.0) IND(59)=1                                      00012880
      DO 545 MM=1,3                                                     00012890
  545 IF (VELI(MM).EQ.0) IND(59)=1                                      00012900
      IF(ICARD(6).NE.1) IND(55)=1                                       00012910
      GO TO 620                                                         00012920
  570 IF ((ICARD(1).EQ.0).AND.(ICARD(2).EQ.0).AND.(ICARD(3).EQ.0).AND.(I00012930
     1CARD(4).EQ.1)) GO TO 590                                          00012940
C                                                                       00012950
C     ERROR EXIT FOR MISSING MANDATORY CARDS                            00012960
C                                                                       00012970
  580 ASSIGN 90 TO M                                                    00012980
      IERR=418                                                          00012990
      GO TO 845                                                         00013000
  590 IF (A1.NE.TERMCD(1)) GO TO 580                                    00013010
      GO TO 615                                                         00013020
C                                                                      0000013030
C     BRANCH TO LOCATION DEPENDING ON THE OPTIONAL DATA DECK            00013040
C     KEYWORD OR A 'FIN' CARD                                           00013050
C                                                                       00013060
  600 DO 610 I=1,8                                                      00013070
      IF (A1.NE.TERMCD(I)) GO TO 610                                    00013080
      ITEM=I                                                            00013090
      GO TO 620                                                         00013100
  610 CONTINUE                                                          00013110
```

```
      IF (IPRINT.NE.1) CALL MSGWTR (2,DUM)                              00013120
      CALL MSGWTR (3,DUM)                                               00013130
      GO TO 660                                                         00013140
  615 CONTINUE                                                          00013150
C  DEACTIVATE THE FOLLOWING TWO LINES OF CODE RELATING TO               00013160
C    THE SATELLITE CENTRAL DIRECTORY.                                   00013170
C     IF( ICARD(5).EQ.0) IND(217) = 1                                   00013180
C     IRDSCD = 1                                                        00013190
C                                                                       00013200
  620 GO TO (630,670,777,690,650,760,680,720),ITEM                      00013210
C                                                                       00013220
C*****BRANCH FOR A 'DMOPT' OPTIONAL DATA DECK*************************00013230
C                                                                       00013240
  630 IF (IFLDM.GT.0) GO TO 635                                         00013250
      CALL SETDM (MSGERR,&775,&805)                                     00013260
      IFLDM=1                                                           00013270
      IF(IND(48).EQ.9) GO TO 660                                        00013280
      IF(INDI48.EQ.10)IND(48) = 1                                       00013290
      CALL WFCONT(&807)                                                 00013300
      WFCALL = .TRUE.                                                   00013310
      IF(IND(48).NE.8) I4EDIT(7) = 0                                    00013320
      I4EDIT(8) = 0                                                     00013330
      DO 633 I=2,6                                                      00013340
  633 I4EDIT(I) = 0                                                     00013350
      I4EDIT(1) = 2                                                     00013360
      I4EDIT(107)=0                                                     00013370
      IND(48) = INDI48                                                  00013380
      GO TO 660                                                         00013390
  635 ASSIGN 660 TO M                                                   00013400
      IERR=455                                                          00013410
      GO TO 846                                                         00013420
C                                                                       00013430
C*****BRANCH FOR THE 'COMPARE' AND 'DATAMST' PROGRAMS*****************00013440
C                                                                       00013450
  660 CALL RDCARD(A1,I1,I2,I3,R1,R2,R3,&810,&800)                       00013460
      NERCRD=NERCRD+1                                                   00013470
      IF(IPRINT.EQ.1) CALL MSGWTR(2,DUM)                                00013480
      GO TO 600                                                         00013490
C                                                                       00013500
C*****BRANCH FOR A 'COMPOPT' OPTIONAL DATA DECK************************00013510
C                                                                       00013520
  650 IF(IND(48).EQ.4) GO TO 655                                        00013530
      ASSIGN 655 TO M                                                   00013540
      IERR=435                                                          00013550
      GO TO 840                                                         00013560
  655 CALL SETCMP(MSGERR,&775,&805)                                     00013570
      GO TO 660                                                         00013580
C                                                                       00013590
C*****BRANCH FOR AN 'OGOPT' OPTIONAL DATA DECK************************00013600
C                                                                       00013610
  670 CALL SETORB (MSGERR,&775,&805)                                    00013620
      GO TO 660                                                         00013630
C                                                                       00013640
C*****BRANCH FOR  A 'FILOPT ' OPTIONAL DATA DECK                       00013650
C                                                                       00013660
  680 CALL SETFIL(MSGERR,&775,&805)                                     00013670
      IND(48) = 1                                                       00013680
```

```
      IF(.NOT.WFCALL) CALL WFCONT(&807)                                 00013690
      WFCALL = .TRUE.                                                   00013700
      IND(48) = INDI48                                                  00013710
      GO TO 660                                                         00013720
C                                                                       00013730
C*****BRANCH FOR A 'DCOPT' OPTIONAL DATA DECK*************************00013740
C                                                                       00013750
  690 IF(IND(48).EQ.9) GO TO 695                                        00013760
      IF(IDCRUN.EQ.0) GO TO 700                                         00013770
      IF(INDI48.EQ.10)IND(48) = 1                                       00013780
      IF(.NOT.WFCALL) CALL WFCONT(&807)                                 00013790
      IND(48) = INDI48                                                  00013800
      WFCALL = .TRUE.                                                   00013810
  695 CALL SETDC(MSGERR,&775,&805)                                      00013820
      GO TO 660                                                         00013830
  700 ASSIGN 660 TO M                                                   00013840
      IERR=420                                                          00013850
      GO TO 845                                                         00013860
C                                                                       00013870
C*****  BRANCH FOR AN 'EAOPT' OPTIONAL DATA DECK  *****                 00013880
C                                                                       00013890
  720 CALL SETEA( MSGERR, &775, &805 )                                  00013900
      GO TO 660                                                         00013910
C                                                                       00013920
C*****BRANCH FOR THE 'DATASIM' AND 'ANALYSIS' PROGRAMS****************00013930
C                                                                       00013940
  740 CONTINUE                                                          00013950
      IND(95) = 1                                                       00013960
      IDM = 0                                                           00013970
      IND(1) = 6                                                        00013980
  742 CALL RDCARD(A1,I1,I2,I3,R1,R2,R3,&810,&800)                       00013990
      NERCRD = NERCRD+1                                                 00014000
      IF (IPRINT .EQ. 1)  CALL MSGWTR(2,DUM)                            00014010
      IF (A1 .EQ. TERMCD(1))  GO TO 744                                 00014020
      IF (IDM .NE. 1)  GO TO 835                                        00014030
      IF(A1.EQ.TERMCD(4)) GO TO 746                                     00014040
      IF (A1 .EQ. TERMCD(2))  GO TO 748                                 00014050
      IF ( A1 .EQ. TERMCD(3) )  GO TO 777                               00014060
      IF ( A1 .EQ. TERMCD(8) )  GO TO 750                               00014070
      GO TO 790                                                         00014080
  744 IDM = 1                                                           00014090
      CALL SETDM (MSGERR,&775,&805)                                     00014100
      IF(IND(48).EQ.9) GO TO 742                                        00014110
      CALL WFCONT(&807)                                                 00014120
      GO TO 742                                                         00014130
  746 CONTINUE                                                          00014140
      CALL GETDSP                                                       00014150
      CALL DSPDUM                                                       00014160
      CALL SETDC (MSGERR,&775,&805)                                     00014170
      GO TO 742                                                         00014180
  748 CALL SETORB (MSGERR,&775,&805)                                    00014190
      GO TO 742                                                         00014200
  750 CALL DSPDUM                                                       00014210
      CALL SETEA( MSGERR, &775, &805)                                   00014220
      GO TO 742                                                         00014230
C                                                                       00014240
C*****BRANCH FOR THE 'FILERPT' PROGRAM********************************00014250
```

```
C                                                                       00014260
  760 CALL SETPFR(&775)                                                 00014270
      IDM = 1                                                           00014280
      GO TO 742                                                         00014290
C                                                                       00014300
C*****ERROR PROCESSING SECTION*************************************************00014310
C                                                                       00014320
C     CARD ERROR RETURN FROM CALLED SUBROUTINE                          00014330
C                                                                       00014340
  775 RETERR = .TRUE.                                                   00014350
      GO TO 660                                                         00014360
C                                                                       00014370
C     BRANCH FOR A 'FIN' CARD                                           00014380
C                                                                       00014390
  777 CONTINUE                                                          00014400
      IF(IND(48).GT.2) GO TO 779                                        00014410
      IF(ICRTYP.EQ.0) WRITE(6,1665)                                     00014420
  779 CONTINUE                                                          00014430
      IF(( NOISTA.LE.0).OR.(NOISTA.GT.20)) GO TO 780                    00014440
      DO 778 I = 1,NOISTA                                               00014450
      IF(SSDP(7,I).LE.0.0D0) SSDP(7,I) = FREQ(ISG(7,I))                 00014460
  778 CONTINUE                                                          00014470
  780 IF(IERROR.EQ.0.AND..NOT.RETERR) GO TO 820                         00014480
C                                                                       00014490
C     DATA CARD ERROR RETURN                                            00014500
C                                                                       00014510
      NCDERR = IERROR + NCDERR                                          00014520
C     PRINT OUT THE NUMBER OF ERRORS FOUND                              00014530
      IDUM = IERROR                                                     00014540
      CALL MSGWTR(8,CONTRL)                                             00014550
  790 RETURN 1                                                          00014560
C                                                                       00014570
C     I/O ERROR RETURN                                                  00014580
C                                                                       00014590
  800 MSGERR = 106                                                      00014600
  805 CALL ERROUT(MSGERR)                                               00014610
  807 IND(48) = INDI48                                                  00014620
      RETURN 2                                                          00014630
C                                                                       00014640
C     PREMATURE END-OF-FILE ENCOUNTERED IN THE INPUT DECK               00014650
C                                                                       00014660
  810 MSGERR = 140                                                      00014670
      CALL ERROUT(MSGERR)                                               00014680
  815 RETURN 3                                                          00014690
C                                                                       00014700
C*****NORMAL RETURN************************************************************00014710
C                                                                       00014720
C     PRINT OUT NO ERRORS FOUND                                         00014730
  820 CONTINUE                                                          00014740
      IF(WFCALL.OR.IND(217).NE.1) GO TO 1815                            00014750
      CALL WFCONT(&807)                                                 00014760
      WFCALL=.TRUE.                                                     00014770
C     SET EVAL INITIALIZATION SWITCH                                    00014780
 1815 IND(31)=1                                                         00014790
      IF (ICTYPE.NE.5) GO TO 1820                                       00014800
      ICTYPE = 3                                                        00014810
      NS(1) = 0                                                         00014820
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

      NB(2) = 0                                                          00014830
      NB(3) = 0                                                          00014840
      CALL EVAL(0.0D0,2,2,2,2,1,NB,XB)                                   00014850
      IND(31) = 1                                                        00014860
      SPHCOR(1) = SPHCOR(1)*DTR +  DMOD(GHA,TWOPI)                       00014870
      SPHCOR(1) = DMOD(SPHCOR(1),TWOPI)                                  00014880
      F = 1.0D0/FLAT(1)                                                  00014890
      SPH = DATAN((1.0D0-F)**2*DTAN(SPHCOR(2)*DTR))                      00014900
      SPL = SPHCOR(2)*DTR-SPH                                            00014910
      FF = 2.0D0*F - F**2                                                00014920
      RC = AB(1) * DSQRT((1.0D0-FF)/(1.0D0-(FF*(DCOS(SPH))**2)))         00014930
      FF = DSQRT(SPHCOR(5)**2- (RC**2)*(DSIN(SPL))**2) - RC*DCOS(SPL)    00014940
      SPHCOR(2) = SPH +DARSIN((FF/SPHCOR(5)) * DSIN(SPL))                00014950
      SPHCOR(3) = -SPHCOR(3)*DTR +  90.0D0*DTR                           00014960
 1820 CONTINUE                                                           00014970
C  DEACTIVATE THE FOLLOWING TWO LINES OF CODE RELATING TO                00014980
C    THE SATELLITE CENTRAL DIRECTORY.                                    00014990
C     IF( IND(48).LE.2.AND.ICARD(5).EQ.0.AND.IRDSCD.EQ.0) CALL CHUSRB    00015000
C                                                                        00015010
      IF(IPRINT.EQ.1) CALL MSGWTR(7,CONTRL)                              00015020
      IPT=1                                                              00015030
C  SET EVAL INITIALIZE SWITCH                                            00015040
      IND(31)=1                                                          00015050
      IF(IOK.NE.0)IPT=IOK                                                00015060
      IF(IND(45).NE.2.OR.IOK.NE.0)CALL OKERR(IPT)                        00015070
      IF(IND(45).NE.2)NINVRS=1                                           00015080
C                                                                        00015090
C     EXIT IF DATA MANAGEMENT RUN                                        00015100
C                                                                        00015110
      IF(IND(48).EQ.5) GO TO 999                                         00015120
C                                                                        00015130
C     STORE GENERAL ORBIT GENERATOR TYPE FROM IND(1) SWITCH VALID FOR    00015140
C     THE ORBIT GENERATOR,DC OR DATA SIMULATION PROGRAMS                 00015150
C                                                                        00015160
      IF(IND(48).EQ.9) GO TO 825                                         00015170
      GO TO (821,821,822,822,821,824,170,170,821,821,                    00015180
     *    90,822,822,822,822), IND1                                      00015190
C                                                                        00015200
C     NUMERICAL( COWELL,TIME-REGULARIZED AND SUPER-GENERAL)              00015210
      GO TO 825                                                          00015220
  821 INDTYP = 1                                                         00015230
       GO TO 825                                                         00015240
C     ANALYTIC(BROUWER,BROUWER-LYDDANE,BROUWER-GORDON,VINTI)             00015250
  822 INDTYP = 2                                                         00015260
      GO TO 825                                                          00015270
C                                                                        00015280
C     PRE-GENERATED ORBIT FILE                                           00015290
  824 INDTYP = 4                                                         00015300
      GO TO 999                                                          00015310
C                                                                        00015320
C     STORE ECCENTRICITY AND 1 - E FOR CENTRAL BODY                      00015330
C                                                                        00015340
  825 IF(IND1.NE.5.OR.IND(3).LE.10) GO TO 8253                           00015350
      IND(22) = 1                                                        00015360
      IF(ICTYPE.NE.4) GO TO 8251                                         00015370
      IND(22) = 2                                                        00015380
      ICTYPE= 2                                                          00015390
```

```
      ELEMS(1) = BMELS(1)                                                    00015400
      ELEMS(2) = BMELS(2)                                                    00015410
      ELEMS(3) = BMELS(3)                                                    00015420
      ELEMS(4) = BMELS(4)                                                    00015430
      ELEMS(5) = BMELS(5)                                                    00015440
      ELEMS(6) = BMELS(6)                                                    00015450
 8251 IF(ICTYPE.NE.6) GO TO 8253                                             00015460
      IND(22) = 2                                                            00015470
      ICTYPE = 2                                                             00015480
 8253 ICENT = INDSEC(1,1)                                                    00015490
      IF(FLAT(ICENT).EQ.0.D0) GO TO 826                                      00015500
      FLATIN = 1.D0 / FLAT(ICENT)                                            00015510
      ESQ = 2.D0 * FLATIN - FLATIN ** 2                                      00015520
      BSQ = 1.D0 - ESQ                                                       00015530
      GO TO 827                                                              00015540
  826 ESQ = 0.D0                                                             00015550
      BSQ = 1.D0                                                             00015560
  827 CONTINUE                                                               00015570
      VAB = AB(ICENT) ** 2                                                   00015580
      IFLAG = 1                                                              00015590
      IF(NTSECS.NE.1) IFLAG = 2                                              00015600
      IF(NMANVS.NE.0) IFLAG = 2                                              00015610
C      TEST IF THIS IS A SHORT TYPE SECTIONING RUN.                          00015620
C                                                                            00015630
      ITEST=INDSEC(14,1) + INDSEC(15,1) + INDSEC(19,1) +                     00015640
     *  INDSEC(29,1)+INDSEC(30,1)                                            00015650
      IF(ITEST.NE.10) IFLAG=2                                                00015660
      NM = 0                                                                 00015670
      IF(IND(48).EQ.9) GO TO 830                                             00015680
      IF(IND(1) .NE. 1 .AND. IND(2) .NE. 1)   GO TO 829                      00015690
      DO 828 I = 1,10                                                        00015700
      IF(INDSEC(28,I) .EQ. 2)   GO TO 828                                    00015710
      INDSEC(28,I) = 2                                                       00015720
      NM = 1                                                                 00015730
  828 CONTINUE                                                               00015740
  829 IF(NM .EQ. 0)   GO TO 830                                              00015750
      ASSIGN 830 TO M                                                        00015760
      IERR=430                                                               00015770
      GO TO 845                                                              00015780
  830 CONTINUE                                                               00015790
C                                                                            00015800
C     ONLY THE FIXED STEP MODE SHOULD BE USED WITH TIME REGULARIZED          00015810
C     COWELL  THE STEP MODE IS CHANGED TO FIXED STEP MODE IF IT IS NOT       00015820
C                                                                            00015830
      NM=0                                                                   00015840
      IF (IND(1).NE.1) GO TO 8306                                            00015850
      DO 8302 I=1,10                                                         00015860
      IF (INDSEC(5,I).EQ.1) GO TO 8302                                       00015870
      INDSEC(5,I)=1                                                          00015880
      NM=1                                                                   00015890
 8302 CONTINUE                                                               00015900
      IF (NM.EQ.0) GO TO 8306                                                00015910
      ASSIGN 8306 TO M                                                       00015920
      IERR=217                                                               00015930
      GO TO 845                                                              00015940
 8306 CONTINUE                                                               00015950
      IF( IND(48).GT.2 ) GO TO 999                                           00015960
```

```
SETRUN              JSS1493.GTDS.UPDATE.FORT

C                                                                       00015970
      IF(IND(48).NE.1.AND.IND(48).NE.10)GO TO 870                       00015980
      IF(IND(68) .EQ. 1) GO TO 850                                      00015990
      DO 833 J=1,NOISTA                                                 00016000
      IF(OBCOR(2,J)) GO TO 834                                          00016010
  833 CONTINUE                                                          00016020
      GO TO 850                                                         00016030
  834 IND(68) = 1                                                       00016040
      IND(63) = 1                                                       00016050
      IF(.NOT. WFCALL) GO TO 860                                        00016060
      CALL IRWF (MSGERR,&855)                                           00016070
      GO TO 870                                                         00016080
  835 ASSIGN 780 TO M                                                   00016090
      IERR=431                                                          00016100
      GO TO 845                                                         00016110
  840 IF(IPRINT.NE.1)  CALL MSGWTR(2,DUM)                               00016120
  845 IERROR=IERROR+1                                                   00016130
  846 CALL ERROUT(IERR)                                                 00016140
      GO TO M, (39,90,655,660,780,790,830,2703,8306)                    00016150
  850 IF(.NOT.WFCALL) GO TO 860                                         00016160
      GO TO 870                                                         00016170
C                                                                       00016180
C     ERROR RETURN FROM IRWF                                            00016190
C                                                                       00016200
  855 CALL ERROUT(MSGERR)                                               00016210
      RETURN 1                                                          00016220
  860 IF(INDI48.EQ.10)IND(48)=1                                         00016230
      CALL WFCONT(&807)                                                 00016240
  870 CONTINUE                                                          00016250
      IND(48) = INDI48                                                  00016260
  999 RETURN                                                            00016270
C                                                                       00016280
 1830 FORMAT(6(A8,2X),A8,I2,1X,I7)                                      00016290
 1840 FORMAT(1H0,7A8,I7)                                                00016300
 1860 FORMAT (A8,3I3,3G21.14)                                           00016310
 1865 FORMAT(//,15X,'***** WARNING ********  ORBTYPE CARD IS NOT BEING  00016320
     1 USED, TWOBODY  OR DEFAULT VALUES APPLY  ********',//)            00016330
C                                                                       00016340
      END                                                               00016350

******** END OF MEMBER SETRUN    1635 RECORDS ********
```

APPENDIX D

STAND-ALONE PARALLEL PROCESSOR ORBIT DETERMINATION SOFTWARE DESCRIPTION

The stand-alone version of the parallel processor orbit determination software is described in this appendix. This version contains both the Picard-Chebyshev parallel orbit propagation algorithm and the Parallel Variable Metric orbit estimation algorithm. The overall structure of the system is described in the first section. Brief descriptions and listings of the subroutines are presented. Listings of the common data blocks are provided and the subroutines using them are indicated. Finally, the format of the input card images and a brief user's guide are presented.

D.1 Overall System Structure

The stand-alone parallel processor orbit determination system was built as a software test bed to demonstrate the capabilities of the Parallel Variable Metric orbit estimation algorithm. The system driver is routine 'MAIN' and it has two modes of operation. In the run set-up mode,

'MAIN' passes control to subroutine 'RUNSET', which processes the input card images and initializes the appropriate data variables and common blocks. In the run execution mode, 'MAIN' passes control to either subroutine 'EPHEX' or subroutine 'ESTEX', depending on the type of run being made.

Subroutine 'EPHEX' is the main executive for the ephemeris generation capability of the stand-alone system. This subroutine controls the Picard-Chebyshev integration of the particular formulation of the equations of motion being integrated. When it is required to take an integration step, 'EPHEX' calls subroutine 'VTEST', which executes the Picard-Chebyshev integration algorithm as described in Chapter 2. If it is only required to interpolate for the integration state components at some intermediate output time, 'EPHEX' calls subroutine 'CSRSTA' to perform the necessary Chebyshev series evaluations. Subroutine 'EPHEX' also controls the output of ephemeris data to the line printer by calls to subroutine 'ELMRPT'. The user has considerable flexibility in tailoring his ephemeris printout to his needs by selecting certain printout options in his input deck. Finally, if it is desired to compute the analytical state partial derivatives formulated in Chapter 3.3.2, 'EPHEX' calls subroutine 'PXTSLV' to perform this task.

The main executive for the orbit estimation capability is subroutine 'ESTEX'. When this subroutine is invoked by 'MAIN', 'ESTEX' must first determine the desired method of computing the cost functional and its gradient vector. If the direct integration method is indicated, 'ESTEX' will call subroutine 'CGINTG' each time a cost or gradient evaluation is required by the Parallel Variable Metric (PVM) estimation algorithm described in Chapter 3. Subroutine 'CGINTG' in turn calls subroutine 'CMPTOB', which numerically integrates the required trajectory through the observation span by making appropriate calls to subroutine 'VTEST'. 'CMPTOB' calls subroutine 'CSRSTA' to evaluate the resulting Chebyshev series to determine the spacecraft position and velocity at the observation times. Additionally, 'CMPTOB' calls subroutine 'PXTSLV' to compute the necessary partial derivatives for use in the gradient computation. If the linearization method is specified, subroutine 'ESTEX' also calls subroutine 'CGLINE' for the appropriate linearization computations, as described in Chapter 3.3.4. When the PVM algorithm requires the univariate search, subroutine 'ESTEX' calls subroutine 'SERCH1'. This subroutine must again call subroutines 'CGINTG' and/or 'CGLINE' for the required cost functional evaluations. After the algorithm has converged, 'ESTEX' calls subroutine

'COVMTX' to compute and output the updated covariance matrix of the estimation errors based on the converged solution.

D.2 Subroutine Descriptions

The functions performed by the subroutines in the stand-alone system are described in Tables 2.2 and 3.15 of this document. Figure D.1 presents a schematic of the subroutine structure for this stand-alone system. The structure indicates the hierarchy of subroutine calls within the system. As an additional aid in understanding the software interfaces in the stand-alone system, Tables D.1 and D.2 have been included in this section. Table D.1 lists each system subroutine in the left hand column and those subroutines that it calls are listed in the right hand column. Conversely, Table D.2 lists each system subroutine in the left hand column and all subroutines that reference it are listed in the right hand column. These tables have been found to be very useful in trouble-shooting the system.

D.3 Common Block Descriptions

A primary means of communication in the stand-alone system is through named common data blocks. These blocks

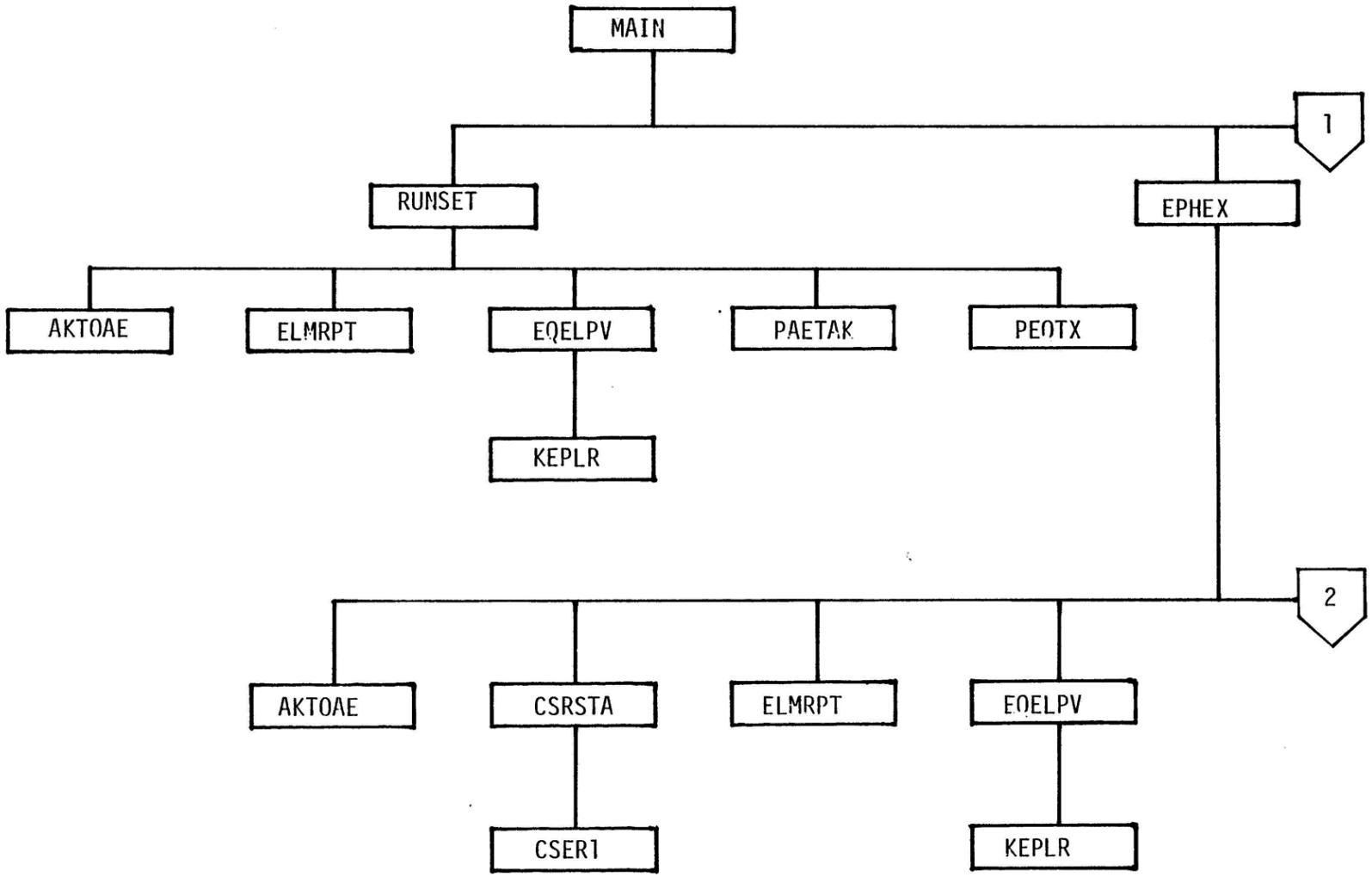


Figure D.1. Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System (Page 1 of 6)

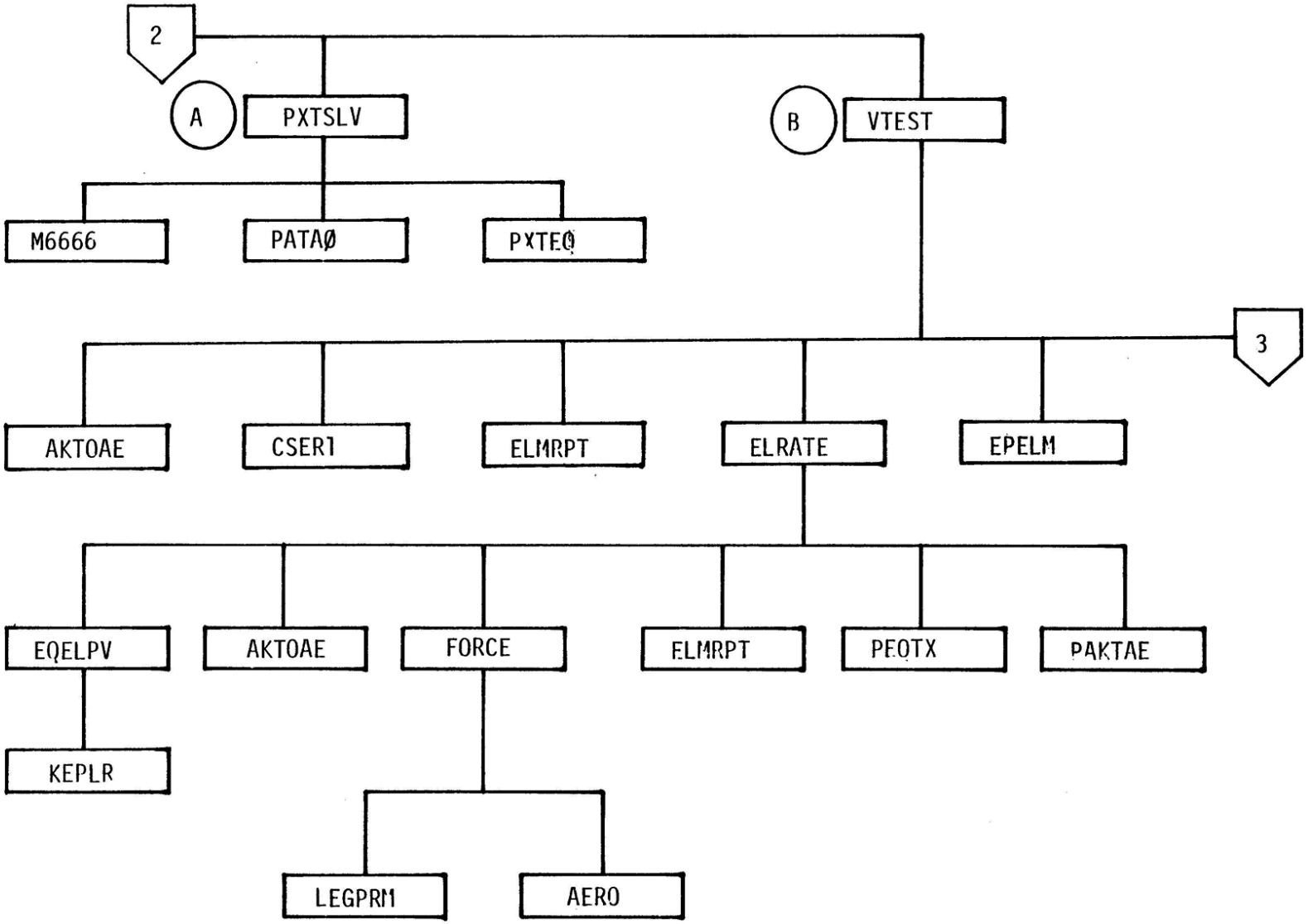


Figure D.1. Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System (Page 2 of 6)

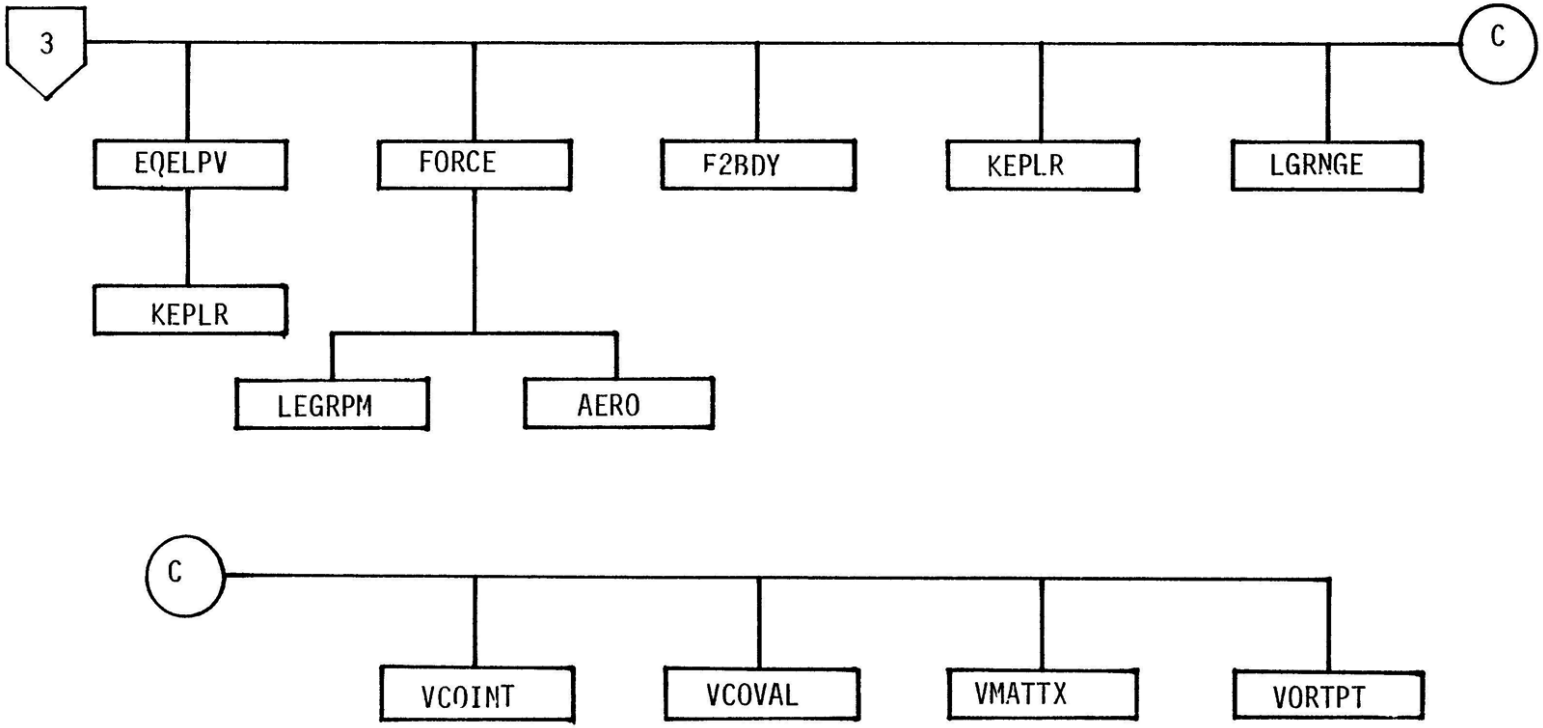


Figure D.1. Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System (Page 3 of 6)

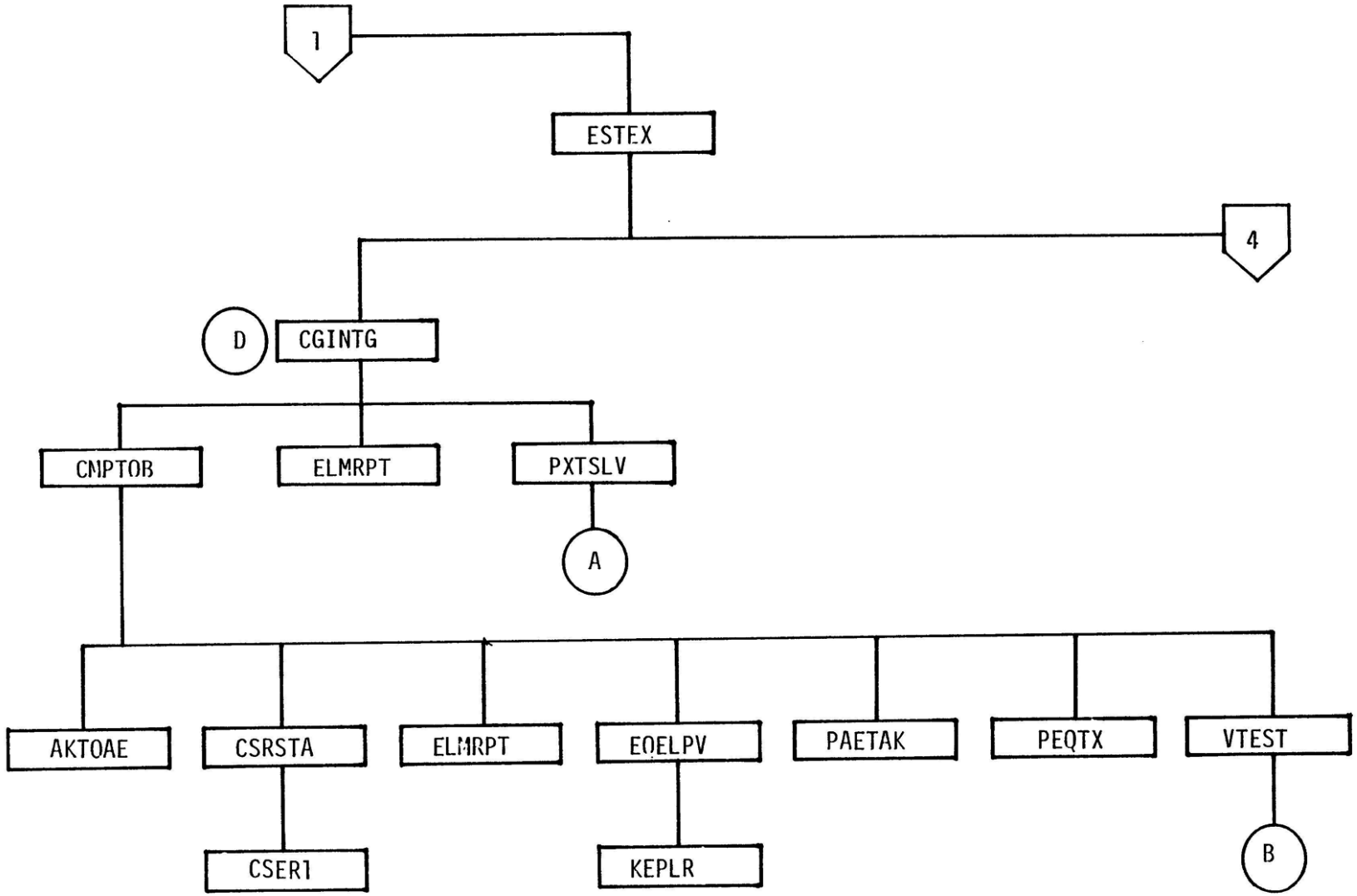


Figure D.1. Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System (Page 4 of 6)

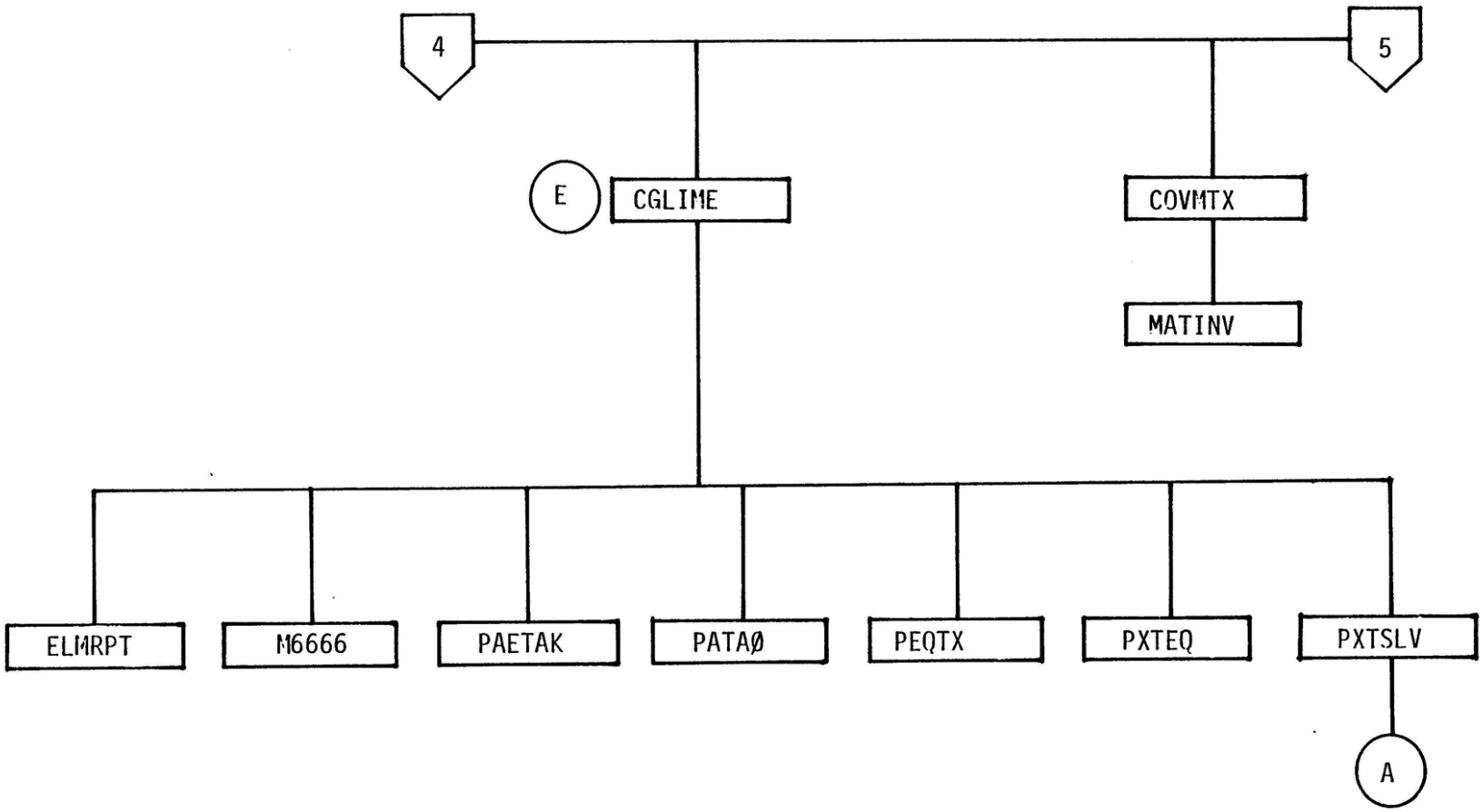


Figure D.1. Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System (Page 5 of 6)

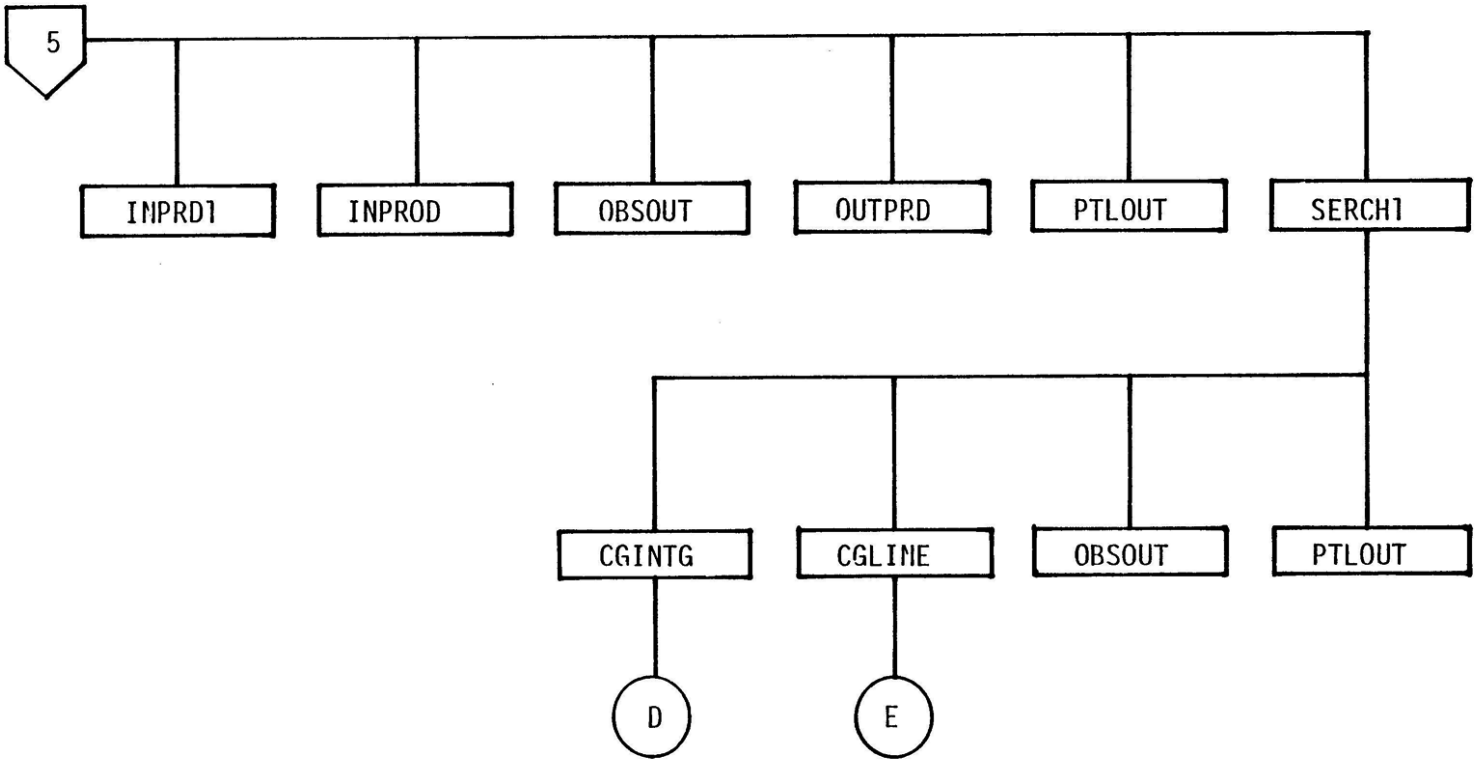


Figure D.1. Subroutine Structure for Stand-Alone Parallel Processor Orbit Determination Software System (Page 6 of 6)

Table D.1. Parallel Processor Orbit Determination System Subroutine Description: "Who Calls Whom" (Page 1 of 3)

SUBROUTINE	CALLED SUBROUTINES
AERO	
AKTOAE	
CGINTG	CMPTOB, ELMRPT, PXTSLV
CGLINE	ELMRPT, M6666, PAETAK, PATAØ, PEQTX, PXTEQ, PXTSLV
CMPTOB	AKTOAE, CSRSTA, ELMRPT, EQELPV, PAETAK, PEQTX, VTEST
COVMTX	MATINV
CSER1	
CSRSTA	CSER1
ELMRPT	
ELRATE	AKTOAE, ELMRPT, EQELPV, FORCE, PAKTAE, PEQTX
EPELM	
EPHEX	AKTOAE, CSRSTA, ELMRPT, EQELPV, PXTSLV, VTEST
EQELPV	KEPLR
ESTEX	CGINTG, CGLINE, COVMTX, INPRD1, INPROD, OBSOUT, OUTPRD, PTLOUT, SERCH1
FORCE	AERO, LEGPRM
F2BDY	
INPRD1	
INPROD	

Table D.1. Parallel Processor Orbit Determination System Subroutine Description: "Who Calls Whom" (Page 2 of 3)

SUBROUTINE	CALLED SUBROUTINES
KEPLR	
LEGPRM	
LGRNGE	
MAIN	EPHEX, ESTEX, RUNSET
MATINV	
M6666	
OBSOUT	
OUTPRD	
PAETAK	
PAKTAE	
PATAØ	
PEQTX	
PTLOUT	
PXTEQ	
PXTSLV	M6666, PATAØ, PXTEQ
RUNSET	AKTOAE, ELMRPT, EQELPV, PAETAK, PEQTX
SERCH1	CGINTG, CGLINE, OBSOUT, PTLOUT
VCOINT	CSER1
VCOVAL	
VMATTX	

Table D.1. Parallel Processor Orbit Determination System Subroutine Description: "Who Calls Whom"
(Page 3 of 3)

SUBROUTINE	CALLED SUBROUTINES
VORTPT	
VTEST	AKTOAE, CSER1, ELMRPT, ELRATE, EPELM, EQELPV, FORCE, F2BDY, KEPLR, LGRNGE, VCOINT, VCOVAL, VMATTX, VORTPT

Table D.2. Parallel Processor Orbit Determination System Subroutine Description: "Who Is Called By Whom" (Page 1 of 3)

SUBROUTINE	CALLED BY THESE SUBROUTINES
AERO	FORCE
AKTOAE	CMPTOB, ELRATE, EPHEX, RUNSET, VTEST
CGINTG	ESTEX, SERCH1
CGLINE	ESTEX, SERCH1
CMPTOB	CGINTG
COVMTX	ESTEX
CSER1	CSRSTA, VCOINT, VTEST
CSRSTA	CMPTOB, EPHEX
ELMRPT	CGINTG, CGLINE, CMPTOB, ELRATE, EPHEX, RUNSET, VTEST
ELRATE	VTEST
EPELM	VTEST
EPHEX	MAIN
EQELPV	CMPTOB, ELRATE, EPHEX, RUNSET, VTEST
ESTEX	MAIN
FORCE	ELRATE, VTEST
F2BDY	VTEST
INPRD1	ESTEX
INPROD	ESTEX
KEPLR	EQELPV, VTEST

Table D.2. Parallel Processor Orbit Determination System Subroutine Description: "Who Is Called By Whom" (Page 2 of 3)

SUBROUTINE	CALLED BY THESE SUBROUTINES
LEGPRM	FORCE
LGRNGE	VTEST
MAIN	
MATINV	COVMTX
M6666	CGLINE, PXTSLV
OBSOUT	ESTEX, SERCH1
OUTPRD	ESTEX
PAETAK	CGLINE, CMPTOB, RUNSET
PAKTAE	ELRATE
PATAØ	CGLINE, PXTSLV
PEQTX	CGLINE, CMPTOB, ELRATE, RUNSET
PTLOUT	ESTEX, SERCH1
PXTEQ	CGLINE, PXTSLV
PXTSLV	CGINTG, CGLINE, EPHEX
RUNSET	MAIN
SERCH1	ESTEX
VCOINT	VTEST
VCOVAL	VTEST
VMATTX	VTEST

Table D.2. Parallel Processor Orbit Determination System Subroutine Description: "Who Is Called By Whom" (Page 3 of 3)

SUBROUTINE	CALLED BY THESE SUBROUTINES
VORTPT	VTEST
VTEST	CMPTOB, EPHEX

are described in the program listings in Section D.5. Additionally, Table D.3 lists each named common block and indicates which subroutines reference data items in that block.

D.4 Input Card Image Formats

Input to the stand-alone system is accomplished by a card deck. The different types of card inputs allow considerable flexibility in the use of the system. Figures D.2 and D.3 show typical input card decks for an ephemeris generation run and an orbit estimation run respectively. Figure D.4 describes the fields of the input card images.

The user is cautioned that the system expects certain cards to be included for each type of run. For an ephemeris generation run, the system requires the following input cards: EPHEMGEN, ELEMENT1, ELEMENT2, P-C INTG, FORC MDL, INT SPAN, PRINTOUT and LASTCARD. For an orbit estimation run, the system requires all the input cards except EPHEMGEN and SCPARAM (if drag is not included in force model). If a mandatory card is not included in the input deck, the subroutine 'RUNSET' will print a warning message but execution of the run will continue until completed or terminated by an error count. Some care should be taken in setting up the input deck.

Table D.3.

Named Common Block References

COMMON BLOCKS	REFERENCED BY THESE SUBROUTINES
/CHBINT/	VCOVAL, VMATTX, VORTPT, VTEST
/CNVRG/	RUNSET, VTEST
/COVAL/	CSRSTA, VCOINT, VCOVAL, VTEST
/DRGPAR/	AERO, FORCE, RUNSET
/EQPARM/	ELMRPT, PEQTX, PXTEQ
/EQPTLX/	CGLINE, CMPTOB, ELRATE, PEQTX, PXTSLV, RUNSET
/ERTSHP/	AERO, CGINTG, CGLINE, CMPTOB, EPHEX, FORCE, PATAØ, RUNSET
/ICNTRL/	CGINTG, CGLINE, CMPTOB, COVMTX, ESTEX, EPHEX, MAIN, PTLOUT, PXTSLV, RUNSET, VTEST
/ISTSTC/	ESTEX, SERCH1
/NPCNTL/	CMPTOB, CSRSTA, ELRATE, EPHEX, RUNSET, VTEST
/OBSDAT/	CGINTG, CGLINE, CMPTOB, COVMTX, OBSOUT, PTLOUT
/STAPAR/	CGINTG, CGLINE, EPHEX, PXTSLV
/STATE/	CMPTOB, EPHEX, RUNSET, VCOINT, VTEST
/TIMER/	CGINTG, CGLINE, CMPTOB, EPHEX, RUNSET, VTEST
/XEPOCH/	CGLINE, CMPTOB, PATAØ, PXTSLV, RUNSET
/XESTMT/	CGLINE, ESTEX, RUNSET, SERCH1
/XPTLEQ/	CGLINE, PXTEQ, PXTSLV
/ZNLCOF/	FORCE, PATAØ, RUNSET

```
**********PARALLEL PROCESSING ORBIT DETERMINATION INPUT DECK**********

EPHEMGEN   0  0  0   0.0                      0.0                      0.0
ELEMENT1   1  0  0  -4.64855860223441D+02    6.66780058109643D+03     5.74230935280705D+02
ELEMENT2   0  0  0  -2.83811863629232D+00   -7.87189812063128D-01     7.08302748864198D+00
FORC MDL   8  0  0   1.39154000000000D+09    6.37814000000000D+03     7.16280000000000D+00
SCPARAM    1  0  0   2.00000000000000D+00    1.86000000000000D-06     6.77000000000000D+02
P-C INTG  64 69  3   4.00000000000000D+01    1.00000000000000D-10     5.40000000000000D+03
INT SPAN   0  0  0   1.80000000000000D+02    0.0                      6.00000000000000D+01
PRINTOUT   1  1  1   0.0                      0.0                      2.00000000000000D+00
PARTIALS   1  1  6   0.0                      1.00000000000000D+00     0.0
LASTCARD   0  0  0   0.0                      0.0                      0.0
```

Figure D.2. Example Input Deck for Ephemeris Generation Run

```
**********PARALLEL PROCESSING ORBIT DETERMINATION INPUT DECK**********

ESTIMATE  0  0  0  0.0                        0.0                        0.0
ELEMENT1  2  0  0  6.900000000000000D+03   2.500000000000000D-02   3.000000000000000D+01
ELEMENT2  0  0  0  1.700000000000000D+02   2.600000000000000D+02   6.000000000000000D+01
FORC MDL  4  0  1  1.391540000000000D+09   6.378140000000000D+03   7.162800000000000D+00
SCPARAM   1  0  0  2.000000000000000D+00   1.860000000000000D-06   6.770000000000000D+02
P-C INTG 64 69  1  9.900000000000000D+01   1.000000000000000D-10   5.400000000000000D+03
INT SPAN  0  0  0  1.800000000000000D+02   0.0                        6.000000000000000D+01
PRINTOUT  1  1  1  0.0                        0.0                        2.000000000000000D+00
PARTIALS  1  3  6  0.0                        1.000000000000000D+00   0.0
EST PARS  2  0  6  0.0                        1.000000000000000D-02   1.000000000000000D+00
1-D SRCH  0  0  0  0.0                        5.000000000000000D-01   1.000000000000000D-03
APSLVFCR  1  0  0  6.900000000000000D+03   2.349231551964750-02   8.550503583141620-03
APSLVFCR  4  0  0 -2.638784421195300-01  -4.652886397299020-02   1.300000000000000D+02
APCOVMTX  1  0  1  0.0                        0.0                        0.0
APCOVMTX  1  0  4  0.0                        0.0                        0.0
APCOVMTX  2  0  1  0.0                        0.0                        0.0
APCOVMTX  2  0  4  0.0                        0.0                        0.0
APCOVMTX  3  0  1  0.0                        0.0                        0.0
APCOVMTX  3  0  4  0.0                        0.0                        0.0
APCOVMTX  4  0  1  0.0                        0.0                        0.0
APCOVMTX  4  0  4  0.0                        0.0                        0.0
APCOVMTX  5  0  1  0.0                        0.0                        0.0
APCOVMTX  5  0  4  0.0                        0.0                        0.0
APCOVMTX  6  0  1  0.0                        0.0                        0.0
APCOVMTX  6  0  4  0.0                        0.0                        0.0
SCLFCTRS  1  0  0  1.000000000000000D+01   1.000000000000000D-04   1.000000000000000D-04
SCLFCTRS  4  0  0  1.000000000000000D-04   1.000000000000000D-04   1.000000000000000D-02
LASTCARD  0  0  0  0.0                        0.0                        0.0
```

Figure D.3. Example Input Deck for Orbit Estimation Run

APCOVMTX

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	APCOVMTX - keyword to input elements of the *a priori* estimation covariance matrix
9-11	I3	Row indicator for the row in the 6x6 *a priori* covariance matrix being initialized (= I1)
12-14	I3	Blank
15-17	I3	Column indicator for the first of the three values to be input in Row I1 in the *a priori* covariance matrix being initialized (= I3)
18-38	G21.14	Covariance data to be loaded into Row I1, Column I3
39-59	G21.14	Covariance data to be loaded into Row I1, Column I3 + 1
60-80	G21.14	Covariance data to be loaded into Row I1, Column I3 + 2

Figure D.4. Input Card Formats (Page 1 of 17)

APSLVFOR

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	APSLVFOR - keyword to input components of the *a priori* solve-for vector
9-11	I3	Index of the first component of the 6x1 solve-for vector to be loaded: 1 = Load the three real fields on this card into the first three components of the *a priori* solve-for vector 4 = Load the three real fields on this card into the last three components of the *a priori* solve-for vector
12-14	I3	Blank
15-17	I3	Blank
18-38	G21.14	Component 1 or 4 of the *a priori* solve-for vector
39-59	G21.14	Component 2 or 5 of the *a priori* solve-for vector
60-80	G21.14	Component 3 or 6 of the *a priori* solve-for vector

Figure D.4. Input Card Formats (Page 2 of 17)

Note: For the first integer field = 1

	Cartesian	Keplerian	Equinoctial
Real Field 1	r_x (km)	a (km)	a (km)
Real Field 2	r_y (km)	e	h
Real Field 3	r_z (km)	i (deg)	k

For the first integer field = 4:

	Cartesian	Keplerian	Equinoctial
Real Field 1	$\dot{r}_x$ (km/sec)	ω (deg)	p
Real Field 2	$\dot{r}_y$ (km/sec)	Ω (deg)	q
Real Field 3	$\dot{r}_z$ (km/sec)	M (deg)	λ (deg)

Figure D.4. Input Card Formats (Page 3 of 17)

ELEMENT1

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	ELEMENT1 - keyword to set the first three components of the initial integration state vector and to identify the input coordinate system type
9-11	I3	Input coordinate system type: 1 = Cartesian 2 = Keplerian 3 = Equinoctial
12-14	I3	Blank
15-17	I3	Blank

The three real fields contain the first three elements of the satellite epoch state vector:

		Cartesian	Keplerian	Equinoctial
18-38	G21.14	r_x (km)	a (km)	a (km)
39-59	G21.14	r_y (km)	e	h
60-80	G21.14	r_z (km)	i (deg)	k

Figure D.4. Input Card Formats (Page 4 of 17)

ELEMENT2

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	ELEMENT2 - keyword to set the last three components of the initial integration state vector
9-11	I3	Blank
12-14	I3	Blank
15-17	I3	Blank

The three real fields contain the last three elements of the satellite epoch state vector. The particular element set is indicated by the first integer field of the ELEMENT1 card.

		Cartesian	Keplerian	Equinoctial
18-38	G21.14	$\dot{r}_x$ (km/sec)	ω (deg)	p
39-59	G21.14	$\dot{r}_y$ (km/sec)	Ω (deg)	q
60-80	G21.14	$\dot{r}_z$ (km/sec)	M (deg)	λ (deg)

Figure D.4. Input Card Formats (Page 5 of 17)

EPHEMGEN

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION
- Detailed format:

Columns	Format	Description
1-8	A8	EPHEMGEN - keyword to indicate the beginning of an input deck for an ephemeris generation run

All other fields are blank.

Figure D.4. Input Card Formats (Page 6 of 17)

ESTIMATE

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	ESTIMATE - keyword to indicate the beginning of an input deck for an orbit estimation run

All other fields are blank.

Figure D.4. Input Card Formats (Page 7 of 17)

EST PARS

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	EST PARS - keyword for input of control parameters for the Parallel Variable Metric orbit estimation algorithm
9-11	I3	Method of computing cost functional and gradient of the cost functional: 1 = Direct numerical integration 2 = Linearized approximation 3 = Linearized approximation with recomputation of the observation partial derivatives
12-14	I3	Blank
15-17	I3	Number of parallel processor elements in the assumed SIMD hardware architecture
18-38	G21.14	Blank
39-59	G21.14	Convergence criterion for estimation procedure
60-80	G21.14	Flag to indicate that the estimation error covariance matrix is to be updated using the converged solve-for vector: 0.0 = Do not update covariance matrix 1.0 = Update covariance matrix

Figure D.4. Input Card Formats (Page 8 of 17)

FORC MDL

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	FORC MDL - keyword for input of desired force model quantities
9-11	I3	Maximum order of zonal harmonic field to be included in the force model (0 or 2-8)
12-14	I3	Flag indicating an update of the atmospheric density model parameters is desired: 0 = Use default values 1 = Use update values in the three real fields
15-17	I3	Flag indicating drag should be included in the force model: 0 = Do not include drag effects 1 = Include drag effects
18-38	G21.14	Atmospheric density ρ_o at reference distance r_o (kg/km^3)
39-59	G21.14	Reference distance r_o, generally taken as the mean radius of central body (km)
60-80	G21.14	Atmospheric scale height (km)

Note: The real fields only need to be included if it is desired to change the parameters in the exponential density model.

Figure D.4. Input Card Formats (Page 9 of 17)

INT SPAN

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	INT SPAN - keyword for input of the length of the integration/observation span in days, hours, minutes, seconds from epoch and for input of the output print interval in seconds
9-11	I3	Blank
12-14	I3	Number of days from epoch in integration/observation span
15-17	I3	Number of hours from epoch in integration/observation span
18-38	G21.14	Number of minutes from epoch in integration/observation span
39-59	G21.14	Number of seconds from epoch in integration/observation span
60-80	G21.14	Output print interval in seconds

Figure D.4. Input Card Formats (Page 10 of 17)

PARTIALS

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	PARTIALS - keyword for input of control parameters for generation of the analytical state partial derivatives
9-11	I3	Flag indicating analytical partial derivatives are to be computed: 0 = Do not compute partials 1 = Compute partial derivatives of current state with respect to selected solve-for parameters Note: This must be set to 1 for an orbit estimation run
12-14	I3	Solve-for parameter set: 1 = Epoch Cartesian elements 2 = Epoch Keplerian elements 3 = Epoch equinoctial elements
15-17	I3	Number of solve-for parameters in the selected solve-for parameter set (currently hard-coded to 6)
18-38	G21.14	Blank
39-59	G21.14	Flag indicating the secular rates due to J_2 are to be included in the partial derivative computation: 0.0 = Do not include J_2 rates 1.0 = Include J_2 secular rates
60-80	G21.14	Blank

Figure D.4. Input Card Formats (Page 11 of 17)

P-C INTEG

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	P-C INTEG - keyword for input of control parameters for Picard-Chebyshev integrator
9-11	I3	Maximum order of Chebyshev polynomial series representing the integrands (<68)
12-14	I3	Subscript of final orthogonality point used in the computation of the Chebyshev series coefficients (<70)
15-17	I3	Flag indicating the equation set being integrated: 1 = Cowell equations 2 = Keplerian VOP equations 3 = Equinoctial VOP equations (λ)
18-38	G21.14	Terminal iteration number
39-59	G21.14	Convergence criterion for integrator
60-80	G21.14	Integration stepsize (seconds)

Figure D.4. Input Card Formats (Page 12 of 17)

PRINTOUT

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	PRINTOUT - keyword for input of print format control options
9-11	I3	Control flag for integration printout at each output time: 0 = Do not print integration report 1 = Print only position velocity and orbital elements at each output time 2 = Print position, velocity, orbital elements and 20 auxiliary quantities at each output time
12-14	I3	Control flag for partial derivative printout at each output time: 0 = Do not print analytical partial derivatives 1 = Print analytical partial derivatives of current state with respect to solve-for parameters at each output time
15-17	I3	Control flag for Picard-Chebyshev integration convergence report: 0 = Do not print convergence report 1 = Print Picard-Chebyshev convergence report at the beginning of each integration arc

Figure D.4. Input Card Formats (Page 13 of 17)

Columns	Format	Description
18-38	G21.14	Control flag for observation processing report: 0.0 = Do not print observation processing report 1.0 = Print observation processing report at each evalution of the cost functional and/or its gradient
39-59	G21.14	Control flag for observation partial derivative report: 0.0 = Do not print observation partials 1.0 = Print observation partial derivatives at each evaluation of the cost functional gradient vector
60-80	G21.14	Output orbital element set for the ephemeris generation integration report: 1.0 = Keplerian orbital elements 2.0 = Equinoctial orbital elements

Figure D.4. Input Card Formats (Page 14 of 17)

SCLFCTRS

- Card Format: (A8, 3I3, 3G21.14)
- Applicable modes: ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	SCLFCTRS - keyword for input of the scale factors used to adjust the magnitudes of the parameter space basis vectors in the Parallel Variable Metric orbit estimation algorithm
9-11	I3	Index of the basis vector which will be scaled by the scale factor in the first real field of this card (= I)
12-14	I3	Blank
15-17	I3	Blank
18-38	G21.14	Scale factor for basis vector $\hat{\sigma}_I$
39-59	G21.14	Scale factor for basis vector $\hat{\sigma}_{I+1}$
60-80	G21.14	Scale factor for basis vector $\hat{\sigma}_{I+2}$

Note: Magnitude of a basis vector should be adjusted to be of the order 10^{-4} (or less) of the order of magnitude of its corresponding component in the solve-for vector.

Figure D.4. Input Card Formats (Page 15 of 17)

SCPARAM

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: EPHEMERIS GENERATION, ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	SCPARAM - keyword for input of certain spacecraft physical data
9-11	I3	Spacecraft identifier (currently unused)
12-14	I3	Blank
15-17	I3	Blank
18-38	G21.14	Coefficient of drag for this spacecraft
39-59	G21.14	Spacecraft cross-sectional area perpendicular to the direction of motion (km^2)
60-80	G21.14	Spacecraft mass (kg)

Figure D.4. Input Card Formats (Page 16 of 17)

1-D SRCH

- Card format: (A8, 3I3, 3G21.14)
- Applicable modes: ORBIT ESTIMATION
- Detailed format:

Columns	Format	Description
1-8	A8	1-D SRCH - keyword for input of control parameters for the univariate search in the Parallel Variable Metric orbit estimation algorithm
9-11	I3	Blank
12-14	I3	Blank
15-17	I3	Blank
18-38	G21.14	Initial estimate of the independent variable α in the univariate search
39-59	G21.14	Initial estimate of the univariate search stepsize
60-80	G21.14	Convergence criterion for the univariate search algorithm

Figure D.4. Input Card Formats (Page 17 of 17)

D.5 Program Listings

The following pages contain the FORTRAN IV listings of the 40 subroutines and 18 named common data blocks in the stand-alone version. These listings are rather extensively commented and make a useful reference in understanding the stand-alone system.

```
      SUBROUTINE AERO(ST,ASUBD)                                          00000010
C                                                                        00000020
C     VERSION OF 08/07/79     **** CHECKED OUT ****                      00000030
C                                                                        00000040
C     PURPOSE                                                            00000050
C         TO COMPUTE THE CONTRIBUTION TO THE TOTAL SPACECRAFT ACCELER-    00000060
C         ATION FROM ATMOSPHERIC DRAG                                    00000070
C                                                                        00000080
C     METHOD                                                             00000090
C         GIVEN THE SPACECRAFT POSITION AND VELOCITY VECTORS IN INERTIAL00000100
C         CARTESIAN COORDINATES, THE DRAG ACCELERATION IS COMPUTED IN    00000110
C         THE FOLLOWING MANNER:                                          00000120
C                                                                        00000130
C                                                                        00000140
C                 -    -                                -    -         00000150
C                 A  = -(1/2) * C  * (A/M) * RHO * V * V                 00000160
C                  D             D                    A   A             00000170
C                                                                        00000180
C             WHERE  C  = DRAG COEFFICIENT                               00000190
C                     D                                                  00000200
C                   A/M = SPACECRAFT AREA-TO-MASS RATIO                  00000210
C                   RHO = ATMOSPHERIC DENSITY AT R                       00000220
C                                                                        00000230
C                    -                                                   00000240
C                    V  = VELOCITY OF THE SPACECRAFT RELATIVE TO THE     00000250
C                     A        ROTATING ATMOSPHERE                      00000260
C                                                                        00000270
C                       -    -    -                                     00000280
C                       = V  -  V                                        00000290
C                               ATM/I                                    00000300
C                                                                        00000310
C                    -                                                   00000320
C                    V  = VELOCITY OF THE SPACECRAFT RELATIVE TO INER-   00000330
C                         TIAL SPACE                                     00000340
C                                                                        00000350
C                  -                                                     00000360
C                  V      = VELOCITY OF THE ATMOSPHERE RELATIVE TO INER- 00000370
C                   ATM/I        TIAL SPACE.  THE ASSUMPTION THAT THE    00000380
C                                ATMOSPHERE ROTATES AS A RIGID BODY ABOUT00000390
C                                THE EARTH ALLOWS US TO COMPUTE THIS AS  00000400
C                                THE CROSS PRODUCT OF THE ATMOSPHERIC    00000410
C                                ANGULAR VELOCITY AND THE SATELLITE POSI-00000420
C                                TION VECTOR.                            00000430
C                                                                        00000440
C         THE DENSITY IS COMPUTED BY USING A SIMPLE EXPONENTIAL DENSITY  00000450
C         MODEL OF THE FORM:                                             00000460
C                                                                        00000470
C                   RHO = RHO  * EXP( -(R - R )/LAMDA )                  00000480
C                             0              0                          00000490
C                                                                        00000500
C                   RHO  = ATMOSPHERIC DENSITY AT THE REFERENCE R        00000510
C                      0                          0                     00000520
C                     R  = MAGNITUDE OF THE SPACECRAFT POSITION VECTOR   00000530
C                     R  = REFERENCE POSITION (TAKEN TO BE THE MEAN      00000540
C                      0       RADIUS OF THE CENTRAL BODY IN THIS        00000550
C                   LAMDA = ATMOSPHERIC SCALE HEIGHT                     00000560
C                                                                        00000570
C     CALLING SEQUENCE                                                   00000580
C         CALL AERO(ST,ASUBD)                                            00000590
C                                                                        00000600
C     CALLING SEQUENCE PARAMETERS                                        00000610
C         ST     I     ARRAY (6X1) CONTAINING THE SPACECRAFT POSITION   00000620
```

```
AERO                JSS1493.PRLLEL.ORBDET.FORT

C                       AND VELOCITY VECTORS                                    00000580
C         ASUBD  O      ARRAY (3X1) CONTAINING THE DRAG ACCELERATION            00000590
C                                                                               00000600
C     EXTERNAL REFERENCES                                                       00000610
C         NONE                                                                  00000620
C                                                                               00000630
C     ANALYSIS                                                                  00000640
C         CAPTAIN JEFFREY S. SHAVER, USAF                                       00000650
C                                                                               00000660
C     PROGRAMMING                                                               00000670
C         CAPTAIN JEFFREY S. SHAVER, USAF                                       00000680
C                                                                               00000690
      IMPLICIT REAL*8(A-H,O-Z)                                                  00000700
C                                                                               00000710
C * * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * * * 00000720
C                                                                               00000730
      DIMENSION       ST(6)   ,   ASUBD(3)  ,   VREL(3)                         00000740
C                                                                               00000750
C * * * * * * * * * *           COMMON BLOCKS         * * * * * * * * * * * * 00000760
C                                                                               00000770
      COMMON/DRGPAR/  CSUBD   ,   SCAREA    ,   SCMASS    ,   RHOZRO    ,00000780
     *                RZERO   ,   SCLHGT    ,   IDRAG                         00000790
C                                                                               00000800
      COMMON/ERTSHP/  REQ     ,   WE        ,   FINV      ,   SPHINF        00000810
C                                                                               00000820
C     COMPUTE DENSITY AT THE ALTITUDE R - R                                     00000830
C                                        O                                      00000840
C                                                                               00000850
      RSQ = ST(1)*ST(1) + ST(2)*ST(2) + ST(3)*ST(3)                             00000860
      R   = DSQRT(RSQ)                                                          00000870
      COFEXP = -(R - RZERO)/SCLHGT                                              00000880
      RHO = RHOZRO * DEXP(COFEXP)                                               00000890
C                                                                               00000900
C     COMPUTE THE BALLISTIC COEFFICIENT                                         00000910
C                                                                               00000920
      BALCOF = CSUBD * (SCAREA/SCMASS)                                          00000930
C                                                                               00000940
C     COMPUTE THE VELOCITY OF THE SPACECRAFT RELATIVE TO THE ROTATING           00000950
C     ATMOSPHERE                                                                00000960
C                                                                               00000970
      VREL(1) = ST(4) + WE * ST(2)                                              00000980
      VREL(2) = ST(5) - WE * ST(1)                                              00000990
      VREL(3) = ST(6)                                                           00001000
      VRELSQ  = VREL(1)**2 + VREL(2)**2 + VREL(3)**2                            00001010
      VRELMG  = DSQRT(VRELSQ)                                                   00001020
C                                                                               00001030
C     COMPUTE THE DRAG ACCELERATION/UNIT MASS                                   00001040
C                                                                               00001050
      FACTOR = (BALCOF * RHO * VRELMG)/2.D0                                     00001060
      DO  10   I = 1,3                                                          00001070
      ASUBD(I) = -FACTOR * VREL(I)                                              00001080
   10 CONTINUE                                                                  00001090
      RETURN                                                                    00001100
      END                                                                       00001110

******** END OF MEMBER AERO       111 RECORDS ********
```

AKTOAE JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE AKTOAE(AK,RETRO,AE)                                    00000010
C                                                                       00000020
C     VERSION OF 08/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO CONVERT KEPLERIAN ELEMENTS TO EQUINOCTIAL ELEMENTS         00000060
C                                                                       00000070
C     METHOD                                                            00000080
C         THIS SUBROUTINE USES THE DEFINITIONS OF THE EQUINOCTIAL       00000090
C         ELEMENTS DOCUMENTED IN APPENDIX A OF THE FOLLOWING REFERENCE: 00000100
C                                                                       00000110
C         (1)  "A RECURSIVELY FORMULATED FIRST-ORDER SEMIANALYTIC       00000120
C         ARTIFICIAL SATELLITE THEORY BASED ON THE GENERALIZED METHOD   00000130
C         OF AVERAGING, VOLUME1:  THE GENERALIZED METHOD OF AVERAGING   00000140
C         APPLIED TO THE ARTIFICIAL SATELLITE PROBLEM,"                 00000150
C         CSC/TR-77/6010, NOVEMBER 1977.                                00000160
C                                                                       00000170
C     CALLING SEQUENCE                                                  00000180
C         CALL AKTOAE(AK,RETRO,AE)                                      00000190
C                                                                       00000200
C     CALLING SEQUENCE PARAMETERS                                       00000210
C         AK      I    ARRAY (6X1) CONTAINING THE KEPLERIAN ELEMENTS    00000220
C         RETRO   O    RETROGRADE FACTOR                                00000230
C         AE      O    ARRAY (6X1) CONTAINING THE EQUINOCTIAL ELEMENTS  00000240
C                                                                       00000250
C     EXTERNAL REFERENCES                                               00000260
C         NONE                                                          00000270
C                                                                       00000280
C     ANALYSIS                                                          00000290
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000300
C                                                                       00000310
C     PROGRAMMING                                                       00000320
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000330
C                                                                       00000340
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000350
C                                                                       00000360
      IMPLICIT REAL*8(A-H,O-Z)                                          00000370
      DIMENSION       AK(6)     ,     AE(6)                             00000380
      PI = 3.1415926535897932D0                                         00000390
C                                                                       00000400
C     COMPUTE EQUINOCTIAL ELEMENT A                                     00000410
C                                                                       00000420
      AE(1) = AK(1)                                                     00000430
C                                                                       00000440
C     COMPUTE RETROGRADE FACTOR FROM INCLINATION                        00000450
C                                                                       00000460
      IF (AK(3).GE.0.D0.AND.AK(3).LE.PI/2.D0)     RETRO =  1.D0         00000470
      IF (AK(3).GT.PI/2.D0.AND.AK(3).LE.PI)       RETRO = -1.D0         00000480
C                                                                       00000490
C     COMPUTE EQUINOCTIAL ELEMENTS H AND K                              00000500
C                                                                       00000510
      AE(2) = AK(2) * DSIN(AK(4) + RETRO * AK(5))                       00000520
      AE(3) = AK(2) * DCOS(AK(4) + RETRO * AK(5))                       00000530
C                                                                       00000540
C     COMPUTE EQUINOCTIAL ELEMENT P                                     00000550
C                                                                       00000560
      IF (RETRO.GT.0.D0)      AE(4) = DTAN(AK(3)/2.D0) * DSIN(AK(5))    00000570
```

```
AKTOAE              JSS1493.PRLLEL.ORBDET.FORT

      IF (RETRO.LT.0.D0)      AE(4) = DCOTAN(AK(3)/2.D0) * DSIN(AK(5))  00000580
C                                                                       00000590
C     COMPUTE EQUINOCTIAL ELEMENT Q                                     00000600
C                                                                       00000610
      IF (RETRO.GT.0.D0)      AE(5) = DTAN(AK(3)/2.D0) * DCOS(AK(5))    00000620
      IF (RETRO.LT.0.D0)      AE(5) = DCOTAN(AK(3)/2.D0) * DCOS(AK(5))  00000630
C                                                                       00000640
C     COMPUTE EQUINOCTIAL ELEMENT LAMDA                                 00000650
C                                                                       00000660
      AE(6) = AK(6) + AK(4) + RETRO * AK(5)                             00000670
      RETURN                                                            00000680
      END                                                               00000690

******** END OF MEMBER AKTOAE      69 RECORDS ********
```

```
      SUBROUTINE CGINTG(X,C,G)                                          00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE VALUE OF THE COST FUNCTION AND ITS GRADIENT    00000060
C         FOR THE ESTIMATION PROBLEM.  THE COST FUNCTION IS THE SUM     00000070
C         OF THE SQUARES OF THE WEIGHTED OBSERVATION RESIDUALS.  THE    00000080
C         COMPUTED OBSERVABLES ARE FOUND BY DIRECT NUMERICAL INTE-      00000090
C         GRATION.                                                      00000100
C                                                                       00000110
C     CALLING SEQUENCE                                                  00000120
C         CALL CGINTG(X,C,G)                                            00000130
C                                                                       00000140
C     CALLING SEQUENCE PARAMETERS                                       00000150
C         X         I   ARRAY (6X1) CONTAINING THE CURRENT ESTIMATE OF  00000160
C                       THE SOLVE-FOR VECTOR                          00000170
C         C         O   VALUE OF THE COST FUNCTIONAL FOR THIS SOLVE-FOR 00000180
C                       VECTOR                                        00000190
C         G         O   ARRAY (6X1) CONTAINING THE COMPONENTS OF THE  00000200
C                       GRADIENT OF THE COST FUNCTIONAL               00000210
C                                                                       00000220
C     EXTERNAL REFERENCES                                               00000230
C         SUBROUTINES                                                   00000240
C              CMPTOB                                                   00000250
C              ELMRPT                                                   00000260
C              PXTSLV                                                   00000270
C                                                                       00000280
C         COMMON BLOCKS                                                 00000290
C              /ERTSHP/                                                 00000300
C              /ICNTRL/                                                 00000310
C              /OBSDAT/                                                 00000320
C              /STAPAR/                                                 00000330
C              /TIMER /                                                 00000340
C                                                                       00000350
C     ANALYSIS                                                          00000360
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000370
C                                                                       00000380
C     PROGRAMMING                                                       00000390
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000400
C                                                                       00000410
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000420
C                                                                       00000430
      IMPLICIT REAL*8(A-H,O-Z)                                          00000440
C                                                                       00000450
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000460
C                                                                       00000470
      DIMENSION       X(6)            ,  G(6)          ,  STA(6)      , 00000480
     1                EL(26)                                           00000490
C                                                                       00000500
C * * * * * * * * *         COMMOM BLOCKS           * * * * * * * * * * 00000510
C                                                                       00000520
      COMMON/ERTSHP/  REQ             ,  WE            ,  FINV        , 00000530
     1                SPHINF          ,  GM                           00000540
C                                                                       00000550
      COMMON/ICNTRL/  IRNTYP          ,  IVINIT        ,  IVINTG      , 00000560
     1                ISLVFR          ,  NSLVFR        ,  IPTLS       , 00000570
```

```
     2               IJ2RTS      , IRVPAR      , IPARRV          , 00000580
     3               IPRNT1      , IPRNT2      , IPRNT3          , 00000590
     4               IPRNT4      , IPRNT5      , IOUT            , 00000600
     5               ICGCMP      , ISCID       , NPAR              00000610
C                                                                  00000620
      COMMON/OBSDAT/ OBSFIL(100,6)           ,     OBSPAR(100,6)  , 00000630
     1               OBSCMP(100,6)           ,     RES(100)       , 00000640
     2               CMTXMS(100)             ,     NOBS             00000650
C                                                                  00000660
      COMMON/STAPAR/ PXDAEO(6,6) , PXDAKO(6,6) , PXDXO(6,6)        00000670
C                                                                  00000680
      COMMON/TIMER / T0          , TSPAN       , TOUT            , 00000690
     1               H           , TINC                            00000700
C                                                                  00000710
C     COMPUTE THE PREDICTED OBSERVABLES AT THE OBSERVATION TIMES BY 00000720
C     DIRECT NUMERICAL INTEGRATION                                 00000730
C                                                                  00000740
      CALL CMPTOB(X)                                               00000750
C                                                                  00000760
C     INITIALIZE OBSERVATION POINTER AND COST TO ZERO              00000770
C                                                                  00000780
      IOBSPT = 0                                                   00000790
      C = 0.D0                                                     00000800
C                                                                  00000810
C     INCREMENT OBSERVATION POINTER; RETRIEVE THE CURRENT OBSERVATION 00000820
C     TIME, OBSERVATION TYPE AND THE ASSOCIATED POSITION AND VELOCITY 00000830
C                                                                  00000340
   10 IOBSPT = IOBSPT + 1                                          00000850
      TOBS   = OBSFIL(IOBSPT,1) * 60.D0                            00000860
      IOTYPE = OBSFIL(IOBSPT,2)                                    00000870
      OO     = OBSFIL(IOBSPT,6)                                    00000880
      DO  20   I = 1,6                                             00000890
      STA(I) = OBSCMP(IOBSPT,I)                                    00000900
   20 CONTINUE                                                     00000910
C                                                                  00000920
C     COMPUTE THE CURRENT OBSERVATION RESIDUAL AND ITS CONTRIBUTION 00000930
C     TO THE WEIGHTED SQUARED ERROR COST FUNCTIONAL                00000940
C                                                                  00000950
      RES(IOBSPT) = OO - STA(IOTYPE)                               00000960
      C = C + RES(IOBSPT) * CMTXMS(IOBSPT) * RES(IOBSPT)           00000970
      IF (ICGCMP.EQ.4.AND.IOBSPT.LT.NOBS)    GO TO 10              00000980
      IF (ICGCMP.EQ.4)                       RETURN                00000990
C                                                                  00001000
C     COMPUTE THE ELEMENTS ASSOCIATED WITH THE PREDICTED STATE AND 00001010
C     COMPUTE THE APPROPRIATE OBSERVATION PARTIAL DERIVATIVES      00001020
C                                                                  00001030
      CALL ELMRPT(STA,T0,TOBS,GM,2,0,EL)                           00001040
      ISAVE  = IPRNT2                                              00001050
      IPRNT2 = 0                                                   00001060
      CALL PXTSLV(STA,EL,TOBS,T0)                                  00001070
      IPRNT2 = ISAVE                                               00001080
      DO  30   I = 1,6                                             00001090
      IF (ISLVFR.EQ.1)        OBSPAR(IOBSPT,I) =  PXDXO(IOTYPE,I)  00001100
      IF (ISLVFR.EQ.2)        OBSPAR(IOBSPT,I) = PXDAKO(IOTYPE,I)  00001110
      IF (ISLVFR.EQ.3)        OBSPAR(IOBSPT,I) = PXDAEO(IOTYPE,I)  00001120
   30 CONTINUE                                                     00001130
C                                                                  00001140
```

```
CGINTG              JSS1493.PRLLEL.ORBDET.FORT

C     IF THERE ARE MORE OBSERVATIONS TO PROCESS, RETURN TO START        00001150
C     OF LOOP                                                           00001160
C                                                                       00001170
      IF (IOBSPT.LT.NOBS)              GO TO 10                         00001180
C                                                                       00001190
C     AFTER ALL OBSERVATIONS HAVE BEEN PROCESSED, COMPUTE THE GRADIENT  00001200
C                                                                       00001210
      DO  50   I = 1,NSLVFR                                             00001220
      G(I) = 0.D0                                                       00001230
      DO  40   J = 1,NOBS                                               00001240
   40 G(I) = G(I) - 2.D0 * OBSPAR(J,I) * CMTXMS(J) * RES(J)             00001250
   50 CONTINUE                                                          00001260
      RETURN                                                            00001270
      END                                                               00001280

******** END OF MEMBER CGINTG     128 RECORDS ********
```

```
CGLINE              JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE CGLINE(X,C,G)                                          00000010
C                                                                       00000020
C     VERSION OF 09/15/79      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE VALUE OF THE COST FUNCTION AND ITS GRADIENT    00000060
C         FOR THE ESTIMATION PROBLEM.  THE COST FUNCTION IS THE SUM     00000070
C         OF THE SQUARES OF THE WEIGHTED OBSERVATION RESIDUALS.  THE    00000080
C         COMPUTED OBSERVABLES ARE FOUND BY APPLYING A LINEARIZATION    00000090
C         ASSUMPTION, SUCH THAT DIRECT NUMERICAL INTEGRATION OF THE     00000100
C         EQUATIONS OF MOTION IS NOT REQUIRED.                          00000110
C                                                                       00000120
C     CALLING SEQUENCE                                                  00000130
C         CALL CGLINE(X,C,G)                                            00000140
C                                                                       00000150
C     CALLING SEQUENCE PARAMETERS                                       00000160
C         X         I   ARRAY (6X1) CONTAINING THE CURRENT ESTIMATE OF  00000170
C                       THE SOLVE-FOR VECTOR                          00000180
C         C         O   VALUE OF THE COST FUNCTIONAL AT THIS SOLVE-FOR  00000190
C                       VECTOR                                        00000200
C         G         O   ARRAY (6X1) CONTAINING THE COMPONENTS OF THE    00000210
C                       GRADIENT OF THE COST FUNCTIONAL               00000220
C                                                                       00000230
C     EXTERNAL REFERENCES                                               00000240
C         SUBROUTINES                                                   00000250
C              ELMRPT                                                   00000260
C              M6666                                                    00000270
C              PAETAK                                                   00000280
C              PATA0                                                    00000290
C              PEQTX                                                    00000300
C              PXTEQ                                                    00000310
C              PXTSLV                                                   00000320
C                                                                       00000330
C         COMMON BLOCKS                                                 00000340
C              /EQPTLX/                                                 00000350
C              /ERTSHP/                                                 00000360
C              /ICNTRL/                                                 00000370
C              /OBSDAT/                                                 00000380
C              /STAPAR/                                                 00000390
C              /TIMER /                                                 00000400
C              /XEPOCH/                                                 00000410
C              /XESTMT/                                                 00000420
C              /XPTLEQ/                                                 00000430
C                                                                       00000440
C     ANALYSIS                                                          00000450
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000460
C                                                                       00000470
C     PROGRAMMING                                                       00000480
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000490
C                                                                       00000500
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000510
C                                                                       00000520
      IMPLICIT REAL*8(A-H,O-Z)                                          00000530
C                                                                       00000540
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000550
C                                                                       00000560
      DIMENSION       X(6)         ,  G(6)         ,  DELRV(6)        , 00000570
```

```
CGLINE              JSS1493.PRLLEL.ORBDET.FORT

     1               DSLVFR(6)    ,  STA(6)       ,  STAPRM(6)       , 00000580
     2               PXDX0(6,6)   ,  PXDAE0(6,6)  ,  PXDAK0(6,6)     , 00000590
     3               PAWRA0(6,6)  ,  PAEDAK(6,6)  ,  EL(26)            00000600
C                                                                      00000610
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000620
C                                                                      00000630
      COMMON/EQPTLX/  PEQDR(6,3)   ,  PEQDV(6,3)   ,  PEQDX(6,6)       00000640
C                                                                      00000650
      COMMON/ERTSHP/  REQ          ,  WE           ,  FINV            , 00000660
     1                SPHINF       ,  GM                              00000670
C                                                                      00000680
      COMMON/ICNTRL/  IRNTYP       ,  IVINIT       ,  IVINTG          , 00000690
     1                ISLVFR       ,  NSLVFR       ,  IPTLS           , 00000700
     2                IJ2RTS       ,  IRVPAR       ,  IPARRV          , 00000710
     3                IPRNT1       ,  IPRNT2       ,  IPRNT3          , 00000720
     4                IPRNT4       ,  IPRNT5       ,  IOUT            , 00000730
     5                ICGCMP       ,  ISCID        ,  NPAR            , 00000740
C                                                                      00000750
      COMMON/OBSDAT/  OBSFIL(100,6)          ,        OBSPAR(100,6)   , 00000760
     1                OBSCMP(100,6)          ,        RES(100)        , 00000770
     2                CMTXMS(100)            ,        NOBS              00000780
C                                                                      00000790
      COMMON/STAPAR/  PXDAE0(6,6)  ,  PXDAK0(6,6)  ,  PXDX0(6,6)       00000800
C                                                                      00000810
      COMMON/TIMER /  T0           ,  TSPAN        ,  TOUT            , 00000820
     1                H            ,  TINC                            00000830
C                                                                      00000840
      COMMON/XEPOCH/  EPST(6)      ,  EPAK(26)     ,  EPAE(26)        , 00000850
     1                PEQDK0(6,6)  ,  PEQDX0(6,6)                     00000860
C                                                                      00000870
      COMMON/XESTMT/  XAP(6)       ,  SCLFAC(6)    ,  EPSILN          , 00000880
     1                ALFA1        ,  STPSZ        ,  C1DIM           , 00000890
     2                CMTXAP(6,6)  ,  XMIN(6)                         00000900
C                                                                      00000910
      COMMON/XPTLEQ/  PRDEQ(3,6)   ,  PVDEQ(3,6)   ,  PXDEQ(6,6)       00000920
C                                                                      00000930
C     INITIALIZE OBSERVATION POINTER AND COST TO ZERO                  00000940
C                                                                      00000950
      IOBSPT = 0                                                       00000960
      C = 0.D0                                                         00000970
C                                                                      00000980
C     INCREMENT OBSERVATION POINTER; RETRIEVE THE CURRENT OBSERVATION  00000990
C     TIME, OBSERVATION TYPE AND THE ASSOCIATED POSITION AND VELOCITY  00001000
C     COMPUTE THE DELTA IN THE SOLVE-FOR VECTOR; INITIALIZE THE DELTA  00001010
C     IN THE CURRENT STATE TO ZERO                                     00001020
C                                                                      00001030
   10 IOBSPT = IOBSPT + 1                                              00001040
      TOBS   = OBSFIL(IOBSPT,1) * 60.D0                                00001050
      IOTYPE = OBSFIL(IOBSPT,2)                                        00001060
      OO     = OBSFIL(IOBSPT,6)                                        00001070
      DO  20   I = 1,6                                                 00001080
      STA(I) = OBSCMP(IOBSPT,I)                                        00001090
      DSLVFR(I) = X(I) - XAP(I)                                        00001100
      DELRV(I)  = 0.D0                                                 00001110
   20 CONTINUE                                                         00001120
C                                                                      00001130
C     COMPUTE THE ELEMENTS ASSOCIATED WITH THE CURRENT STATE; COMPUTE  00001140
```

```
C     THE APPROPRIATE PARTIAL DERIVATIVES OF CURRENT STATE WITH RESPECT 00001150
C     TO THE SOLVE-FOR PARAMETERS; COMPUTE THE DELTA IN THE CURRENT     00001160
C     STATE DUE TO THE DELTA IN THE SOLVE-FOR PARAMETERS (LINEARIZATION 00001170
C     ASSUMPTION)                                                       00001180
C                                                                       00001190
      CALL ELMRPT(STA,T0,TOBS,GM,2,0,EL)                                00001200
      ISAVE  = IPRNT2                                                   00001210
      IPRNT2 = 0                                                        00001220
      CALL PXTSLV(STA,EL,TOBS,T0,ISLVFR)                                00001230
      IPRNT2 = ISAVE                                                    00001240
      DO  40   I = 1,6                                                  00001250
      DO  30   J = 1,6                                                  00001260
      IF (ISLVFR.EQ.1)       DELRV(I) = DELRV(I) +  PXDX0(I,J)*DSLVFR(J)00001270
      IF (ISLVFR.EQ.2)       DELRV(I) = DELRV(I) + PXDAK0(I,J)*DSLVFR(J)00001280
      IF (ISLVFR.EQ.3)       DELRV(I) = DELRV(I) + PXDAE0(I,J)*DSLVFR(J)00001290
   30 CONTINUE                                                          00001300
      STAPRM(I) = STA(I) + DELRV(I)                                     00001310
   40 CONTINUE                                                          00001320
C                                                                       00001330
C     COMPUTE THE CURRENT OBSERVATION RESIDUAL AND ITS CONTRIBUTION TO  00001340
C     THE WEIGHTED SQUARED ERROR COST FUNCTIONAL                        00001350
C                                                                       00001360
      RES(IOBSPT) = OO - STAPRM(IOTYPE)                                 00001370
      C = C + RES(IOBSPT) * CMTXMS(IOBSPT) * RES(IOBSPT)                00001380
      IF (ICGCMP.EQ.5.AND.IOBSPT.LT.NOBS)    GO TO 10                   00001390
      IF (ICGCMP.EQ.5)                       RETURN                     00001400
C                                                                       00001410
C     RECOMPUTE THE OBSERVATION PARTIAL DERIVATIVES USING THE VARIED    00001420
C     CURRENT STATE AND THE VARIED SOLVE-FOR VECTOR, IF REQUIRED        00001430
C                                                                       00001440
      IF (ICGCMP.EQ.2)                  GO TO 80                        00001450
      CALL ELMRPT(STAPRM,T0,TOBS,GM,2,0,EL)                             00001460
      CALL PXTEQ(STAPRM,EL,0)                                           00001470
      CALL PATA0(EL,TOBS,T0,IJ2RTS,PAWRAO)                              00001480
      CALL M6666(PXDEQ,PAWRAO,PXDAE0)                                   00001490
      IF (ISLVFR.NE.1)                  GO TO 50                        00001500
      CALL ELMRPT(X,T0,T0,GM,2,0,EL)                                    00001510
      CALL PEQTX(X,EL,0)                                                00001520
      CALL M6666(PXDAE0,PEQDX,PXDX0)                                    00001530
      GO TO 60                                                          00001540
   50 IF (ISLVFR.NE.2)                  GO TO 60                        00001550
      CALL PAETAK(X,PAEDAK)                                             00001560
      CALL M6666(PXDAE0,PAEDAK,PXDAK0)                                  00001570
   60 DO  70   I = 1,6                                                  00001580
      IF (ISLVFR.EQ.1)       OBSPAR(IOBSPT,I) =  PXDX0(IOTYPE,I)       00001590
      IF (ISLVFR.EQ.2)       OBSPAR(IOBSPT,I) = PXDAK0(IOTYPE,I)       00001600
      IF (ISLVFR.EQ.3)       OBSPAR(IOBSPT,I) = PXDAE0(IOTYPE,I)       00001610
   70 CONTINUE                                                          00001620
      GO TO 100                                                         00001630
C                                                                       00001640
C     IF IT IS NOT DESIRED TO RECOMPUTE THE OBSERVATION PARTIAL         00001650
C     DERIVATIVES, USE THOSE FROM THE A PRIORI TRAJECTORY               00001660
C                                                                       00001670
   80 DO  90   I = 1,6                                                  00001680
      IF (ISLVFR.EQ.1)       OBSPAR(IOBSPT,I) =  PXDX0(IOTYPE,I)       00001690
      IF (ISLVFR.EQ.2)       OBSPAR(IOBSPT,I) = PXDAK0(IOTYPE,I)       00001700
      IF (ISLVFR.EQ.3)       OBSPAR(IOBSPT,I) = PXDAE0(IOTYPE,I)       00001710
```

```
CGLINE              JSS1493.PRLLEL.ORBDET.FORT

   90 CONTINUE                                                                  00001720
C                                                                               00001730
C     IF MORE OBSERVATIONS REMAIN TO BE PROCESSED, RETURN TO THE START          00001740
C     OF THE OBSERVATION PROCESSING LOOP                                        00001750
C                                                                               00001760
  100 IF (IOBSPT.LT.NOBS)                GO TO 10                               00001770
C                                                                               00001780
C     AFTER ALL OBSERVATIONS HAVE BEEN PROCESSED, COMPUTE THE GRADIENT          00001790
C                                                                               00001800
      DO  120  I = 1,NSLVFR                                                     00001810
      G(I) = 0.D0                                                               00001820
      DO  110  J = 1,NOBS                                                       00001830
  110 G(I) = G(I) - 2.D0 * OBSPAR(J,I) * CMTXMS(J) * RES(J)                     00001840
  120 CONTINUE                                                                  00001850
C                                                                               00001860
      RETURN                                                                    00001870
C                                                                               00001880
      END                                                                       00001890

******** END OF MEMBER CGLINE     189 RECORDS ********
```

```
CHBINT              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK CHBINT   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/CHBINT/     T(70)     ,     X(70)     ,     TX(70,70)     00000120
C                                                                       00000130
C                                                                       00000140
C                    #LOCA-                                             00000150
C             DIMEN-  TION                                              00000160
C  VARIABLE    SION   (REAL)              DESCRIPTION                   00000170
C  ________   _____   _____   ____________________________________      00000180
C                                                                       00000190
C  T             70        1   ARRAY (70X1) CONTAINING THE N+1 OR-     00000200
C                              THOGONALITY POINTS IN THE TRUE INDEPEN-  00000210
C                              DENT VARIABLE T.                         00000220
C  X             70       71   ARRAY (70X1) CONTAINING THE N+1 OR-     00000230
C                              THOGONALITY POINTS IN THE NORMALIZED     00000240
C                              INDEPENDENT VARIABLE X.                  00000250
C  TX         70,70      141   ARRAY (70X70) CONTAINING THE N+1        00000260
C                              CHEBYSHEV POLYNOMIALS EVALUATED AT THE   00000270
C                              N+1 ORTHOGONALITY POINTS.                00000280
C                                                                       00000290
      DATA T      / 70*0.D0     /                                       00000300
C                                                                       00000310
      DATA X      / 70*0.D0     /                                       00000320
C                                                                       00000330
      DATA TX     / 4900*0.D0   /                                       00000340
C                                                                       00000350
      END

******** END OF MEMBER CHBINT      35 RECORDS ********
```

```
CMPTOB              JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE CMPTOB(XSLV)                                                00000010
C                                                                            00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                          00000030
C                                                                            00000040
C     PURPOSE                                                                00000050
C         TO COMPUTE THE PREDICTED OBSERVABLES IN THE ESTIMATION             00000060
C         PROBLEM, GIVEN THE GUESS FOR THE SOLVE-FOR VECTOR, XSLV.           00000070
C                                                                            00000080
C     CALLING SEQUENCE                                                       00000090
C         CALL CMPTOB(XSLV)                                                  00000100
C                                                                            00000110
C     CALLING SEQUENCE PARAMETERS                                            00000120
C         XSLV     I   ARRAY (6X1) CONTAINING THE CURRENT ESTIMATE OF        00000130
C                      THE SOLVE-FOR VECTOR X                                00000140
C                                                                            00000150
C     EXTERNAL REFERENCES                                                    00000160
C         SUBROUTINES                                                        00000170
C              AKTOAE                                                        00000180
C              CSRSTA                                                        00000190
C              ELMRPT                                                        00000200
C              EQELPV                                                        00000210
C              PAETAK                                                        00000220
C              PEQTX                                                         00000230
C              VTEST                                                         00000240
C                                                                            00000250
C         COMMON BLOCKS                                                      00000260
C              /EQPTLX/                                                      00000270
C              /ERTSHP/                                                      00000280
C              /ICNTRL/                                                      00000290
C              /NPCNTL/                                                      00000300
C              /OBSDAT/                                                      00000310
C              /STATE /                                                      00000320
C              /TIMER /                                                      00000330
C              /XEPOCH/                                                      00000340
C                                                                            00000350
C     ANALYSIS                                                               00000360
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00000370
C                                                                            00000380
C     PROGRAMMING                                                            00000390
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00000400
C                                                                            00000410
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * *   00000420
C                                                                            00000430
      IMPLICIT REAL*8(A-H,O-Z)                                               00000440
C                                                                            00000450
C * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * *   00000460
C                                                                            00000470
      DIMENSION       XSLV(6)        ,  ELDUM(6)        ,  STA(6)     ,     00000480
     1                POSVEL(6)                                             00000490
C                                                                            00000500
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * *   00000510
C                                                                            00000520
      COMMON/EQPTLX/  PEQDR(6,3)  ,  PEQDV(6,3)  ,  PEQDX(6,6)             00000530
C                                                                            00000540
      COMMON/ERTSHP/  REQ         ,  WE          ,  FINV              ,     00000550
     1                SPHINF      ,  GM                                    00000560
C                                                                            00000570
```

```
      COMMON/ICNTRL/  IRNTYP      ,  IVINIT      ,  IVINTG          , 00000580
     1                ISLVFR      ,  NSLVFR      ,  IPTLS           , 00000590
     2                IJ2RTS      ,  IRVPAR      ,  IPARRV          , 00000600
     3                IPRNT1      ,  IPRNT2      ,  IPRNT3          , 00000610
     4                IPRNT4      ,  IPRNT5      ,  IOUT            , 00000620
     5                ICGCMP      ,  ISCID       ,  NPAR              00000630
C                                                                     00000640
      COMMON/NPCNTL/  MORD        ,  NSUB        ,  IEQSET          , 00000650
     1                ITERM                                          00000660
C                                                                     00000670
      COMMON/OBSDAT/  OBSFIL(100,6)       ,          OBSPAR(100,6)  , 00000680
     1                OBSCMP(100,6)       ,          RES(100)       , 00000690
     2                CMTXMS(100)         ,          NOBS             00000700
C                                                                     00000710
      COMMON/STATE /  S0(6)       ,  STK(70,6)   ,  STK1(70,6)         00000720
C                                                                     00000730
      COMMON/TIMER /  T0          ,  TSPAN       ,  TOUT            , 00000740
     1                H           ,  TINC                             00000750
C                                                                     00000760
      COMMON/XEPOCH/  EPST(6)     ,  EPAK(26)    ,  EPAE(26)        , 00000770
     1                PEODKO(6,6) ,  PEODXO(6,6)                      00000780
C * *                                                                 00000790
C     HARD-CODE THE RETROGRADE FACTOR TO BE 1                         00000800
C * *                                                                 00000810
      RETRO = 1.D0                                                    00000820
C * *                                                                 00000830
C     REMOVE ABOVE CODE IN THE GENERAL IMPLEMENTATION                 00000840
C * *                                                                 00000850
C                                                                     00000860
C                                                                     00000870
C     IF THE TYPE OF SOLVE-FOR PARAMETERS (ISLVFR) IS THE SAME AS THE 00000880
C     TYPE OF INTEGRATION VARIABLE (IVINTG), THEN THE ESTIMATE OF THE 00000890
C     SOLVE-FOR VECTOR IS ALSO THE INITIAL CONDITION VECTOR FOR THE   00000900
C     NUMERICAL INTEGRATION PROCESS                                   00000910
C                                                                     00000920
      IF (ISLVFR.NE.IVINTG)             GO TO 20                      00000930
      DO  10   I = 1,6                                                00000940
      S0(I) = XSLV(I)                                                 00000950
   10 CONTINUE                                                        00000960
      GO TO 100                                                       00000970
C                                                                     00000980
C     OTHERWISE, THE PROPER INITIAL CONDITION VECTOR MUST BE COMPUTED, 00000990
C     BASED ON THE ESTIMATE OF THE SOLVE-FOR VECTOR                   00001000
C                                                                     00001010
C     IF ISLVFR = 1, THE ESTIMATE OF THE SOLVE-FOR VECTOR IS THE EPOCH 00001020
C     STATE VECTOR (POSITION AND VELOCITY)                            00001030
C                                                                     00001040
   20 IF (ISLVFR.NE.1)                  GO TO 40                      00001050
      DO  30   I = 1,6                                                00001060
      EPST(I) = XSLV(I)                                               00001070
   30 CONTINUE                                                        00001080
      GO TO 60                                                        00001090
C                                                                     00001100
C     IF ISLVFR = 2, THE ESTIMATE OF THE SOLVE-FOR VECTOR IS THE VECTOR 00001110
C     OF EPOCH KEPLERIAN ORBITAL ELEMENTS                             00001120
C                                                                     00001130
   40 IF (ISLVFR.NE.2)                  GO TO 50                      00001140
```

```
      CALL AKTOAE(XSLV,RETRO,ELDUM)                                          00001150
      CALL EQELPV(ELDUM,RETRO,GM,EPST)                                       00001160
      GO TO 60                                                               00001170
C                                                                            00001180
C     IF ISLVFR = 3, THE ESTIMATE OF THE SOLVE-FOR VECTOR IS THE VECTOR      00001190
C     OF EPOCH EQUINOCTIAL ELEMENTS                                          00001200
C                                                                            00001210
   50 CALL EQELPV(XSLV,RETRO,GM,EPST)                                        00001220
C                                                                            00001230
C     DETERMINE PROPER EPOCH CONDITIONS FOR INTEGRATION                      00001240
C                                                                            00001250
   60 CALL ELMRPT(EPST,T0,T0,GM,1,0,EPAK)                                    00001260
      CALL ELMRPT(EPST,T0,T0,GM,2,0,EPAE)                                    00001270
      CALL PAETAK(EPAK,PEQDKO)                                               00001280
      CALL PEQTX(EPST,EPAE,0)                                                00001290
      DO  80   I = 1,6                                                       00001300
      DO  70   J = 1,6                                                       00001310
   70 PEQDXO(I,J) = PEQDX(I,J)                                               00001320
   80 CONTINUE                                                               00001330
      DO  90   I = 1,6                                                       00001340
      IF (IVINTG.EQ.1)                 S0(I) = EPST(I)                       00001350
      IF (IVINTG.EQ.2)                 S0(I) = EPAK(I)                       00001360
      IF (IVINTG.EQ.3)                 S0(I) = EPAE(I)                       00001370
   90 CONTINUE                                                               00001380
C                                                                            00001390
C     INITIALIZE INTEGRATION TIMES AND AUXILIARY QUANTITIES; SAVE            00001400
C     PARTIALS FLAG AND PRINT FLAGS                                          00001410
C                                                                            00001420
  100 TL = T0                                                                00001430
      TU = H                                                                 00001440
      N1 = NSUB + 1                                                          00001450
      IS1  = IPTLS                                                           00001460
      IS2  = IPRNT1                                                          00001470
      IS3  = IPRNT2                                                          00001480
      IS4  = IPRNT3                                                          00001490
      IPTLS  = 0                                                             00001500
      IPRNT1 = 0                                                             00001510
      IPRNT2 = 0                                                             00001520
      IPRNT3 = 0                                                             00001530
      IOBSPT = 1                                                             00001540
C                                                                            00001550
C     TAKE AN INTEGRATION STEP                                               00001560
C                                                                            00001570
  110 CALL VTEST(TL,TU,GM)                                                   00001580
C                                                                            00001590
C     INTERPOLATE AND/OR COMPUTE THE POSITION AND VELOCITY AT EACH           00001600
C     OBSERVATION TIME CONTAINED IN THIS INTEGRATION STEP                    00001610
C                                                                            00001620
  120 TOBS = OBSFIL(IOBSPT,1) * 60.D0                                        00001630
      IF (TOBS.GE.TL.AND.TOBS.LE.TU)    GO TO 130                            00001640
      GO TO 180                                                              00001650
  130 CALL CSRSTA(STA,TU,TOBS,H,1)                                           00001660
      GO TO (160,140,150) , IVINTG                                           00001670
  140 CALL AKTOAE(STA,RETRO,ELDUM)                                           00001680
      CALL EQELPV(ELDUM,RETRO,GM,POSVEL)                                     00001690
      GO TO 160                                                              00001700
  150 CALL EQELPV(STA,RETRO,GM,POSVEL)                                       00001710
```

```
  160 DO  170  I = 1,6                                                    00001720
      IF (IVINTG.EQ.1)                OBSCMP(IOBSPT,I) = STA(I)         00001730
      IF (IVINTG.NE.1)                OBSCMP(IOBSPT,I) = POSVEL(I)      00001740
  170 CONTINUE                                                            00001750
C                                                                         00001760
C     UPDATE THE OBSERVATION POINTER AND RETURN TO THE INTERPOLATOR       00001770
C                                                                         00001780
      IOBSPT = IOBSPT + 1                                                 00001790
      GO TO 120                                                           00001800
C                                                                         00001810
C     IF ALL OBSERVATIONS HAVE BEEN COMPUTED, THEN RETURN; OTHERWISE,     00001820
C     UPDATE INTEGRATION VARIABLES AND TAKE ANOTHER STEP                  00001830
C                                                                         00001840
  180 IF (IOBSPT.GT.NOBS)             GO TO 230                           00001850
      IF (TSPAN - TU)                 230,230,190                         00001860
  190 IF (DABS(TSPAN-TU).GE.H)        GO TO 200                           00001870
      TL = TU                                                             00001880
      TU = TSPAN                                                          00001890
      GO TO 210                                                           00001900
  200 TL = TU                                                             00001910
      TU = TU + H                                                         00001920
  210 DO  220  I = 1,6                                                    00001930
      S0(I) = STK1(N1,I)                                                  00001940
  220 CONTINUE                                                            00001950
      GO TO 110                                                           00001960
C                                                                         00001970
C     RESTORE PARTIALS FLAG AND PRINT FLAGS                               00001980
C                                                                         00001990
  230 IPTLS  = IS1                                                        00002000
      IPRNT1 = IS2                                                        00002010
      IPRNT2 = IS3                                                        00002020
      IPRNT3 = IS4                                                        00002030
      RETURN                                                              00002040
      END                                                                 00002050

******** END OF MEMBER CMPTOB     205 RECORDS ********
```

```
CNVRG                JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK CNVRG   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/CNVRG /     CON(6)     ,     CONKEP                      00000120
C                                                                       00000130
C                        #LOCA-                                         00000140
C              DIMEN-     TION                                         00000150
C  VARIABLE    SION      (REAL)                  DESCRIPTION           00000160
C  ________   _______   ________   ____________________________________00000170
C                                                                       00000180
C  CON             6          1   ARRAY (6X1) CONTAINING THE CONVERGENCE 00000190
C                                 CRITERION FOR EACH OF THE STATE EQUA- 00000200
C                                 TIONS BEING INTEGRATED.               00000210
C  CONKEP          1          7   CONVERGENCE CRITERION TO BE USED BY THE 00000220
C                                 SUBROUTINE 'KEPLR' IN SOLVING FOR THE 00000230
C                                 ECCENTRIC (OR TRUE) ANOMALY BY NEWTON'S 00000240
C                                 METHOD.                               00000250
C                                                                       00000260
      DATA CON    / 6*1.D-10    /                                       00000270
C                                                                       00000280
      DATA CONKEP / 1.D-10      /                                       00000290
C                                                                       00000300
      END                                                               00000310

******** END OF MEMBER CNVRG       31 RECORDS ********
```

```
COVAL                JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK COVAL    00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/COVAL /     A(70,6)     ,     BC(70,6)     ,     FORC(70,6)00000120
C                                                                       00000130
C                          #LOCA-                                       00000140
C               DIMEN-      TION                                        00000150
C  VARIABLE      SION       (REAL)                DESCRIPTION           00000160
C  ________    _______     _______   ____________________________________00000170
C                                                                       00000180
C         A      70,6           1    ARRAY (70X6) IN WHICH THE KTH COLUMN 00000190
C                                    CONTAINS THE M+1 COEFFICIENTS OF THE 00000200
C                                    CHEBYSHEV SERIES APPROXIMATING THE   00000210
C                                    INTEGRAND OF THE KTH EQUATION TO BE IN-00000220
C                                    TEGRATED (AS SPECIFIED BY THE VALUE OF 00000230
C                                    THE FLAG IEQSET IN COMMON/NPCNTL/).  00000240
C        BC      70,6         421    ARRAY (70X6) IN WHICH THE KTH COLUMN 00000250
C                                    CONTAINS THE COEFFICIENTS OF THE     00000260
C                                    CHEBYSHEV SERIES APPROXIMATING THE IN- 00000270
C                                    TEGRAL OF THE KTH INTEGRAND WHOSE CO- 00000280
C                                    EFFICIENTS WERE STORED IN THE KTH    00000290
C                                    COLUMN OF ARRAY A.                   00000300
C     FORC       70,6         841    ARRAY (70X6) IN WHICH THE KTH COLUMN 00000310
C                                    CONTAINS THE INTEGRAND OF THE KTH    00000320
C                                    EQUATION TO BE INTEGRATED, EVALUATED 00000330
C                                    AT THE N+1 POINTS OF ORTHOGONALITY.  00000340
C                                                                       00000350
      DATA A       / 420*0.D0     /                                     00000360
C                                                                       00000370
      DATA BC      / 420*0.D0     /                                     00000380
C                                                                       00000390
      DATA FORC    / 420*0.D0     /                                     00000400
C                                                                       00000410
      END                                                               00000420

******** END OF MEMBER COVAL       42 RECORDS ********
```

COVMTX JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE COVMTX(X,P,RSSPOS,RSSVEL,RMSPOS,RMSVEL)                00000010
C                                                                       00000020
C     VERSION OF 10/31/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO UPDATE THE A PRIORI COVARIANCE MATRIX OF THE SOLVE-FOR     00000060
C         PARAMETERS AFTER A PARALLEL VARIABLE METRIC FUNCTION          00000070
C         MINIMIZATION RUN                                              00000080
C                                                                       00000090
C     CALLING SEQUENCE                                                  00000100
C         CALL COVMTX(X,P,RSSPOS,RSSVEL,RMSPOS,RMSVEL)                  00000110
C                                                                       00000120
C     CALLING SEQUENCE PARAMETERS                                       00000130
C         X        I   ARRAY (6X1) CONTAINING THE MINIMIZING SOLVE-FOR  00000140
C                      VECTOR FROM THE PARALLEL VARIABLE METRIC         00000150
C                      ALGORITHM                                        00000160
C         P      I/O   ON INPUT, THIS ARRAY (6X6) CONTAINS THE A PRIORI 00000170
C                      COVARIANCE MATRIX OF THE ESTIMATION ERRORS.  ON  00000180
C                      OUTPUT, THIS ARRAY CONTAINS THE UPDATED COVA-    00000190
C                      RIANCE MATRIX OF THE ESTIMATION ERRORS, FORMED BY00000200
C                                                                       00000210
C                                  T      -1                            00000220
C                           P  = (F *W*F )                              00000230
C                            X                                          00000240
C                                                                       00000250
C                                                                       00000260
C     EXTERNAL REFERENCES                                               00000270
C         SUBROUTINES                                                   00000280
C              MATINV                                                   00000290
C                                                                       00000300
C         COMMON BLOCKS                                                 00000310
C              /ICNTRL/                                                 00000320
C              /OBSDAT/                                                 00000330
C                                                                       00000340
C     ANALYSIS                                                          00000350
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000360
C                                                                       00000370
C     PROGRAMMING                                                       00000380
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000390
C                                                                       00000400
C * * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000410
C                                                                       00000420
      IMPLICIT REAL*8(A-H,O-Z)                                          00000430
C                                                                       00000440
C * * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000450
C                                                                       00000460
      DIMENSION       X(6)          ,  P(6,6)         ,  FTWF(6,6)      , 00000470
     *                G(6)          ,  L(6)           ,  M(6)           , 00000480
     *                PINV(6,6)                                          00000490
C                                                                       00000500
C * * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000510
C                                                                       00000520
      COMMON/ICNTRL/  IRNTYP                ,           IVINIT        , 00000530
     *                IVINTG                ,           ISLVFR          00000540
C                                                                       00000550
      COMMON/OBSDAT/  OBSFIL(100,6)         ,           OBSPAR(100,6) , 00000560
     *                OBSCMP(100,6)         ,           RES(100)      , 00000570
```

```
     *                CMTXMS(100)          ,          NOBS                00000580
C                                                                          00000590
C     COMPUTE THE INVERSE OF THE A PRIORI COVARIANCE MATRIX OF THE         00000600
C     ESTIMATION ERRORS:  THIS CODE IS NOT CURRENTLY ACTIVATED             00000610
C                                                                          00000620
C     DO  20   I = 1,6                                                     00000630
C     DO  10   J = 1,6                                                     00000640
C     FTWF(I,J) = 0.D0                                                     00000650
C  10 PINV(I,J) = P(I,J)                                                   00000660
C  20 CONTINUE                                                             00000670
C     CALL MATINV(PINV,6,DET,L,M)                                          00000680
C                                                                          00000690
C                          T                                               00000700
C     COMPUTE THE ARRAY F   * W * F, WHERE                                 00000710
C                                                                          00000720
C                                    I                                     00000730
C                              ∂ F(X)  I                                   00000740
C                        F = ________  I                                   00000750
C                                    I                                     00000760
C                              ∂ X     I X = X                             00000770
C                                        MIN                               00000780
C                                                                          00000790
      CALL CGINTG(X,C,G)                                                   00000800
      DO  50   I = 1,6                                                     00000810
      DO  40   J = 1,6                                                     00000820
      DO  30   K = 1,NOBS                                                  00000830
   30 FTWF(I,J) = FTWF(I,J) + OBSPAR(K,I) * CMTXMS(K) * OBSPAR(K,J)        00000840
   40 CONTINUE                                                             00000850
   50 CONTINUE                                                             00000860
C                                                                          00000870
C                                                    T     -1  -1          00000880
C     COMPUTE THE NEW COVARIANCE MATRIX:  P  = (F WF+ P    )               00000890
C                                        X           X                     00000900
C                                                    0                     00000910
C                                                                          00000920
      DO  70   I = 1,6                                                     00000930
      DO  60   J = 1,6                                                     00000940
   60 P(I,J) = FTWF(I,J)                                                   00000950
   70 CONTINUE                                                             00000960
      CALL MATINV(P,6,DET,L,M)                                             00000970
C                                                                          00000980
C     IF THE SOLVE-FOR PARAMETERS ARE THE EPOCH POSITION AND VELOCITY      00000990
C     COMPONENTS, THEN COMPUTE THE RMS AND RSS POSITION AND VELOCITY       00001000
C     ERRORS                                                               00001010
C                                                                          00001020
      IF (ISLVFR.NE.1)                  RETURN                             00001030
      RSSPOS = P(1,1) + P(2,2) + P(3,3)                                    00001040
      RSSVEL = P(4,4) + P(5,5) + P(6,6)                                    00001050
      RMSPOS = DSQRT(RSSPOS)                                               00001060
      RMSVEL = DSQRT(RSSVEL)                                               00001070
C                                                                          00001080
      RETURN                                                               00001090
C                                                                          00001100
      END                                                                  00001110
```

******** END OF MEMBER COVMTX 111 RECORDS ********

CSER1 JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE CSER1(MORD,XPT,AM,CSER)                                00000010
C                                                                       00000020
C     VERSION OF 2/23/79      **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO EVALUATE A FUNCTION Y(X), REPRESENTED AS A SERIES OF       00000060
C         CHEBYSHEV POLYNOMIALS T(X) IN THE FORM                        00000070
C                                                                       00000080
C                          M                                            00000090
C     Y(X) = A /2 + SUM/  A T (X)                                       00000100
C             0       K=1  K K                                          00000110
C                                                                       00000120
C          = A T (X) + A   T   (X) + A   T   (X) + . . . + A T (X) +    00000130
C             M M       M-1 M-1       M-2 M-2               2 2       00000140
C                                                                       00000150
C                             + A T (X) + A /2                          00000160
C                                1 1       0                          00000170
C                                                                       00000180
C         AT A PARTICULAR POINT XPT.                                    00000190
C                                                                       00000200
C                                                                       00000210
C     CALLING SEQUENCE                                                  00000220
C         CALL CSER1(MORD,XPT,AM,CSER)                                  00000230
C                                                                       00000240
C     PARAMETERS                                                        00000250
C         MORD      I    HIGHEST ORDER OF CHEBYSHEV POLYNOMIALS IN      00000260
C                        THE SERIES REPRESENTATION.                     00000270
C         XPT       I    POINT AT WHICH THE SERIES IS TO BE EVALUATED.  00000280
C         AM        I    ARRAY (70X1) CONTAINING THE COEFFICIENTS OF    00000290
C                        THE CHEBYSHEV POLYNOMIALS IN THE SERIES.       00000300
C         CSER      O    RESULT OF EVALUATING Y(XPT) BY NESTED          00000310
C                        ITERATION BASED ON THE THREE TERM RECURRENCE   00000320
C                        RELATION.                                      00000330
C                                                                       00000340
C     EXTERNAL REFERENCES                                               00000350
C         NONE                                                          00000360
C                                                                       00000370
C     ANALYSIS                                                          00000380
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000390
C                                                                       00000400
C     PROGRAMMER                                                        00000410
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000420
C                                                                       00000430
      IMPLICIT REAL*8(A-H,O-Z)                                          00000440
      DIMENSION     AM(70)                                              00000450
C                                                                       00000460
C     INITIALIZE THE RECURRENCE RELATION                                00000470
C                                                                       00000480
      B = AM(MORD+1) * XPT                                              00000490
      C = AM(MORD+1)                                                    00000500
      A = 2.D0 * XPT * B - C                                            00000510
      M2 = MORD - 2                                                     00000520
C                                                                       00000530
C     BEGIN RECURRENCE                                                  00000540
C                                                                       00000550
      DO 50 IN = 1,M2                                                   00000560
      C = B + AM(MORD+1-IN)                                             00000570
```

```
CSER1                JSS1493.PRLLEL.ORBDET.FORT

      B = A + AM(MORD+1-IN) * XPT                                       00000580
      A = 2.D0 * XPT * B - C                                            00000590
   50 CONTINUE                                                          00000600
C                                                                       00000610
C     COMPUTE FINAL VALUE                                               00000620
C                                                                       00000630
      CSER = A + AM(2) * XPT + AM(1)/2.D0                               00000640
      RETURN                                                            00000650
      END                                                               00000660

******** END OF MEMBER CSER1       66 RECORDS ********
```

```
CSRSTA              JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE CSRSTA(STA,TU,TOUT,H,IV)                                00000010
C                                                                        00000020
C     VERSION OF 06/15/79     **** CHECKED OUT ****                      00000030
C                                                                        00000040
C         THIS IS A REWRITE OF SUBROUTINE CHIRP, BUT MODIFIED AS FOLLOWS 00000050
C         1.  THE LINEAR CHANGE OF VARIABLES BETWEEN TRUE INDEPENDENT    00000060
C             VARIABLE T (TL.LE.T.LE.TU) AND CHEBYSHEV INDEPENDENT       00000070
C             VARIABLE X (-1.LE.X.LE.+1) HAS BEEN MODIFIED SUCH THAT     00000080
C             WHEN T=TL IMPLIES X=-1 AND WHEN T=TU IMPLIES X=+1.         00000090
C             THIS IS TO REMAIN COMPATIBLE WITH THE COMPLETELY NEW       00000100
C             VERSION OF SUBROUTINE COPS.                                00000110
C         2.  THE COEFFICIENTS FOR THE INTERPOLATION ARE PASSED THROUGH  00000120
C             THE BC ARRAY OF COMMON/COVAL /.                            00000130
C         3.  THE EVALUATION OF EACH SERIES IS PERFORMED BY SUBROUTINE   00000140
C             CSER1 INSTEAD OF SUBROUTINE EVAPT.                         00000150
C         4.  THE PROGRAM HAS BEEN GENERALIZED TO INTERPOLATE THE SPACE- 00000160
C             CRAFT STATE, THE DEFINITION OF WHICH DEPENDS ON THE VALUE  00000170
C             OF THE FLAG IEQSET.                                        00000180
C             IEQSET=1:  STA=X,Y,Z COMPONENTS OF POSITION AND VELOCITY.  00000190
C             IEQSET=2:  STA=KEPLERIAN ELEMENTS (MEAN ANOMALY).          00000200
C             IEQSET=3:  STA=EQUINOCTIAL ELEMENTS (MEAN LONGITUDE).      00000210
C             IEQSET=4:  STA=EQUINOCTIAL ELEMENTS (ECCENTRIC LONGITUDE). 00000220
C             IEQSET=5:  STA=AVERAGED EQUINOCTIAL ELEMENTS (MEAN MEAN    00000230
C                            LONGITUDE).                                 00000240
C             IEQSET=6:  STA=AVERAGED KEPLERIAN ELEMENTS (MEAN MEAN      00000250
C                            ANOMALY).                                   00000260
C                                                                        00000270
C     PURPOSE                                                            00000280
C         TO EVALUATE THE STATE OF A SPACECRAFT BY INTERPOLATING THE     00000290
C         APPROPRIATE CHEBYSHEV SERIES.                                  00000300
C                                                                        00000310
C     CALLING SEQUENCE                                                   00000320
C         CALL CSRSTA(STA,TU,TOUT,H,IV)                                  00000330
C                                                                        00000340
C     PARAMETERS                                                         00000350
C         STA       O    ARRAY (6X1) CONTAINING THE COMPONENTS OF THE    00000360
C                        SPACECRAFT STATE VECTOR, AS DEFINED BY THE      00000370
C                        VALUE OF IEQSET.                                00000380
C         TU        I    UPPER LIMIT OF THE TIME SPAN BEING INTEGRATED.  00000390
C         TOUT      I    OUTPUT TIME AT WHICH THE SPACECRAFT STATE IS    00000400
C                        TO BE INTERPOLATED.                             00000410
C         H         I    ARC LENGTH OF INTEGRATION SPAN.                 00000420
C         IV        I    INTERPOLATION FLAG:                             00000430
C                        IV=1:  POSITION AND VELOCITY                    00000440
C                        IV=2:  POSITION ONLY                            00000450
C                        IV=3:  VELOCITY ONLY                            00000460
C                                                                        00000470
C     EXTERNAL REFERENCES                                                00000480
C         CSER1                                                          00000490
C                                                                        00000500
C     ANALYSIS                                                           00000510
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000520
C                                                                        00000530
C     PROGRAMMER                                                         00000540
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000550
C                                                                        00000560
      IMPLICIT REAL*8(A-H,O-Z)                                           00000570
```

```
CSRSTA              JSS1493.PRLLEL.ORBDET.FORT

      DIMENSION       STA(6)                                                    00000580
      COMMON/COVAL /  A(70,6)  ,  BC(70,6)  ,  FORC(70,6)                       00000590
      COMMON/NPCNTL/  MORD     ,  NSUB      ,  IEQSET      ,  ITERM             00000600
C                                                                               00000610
C     COMPUTE THE CHEBYSHEV VARIABLE FOR THE REQUESTED OUTPUT TIME.             00000620
C                                                                               00000630
      PI = 3.141592653589793D0                                                  00000640
      XOUT = 2.D0 * ((TOUT - TU + H)/H) - 1.D0                                  00000650
      M1 = MORD + 1                                                             00000660
      M2 = MORD + 2                                                             00000670
      IF  (IEQSET.NE.1)       GO TO 100                                         00000680
      IF  (IV.EQ.3)           GO TO 20                                          00000690
C                                                                               00000700
C     IEQSET.EQ.1:  INTERPOLATE SPACECRAFT POSITION                             00000710
C                                                                               00000720
      DO  10   I=1,3                                                            00000730
      CALL CSER1(M2,XOUT,BC(1,I+3),STA(I))                                      00000740
   10 CONTINUE                                                                  00000750
      IF  (IV.EQ.2)           RETURN                                            00000760
C                                                                               00000770
C     IEQSET.EQ.1:  INTERPOLATE SPACECRAFT VELOCITY                             00000780
C                                                                               00000790
   20 DO  30   I=1,3                                                            00000800
      CALL CSER1(M2,XOUT,BC(1,I),STA(I+3))                                      00000810
   30 CONTINUE                                                                  00000820
      RETURN                                                                    00000830
C                                                                               00000840
C     IEQSET.NE.1:  THIS ROUTINE IS BEING USED TO INTERPOLATE ELEMENTS          00000850
C     FROM THE SOLUTION OF THE APPROPRIATE VOP EQUATIONS.  THE                  00000860
C     INTERPOLATION IS AS FOLLOWS:                                              00000870
C                                                                               00000880
  100 DO 110   I=1,6                                                            00000890
      CALL CSER1(M1,XOUT,BC(1,I),STA(I))                                        00000900
  110 CONTINUE                                                                  00000910
      IF (IEQSET.EQ.2.OR.IEQSET.EQ.6)   GO TO 120                               00000920
      N2PI = STA(6)/(2.D0 * PI)                                                 00000930
      STA(6) = STA(6) - DFLOAT(N2PI) * 2.D0 * PI                                00000940
      RETURN                                                                    00000950
  120 DO  130  I = 3,6                                                          00000960
      N2PI = STA(I)/(2.D0 * PI)                                                 00000970
  130 STA(I) = STA(I) - DFLOAT(N2PI) * 2.D0 * PI                                00000980
      RETURN                                                                    00000990
      END                                                                       00001000

******** END OF MEMBER CSRSTA     100 RECORDS ********
```

```
      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 08/07/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE VARIABLES IN THE NAMED COMMON BLOCK DRGPAR      00000060
C         FOR THE TEST SATELLITE "AEC-CIRCULAR"                         00000070
C                                                                       00000080
C     PROGRAMMING                                                       00000090
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000100
C                                                                       00000110
C * * * * * * * * *            /DRGPAR/           * * * * * * * * * * * 00000120
C                                                                       00000130
      IMPLICIT REAL*8(A-H,O-Z)                                          00000140
      COMMON/DRGPAR/  CSUBD   , SCAREA  , SCMASS  , RHOZRO  ,           00000150
     *                RZERO   , SCLHGT  , IDRAG                        00000160
C                                                                       00000170
C                      #LOCA-                                           00000180
C              DIMEN-   TION                                            00000190
C  VARIABLE     SION    (REAL)              DESCRIPTION                 00000200
C  ________    _____    ______  ____________________________________    00000210
C                                                                       00000220
C  CSUBD         1        1    DRAG COEFFICIENT                        00000230
C  SCAREA        1        2    S/C CROSS SECTIONAL AREA PERPENDICULAR  00000240
C                              TO DIRECTION OF MOTION (KM**2)          00000250
C  SCMASS        1        3    S/C MASS (KG)                           00000260
C  RHOZRO        1        4    ATMOSPHERIC DENSITY AT THE REFERENCE    00000270
C                              DISTANCE RZERO (KG/KM**3)               00000280
C  RZERO         1        5    REFERENCE DISTANCE (KM), TAKEN AS THE   00000290
C                              MEAN RADIUS OF THE CENTRAL BODY         00000300
C  SCLHGT        1        6    ATMOSPHERIC SCALE HEIGHT (KM)           00000310
C                                                                       00000320
C                      #LOCA-                                           00000330
C              DIMEN-   TION                                            00000340
C  VARIABLE     SION   INTEGER              DESCRIPTION                 00000350
C  ________    _____   _______  ____________________________________    00000360
C                                                                       00000370
C  IDRAG         1        1    FLAG INDICATING DRAG IS TO BE INCLUDED  00000380
C                              IN THE FORCE MODEL:                     00000390
C                              IDRAG = 0  =>  NO                       00000400
C                              IDRAG = 1  =>  YES                      00000410
C                                                                       00000420
      DATA CSUBD    /0.200000000 01/                                    00000430
C                                                                       00000440
      DATA SCAREA   /0.186000000-05/                                    00000450
C                                                                       00000460
      DATA SCMASS   /0.677000000 03/                                    00000470
C                                                                       00000480
      DATA RHOZRO   /0.139154000 10/                                    00000490
C                                                                       00000500
      DATA RZERO    /0.637814000 04/                                    00000510
C                                                                       00000520
      DATA SCLHGT   /0.716280000 01/                                    00000530
C                                                                       00000540
      DATA IDRAG    /0             /                                    00000550
C                                                                       00000560
      END                                                               00000570
```

******** END OF MEMBER DRGPAR 57 RECORDS ********

```
      SUBROUTINE ELMRPT(RV,T0,TOUT,GM,ISET,IPRNT,ELMNTS)                00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE KEPLERIAN OR EQUINOCTIAL ELEMENTS AT THE CUR-  00000060
C         RENT TIME TOUT, GIVEN THE INERTIAL CARTESIAN COMPONENTS OF    00000070
C         POSITION AND VELOCITY AT THAT TIME.  IF ISET=1, THIS SUB-     00000080
C         ROUTINE WILL RETURN THE KEPLERIAN ELEMENTS AND CERTAIN        00000090
C         AUXILIARY QUANTITIES.  IF ISET=2, THIS SUBROUTINE WILL        00000100
C         RETURN THE NONSINGULAR EQUINOCTIAL ELEMENTS AND CERTAIN       00000110
C         AUXILIARY QUANTITIES.                                         00000120
C                                                                       00000130
C     CALLING SEQUENCE                                                  00000140
C         CALL ELMRPT(RV,T0,TOUT,GM,ISET,IPRNT,ELMNTS)                  00000150
C                                                                       00000160
C     PARAMETERS                                                        00000170
C         RV        I    ARRAY (6X1) CONTAINING THREE INERTIAL CAR-     00000180
C                        TESIAN COORDINATES OF POSITION AND THREE       00000190
C                        INERTIAL CARTESIAN COORDINATES OF VELOCITY     00000200
C                        AT THE OUTPUT TIME.                            00000210
C         T0        I    EPOCH TIME.                                    00000220
C         TOUT      I    OUTPUT TIME TO COMPUTE ELEMENTS.               00000230
C         GM        I    GRAVITATIONAL PARAMETER.                       00000240
C         ISET      I    FLAG TO SELECT DESIRED ELEMENT SET.            00000250
C         IPRNT     I    FLAG TO DETERMINE IF THE ELEMENTS SHOULD BE    00000260
C                        PRINTED:                                       00000270
C                        IPRNT = 0:  DO NOT PRINT                       00000280
C                        IPRNT = 1:  PRINT ONLY POSITION, VELOCITY AND  00000290
C                                    ORBITAL ELEMENTS                   00000300
C                        IPRNT = 2:  PRINT POSITION, VELOCITY, ORBITAL  00000310
C                                    ELEMENTS AND AUXILIARY QUANTITIES  00000320
C         ELMNTS    O    ARRAY (26X1) CONTAINING APPROPRIATE ELEMENTS   00000330
C                        AND AUXILIARY QUANTITIES.                      00000340
C                        FOR ISET=1:  KEPLERIAN ELEMENTS                00000350
C                        ELMNTS(1)  = SEMIMAJOR AXIS, A                 00000360
C                        ELMNTS(2)  = ECCENTRICITY, E                   00000370
C                        ELMNTS(3)  = INCLINATION, I                    00000380
C                        ELMNTS(4)  = ARGUMENT OF PERIAPSIS             00000390
C                        ELMNTS(5)  = LONGITUDE OF THE ASCENDING NODE   00000400
C                        ELMNTS(6)  = TIME OF PERIAPSIS PASSAGE (SEC)   00000410
C                        ELMNTS(7)  = MEAN MOTION (DEG/SEC)             00000420
C                        ELMNTS(8)  = SEMI-LATUS RECTUM (PARAMETER)     00000430
C                        ELMNTS(9)  = LONGITUDE OF PERIAPSIS            00000440
C                        ELMNTS(10) = TRUE ANOMALY                      00000450
C                        ELMNTS(11) = ARGUMENT OF LATITUDE              00000460
C                        ELMNTS(12) = TRUE LONGITUDE                    00000470
C                        ELMNTS(13) = MAGNITUDE OF ANGULAR MOMENTUM     00000480
C                        ELMNTS(14) = MAGNITUDE OF POSITION VECTOR      00000490
C                        ELMNTS(15) = MAGNITUDE OF VELOCITY VECTOR      00000500
C                        ELMNTS(16) = Z-COMPONENT OF ANGULAR MOMENTUM   00000510
C                        ELMNTS(17) = PERIFOCAL RADIUS                  00000520
C                        ELMNTS(18) = APOFOCAL RADIUS                   00000530
C                        ELMNTS(19) = PERIOD                            00000540
C                        ELMNTS(20) = ENERGY (OSCULATING)               00000550
C                        ELMNTS(21) = ENERGY (ZONAL PERTURBED)          00000560
C                        ELMNTS(22) = ECCENTRIC ANOMALY                 00000570
```

```
C                       ELMNTS(23) = MEAN ANOMALY                          00000580
C                       ELMNTS(24) = RIGHT ASCENSION                       00000590
C                       ELMNTS(25) = DECLINATION                           00000600
C                       ELMNTS(26) = MEAN LONGITUDE                        00000610
C                                                                          00000620
C                       FOR ISET=2:  EQUINOCTIAL ELEMENTS                  00000630
C                       ELMNTS(1)  = SEMIMAJOR AXIS, A                     00000640
C                       ELMNTS(2)  = H                                     00000650
C                       ELMNTS(3)  = K                                     00000660
C                       ELMNTS(4)  = P                                     00000670
C                       ELMNTS(5)  = Q                                     00000680
C                       ELMNTS(6)  = LAMDA, MEAN LONGITUDE                 00000690
C                       ELMNTS(7)  = MEAN MOTION (DEG/SEC)                 00000700
C                       ELMNTS(8)  = SEMI-LATUS RECTUM (PARAMETER)         00000710
C                       ELMNTS(9)  = ECCENTRICITY, E                       00000720
C                       ELMNTS(10) = INCLINATION, I                        00000730
C                       ELMNTS(11) = ARGUMENT OF PERIAPSIS                 00000740
C                       ELMNTS(12) = LONGITUDE OF THE ASCENDING NODE       00000750
C                       ELMNTS(13) = RETROGRADE FACTOR                     00000760
C                       ELMNTS(14) = MAGNITUDE OF THE POSITION VECTOR      00000770
C                       ELMNTS(15) = MAGNITUDE OF THE VELOCITY VECTOR      00000780
C                       ELMNTS(16) = Z-COMPONENT OF ANGULAR MOMENTUM       00000790
C                       ELMNTS(17) = PERIFOCAL RADIUS                      00000800
C                       ELMNTS(18) = APOFOCAL RADIUS                       00000810
C                       ELMNTS(19) = PERIOD                                00000820
C                       ELMNTS(20) = ENERGY (OSCULATING)                   00000830
C                       ELMNTS(21) = ENERGY (ZONAL PERTURBED)              00000840
C                       ELMNTS(22) = ECCENTRIC LONGITUDE                   00000850
C                       ELMNTS(23) = MEAN ANOMALY                          00000860
C                       ELMNTS(24) = TRUE LONGITUDE                        00000870
C                       ELMNTS(25) = TIME OF PERIAPSIS PASSAGE             00000880
C                       ELMNTS(26) = MAGNITUDE OF ANGULAR MOMENTUM         00000890
C                                                                          00000900
C     EXTERNAL REFERENCES                                                  00000910
C         NONE                                                             00000920
C                                                                          00000930
C     ANALYSIS                                                             00000940
C         CAPTAIN JEFFREY S. SHAVER, USAF                                  00000950
C                                                                          00000960
C     PROGRAMMER                                                           00000970
C         CAPTAIN JEFFREY S. SHAVER, USAF                                  00000980
C                                                                          00000990
      IMPLICIT REAL*8(A-H,O-Z)                                             00001000
C                                                                          00001010
C * * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * 00001020
C                                                                          00001030
      DIMENSION     RV(6)  ,   ELMNTS(26)                                  00001040
C                                                                          00001050
C * * * * * * * * * *         COMMON BLOCKS           * * * * * * * * * * 00001060
C                                                                          00001070
C     COMMON/CNSTNT/  PI        ,  TWOPI     ,  DTR      ,  RTD           00001080
C                                                                          00001090
      COMMON/EQPARM/  FGW(3,3)  ,  YAUX(3)   ,  PQ       ,  BETA        , 00001100
     *                XX        ,  YY        ,  XXD      ,  YYD         , 00001110
     *                AHK       ,  GG                                     00001120
C                                                                          00001130
      COMMON/ERTSHP/  REQ       ,  WE        ,  FINV     ,  SPHINF        00001140
```

```
C                                                                       00001150
      COMMON/ZNLCOF/  SHJ(8)    ,  IZNLS                                00001160
C                                                                       00001170
C * * * * * * * * * *     EQUIVALENCE STATEMENTS     * * * * * * * * * * 00001180
C                                                                       00001190
C     FROM COMMON/EQPARM/:                                              00001200
      EQUIVALENCE     (FGW(1,1) , FX ),(FGW(1,2) , GX ),(FGW(1,3), WX ),00001210
     *                (FGW(2,1) , FY ),(FGW(2,2) , GY ),(FGW(2,3), WY ),00001220
     *                (FGW(3,1) , FZ ),(FGW(3,2) , GZ ),(FGW(3,3), WZ ) 00001230
      PI = 3.1415926535897932D0                                         00001240
      DTR  = 180.D0/PI                                                  00001250
C                                                                       00001260
C     CONVERT ELAPSED TIME FROM EPOCH (GIVEN IN SECONDS) INTO DAYS,     00001270
C     HOURS, MINUTES AND SECONDS                                        00001280
C                                                                       00001290
      TELPSD = TOUT - T0                                                00001300
      TSEC   = DMOD(TELPSD,60.D00)                                      00001310
      TMINS1 = (TELPSD - TSEC)/60.D00                                   00001320
      TMIN   = DMOD(TMINS1,60.D00)                                      00001330
      THRS1  = (TMINS1 - TMIN)/60.D00                                   00001340
      THRS   = DMOD(THRS1,24.D00)                                       00001350
      TDAYS  = (THRS1 - THRS)/24.D00                                    00001360
C                                                                       00001370
C     COMPUTE MAGNITUDES OF THE POSITION AND VELOCITY VECTORS AND THE   00001380
C     DOT PRODUCT OF THE POSITION VECTOR WITH THE VELOCITY VECTOR       00001390
C                                                                       00001400
      R0 = DSQRT(RV(1)**2 + RV(2)**2 + RV(3)**2)                        00001410
      ELMNTS(14) = R0                                                   00001420
      V0 = DSQRT(RV(4)**2 + RV(5)**2 + RV(6)**2)                        00001430
      ELMNTS(15) = V0                                                   00001440
      RDOTV = RV(1)*RV(4) + RV(2)*RV(5) + RV(3)*RV(6)                   00001450
C                                                                       00001460
C     COMPUTE ANGULAR MOMENTUM VECTOR COMPONENTS AND MAGNITUDE          00001470
C                                                                       00001480
      HX = RV(2)*RV(6) - RV(3)*RV(5)                                    00001490
      HY = RV(3)*RV(4) - RV(1)*RV(6)                                    00001500
      HZ = RV(1)*RV(5) - RV(2)*RV(4)                                    00001510
      ELMNTS(16) = HZ                                                   00001520
      ELMNTS(13) = DSQRT(HX**2 + HY**2 + HZ**2)                         00001530
C                                                                       00001540
C     COMPUTE ECCENTRICITY VECTOR COMPONENTS AND MAGNITUDE              00001550
C                                                                       00001560
      EX = (RV(5)*HZ - RV(6)*HY)/GM - RV(1)/R0                          00001570
      EY = (RV(6)*HX - RV(4)*HZ)/GM - RV(2)/R0                          00001580
      EZ = (RV(4)*HY - RV(5)*HX)/GM - RV(3)/R0                          00001590
      ELMNTS(2) = DSQRT(EX**2 + EY**2 + EZ**2)                          00001600
C                                                                       00001610
C     COMPUTE SEMI-LATUS RECTUM, SEMIMAJOR AXIS, MEAN MOTION            00001620
C                                                                       00001630
      ELMNTS(8) = ELMNTS(13)**2/GM                                      00001640
      ELMNTS(1) = ELMNTS(8)/(1.D0-ELMNTS(2)**2)                         00001650
      ELMNTS(7) = DSQRT(GM/ELMNTS(1) ** 3)                              00001660
C                                                                       00001670
C     COMPUTE PERIFOCAL AND APOFOCAL RADII, PERIOD, ENERGY (OSCULATING) 00001680
C                                                                       00001690
      ELMNTS(17) = ELMNTS(1) * (1.D0 - ELMNTS(2))                       00001700
      ELMNTS(18) = ELMNTS(1) * (1.D0 + ELMNTS(2))                       00001710
```

```
      ELMNTS(19) = (2.D0 * PI/ELMNTS(7))/3.6D03                          00001720
      ELMNTS(20) = -GM/(2.D0 * ELMNTS(1))                                00001730
C                                                                        00001740
C     COMPUTE ENERGY (ZONAL PERTURBED):  THIS IS A CONSTANT OF THE       00001750
C     MOTION WHEN DISTURBING FUNCTION CONSISTS ONLY OF ZONAL HARMONICS   00001760
C                                                                        00001770
      ELMNTS(21) = ELMNTS(20)                                            00001780
      IF (IZNLS.LT.2)                    GO TO 5                         00001790
      ARG    = RV(3)/R0                                                  00001800
      CFAC   = GM/R0                                                     00001810
      PKMN1 = ARG                                                        00001820
      PKMN2 = 1.D0                                                       00001830
      K = 2                                                              00001840
    1 DK = DFLOAT(K)                                                     00001850
      COFK1 = ((2.D0 * DK - 1.D0)/DK) * ARG                              00001860
      COFK2 = (DK - 1.D0)/DK                                             00001870
      FK     = COFK1 * PKMN1 - COFK2 * PKMN2                             00001880
      COF    = SHJ(K) * (REQ/R0)**K                                      00001890
      ELMNTS(21) = ELMNTS(21) + CFAC * COF * FK                          00001900
      IF (K.GE.IZNLS)                    GO TO 5                         00001910
      PKMN2 = PKMN1                                                      00001920
      PKMN1 = FK                                                         00001930
      K = K + 1                                                          00001940
      GO TO 1                                                            00001950
    5 CONTINUE                                                           00001960
C                                                                        00001970
C     COMPUTE THE ORBITAL INCLINATION                                    00001980
C                                                                        00001990
      ELMNTS(3) = DARCOS(HZ/ELMNTS(13))                                  00002000
      IF (ISET.EQ.2)  GO TO 50                                           00002010
C                                                                        00002020
C     COMPUTE UNIT VECTOR IN NODAL DIRECTION, IF NODE IS DEFINED         00002030
C                                                                        00002040
      IF (ELMNTS(3).LT.1.D-06)  GO TO 20                                 00002050
      UNITN  = DSQRT(HY**2 + HX**2)                                      00002060
      UNITNX = -HY/UNITN                                                 00002070
      UNITNY =  HX/UNITN                                                 00002080
C                                                                        00002090
C     COMPUTE LONGITUDE OF ASCENDING NODE, IF INCLINATION IS NONZERO     00002100
C                                                                        00002110
      ELMNTS(5) = DARCOS(UNITNX)                                         00002120
      IF (UNITNY.LT.0.D0) ELMNTS(5) = 2.D0 * PI - ELMNTS(5)              00002130
C                                                                        00002140
C     COMPUTE ARGUMENT OF PERIAPSIS, I.NE.0 AND E.NE.0                   00002150
C                                                                        00002160
      IF (ELMNTS(2).LT.1.D-06)  GO TO 10                                 00002170
      ELMNTS(4) = DARCOS((UNITNX*EX + UNITNY*EY)/ELMNTS(2))              00002180
      IF (EZ.LT.0.D0) ELMNTS(4) = 2.D0 * PI - ELMNTS(4)                  00002190
C                                                                        00002200
C     COMPUTE LONGITUDE OF PERIAPSIS, I.NE.0 AND E.NE.0                  00002210
C                                                                        00002220
      ELMNTS(9) = ELMNTS(4) + ELMNTS(5)                                  00002230
C                                                                        00002240
C     COMPUTE TRUE ANOMALY AT EPOCH, I.NE.0 AND E.NE.0                   00002250
C                                                                        00002260
      ELMNTS(10) = DARCOS((EX*RV(1)+EY*RV(2)+EZ*RV(3))/(R0*ELMNTS(2)))  00002270
      IF (RDOTV.LT.0.D0) ELMNTS(10) = 2.D0 * PI - ELMNTS(10)             00002280
```

```
C                                                                       00002290
C     COMPUTE ARGUMENT OF LATITUDE, TRUE LONGITUDE AT EPOCH             00002300
C                                                                       00002310
      ELMNTS(11) = ELMNTS(4) + ELMNTS(10)                               00002320
      ELMNTS(12) = ELMNTS(5) + ELMNTS(11)                               00002330
      GO TO 40                                                          00002340
C                                                                       00002350
C     BEGIN COMPUTATIONS FOR INCLINATION.NE.0 BUT ECCENTRICITY.EQ.0     00002360
C                                                                       00002370
   10 ELMNTS(4)  = 0.D0                                                 00002380
      ELMNTS(9)  = ELMNTS(5)                                            00002390
      ELMNTS(11) = DARCOS((UNITNX*RV(1) + UNITNY*RV(2))/R0)             00002400
      IF (RV(3).LT.0.D0)  ELMNTS(11) = 2.D0 * PI - ELMNTS(11)           00002410
      ELMNTS(10) = ELMNTS(11)                                           00002420
      ELMNTS(12) = ELMNTS(5) + ELMNTS(11)                               00002430
      GO TO 40                                                          00002440
C                                                                       00002450
C     BEGIN COMPUTATIONS FOR INCLINATION.EQ.0 BUT ECCENTRICITY.NE.0     00002460
C                                                                       00002470
   20 IF (ELMNTS(2).LT.1.D-06)  GO TO 30                                00002480
      ELMNTS(5)  = 0.D0                                                 00002490
      ELMNTS(4)  = DARCOS(EX/ELMNTS(2))                                 00002500
      IF (EY.LT.0.D0)  ELMNTS(4) = 2.D0 * PI - ELMNTS(4)                00002510
      ELMNTS(9)  = ELMNTS(4)                                            00002520
      ELMNTS(10) = DARCOS((EX*RV(1)+EY*RV(2)+EZ*RV(3))/(R0*ELMNTS(2)))  00002530
      IF (RDOTV.LT.0.D0)  ELMNTS(10) = 2.D0 * PI - ELMNTS(10)           00002540
      ELMNTS(11) = ELMNTS(4) + ELMNTS(10)                               00002550
      ELMNTS(12) = ELMNTS(11)                                           00002560
      GO TO 40                                                          00002570
C                                                                       00002580
C     BEGIN COMPUTATIONS FOR INCLINATION.EQ.0 AND ECCENTRICITY.EQ.0     00002590
C                                                                       00002600
   30 ELMNTS(4)  = 0.D0                                                 00002610
      ELMNTS(5)  = 0.D0                                                 00002620
      ELMNTS(9)  = 0.D0                                                 00002630
      ELMNTS(12) = DARCOS(RV(1)/R0)                                     00002640
      IF (RV(2).LT.0.D0)  ELMNTS(12) = 2.D0 * PI - ELMNTS(12)           00002650
      ELMNTS(10) = ELMNTS(12)                                           00002660
      ELMNTS(11) = ELMNTS(12)                                           00002670
C                                                                       00002680
C     COMPUTE TIME OF PERIAPSIS PASSAGE                                 00002690
C                                                                       00002700
   40 CE0 = (ELMNTS(2) + DCOS(ELMNTS(10)))/                             00002710
     1                        (1.D0 + ELMNTS(2) * DCOS(ELMNTS(10)))     00002720
      SE0 = DSQRT(1.D0 - ELMNTS(2)**2) * DSIN(ELMNTS(10))/              00002730
     1                        (1.D0 + ELMNTS(2) * DCOS(ELMNTS(10)))     00002740
      E0  = DATAN2(SE0,CE0)                                             00002750
      IF (E0.LT.0.D0) E0 = 2.D0 * PI + E0                               00002760
      ELMNTS(22) = E0                                                   00002770
      EPM = E0 - ELMNTS(2) * DSIN(E0)                                   00002780
      IF (EPM.GE.2.D0 * PI)  EPM = EPM - 2.D0 * PI                      00002790
      ELMNTS(23) = EPM                                                  00002800
      ELMNTS(6) = (ELMNTS(7) * T0 - EPM)/ELMNTS(7)                      00002810
C                                                                       00002820
C     COMPUTE RIGHT ASCENSION, DECLINATION, MEAN LONGITUDE              00002830
C                                                                       00002840
      ELMNTS(24) = DARCOS(RV(1)/DSQRT(RV(1)**2 + RV(2)**2))             00002850
```

```
      IF (RV(2).LT.0.D0)  ELMNTS(24) = 2.D0 * PI - ELMNTS(24)          00002860
      ELMNTS(25) = DARSIN(RV(3)/R0)                                    00002870
      ELMNTS(26) = ELMNTS(9) + ELMNTS(23)                              00002880
      IF ( ELMNTS(9).GE.2.D0*PI)  ELMNTS(9)  = ELMNTS(9)  - 2.D0 * PI  00002890
      IF (ELMNTS(11).GE.2.D0*PI)  ELMNTS(11) = ELMNTS(11) - 2.D0 * PI  00002900
      IF (ELMNTS(12).GE.2.D0*PI)  ELMNTS(12) = ELMNTS(12) - 2.D0 * PI  00002910
      IF (ELMNTS(26).GE.2.D0*PI)  ELMNTS(26) = ELMNTS(26) - 2.D0 * PI  00002920
      GO TO 100                                                        00002930
C                                                                      00002940
C     IF ISET=2, COMPUTE THE EQUINOCTIAL ELEMENTS                      00002950
C     COMPUTE UNIT VECTOR W AND AUXILIARY VECTOR YAUX                  00002960
C                                                                      00002970
   50 ELMNTS(26) = ELMNTS(13)                                          00002980
      GG = ELMNTS(26)                                                  00002990
      ELMNTS(9)  = ELMNTS(2)                                           00003000
      ELMNTS(10) = ELMNTS(3)                                           00003010
      WX = HX/ELMNTS(13)                                               00003020
      WY = HY/ELMNTS(13)                                               00003030
      WZ = HZ/ELMNTS(13)                                               00003040
      YAUX(1) = (WY * RV(3) - WZ * RV(2))/R0                           00003050
      YAUX(2) = (WZ * RV(1) - WX * RV(3))/R0                           00003060
      YAUX(3) = (WX * RV(2) - WY * RV(1))/R0                           00003070
C                                                                      00003080
C     COMPUTE RETROGRADE FACTOR                                        00003090
C                                                                      00003100
      IF (ELMNTS(3).GE.0.D0.AND.ELMNTS(3).LE.PI/2.D0) ELMNTS(13) = 1.D0 00003110
      IF (ELMNTS(3).GT.PI/2.D0.AND.ELMNTS(3).LE.PI)   ELMNTS(13) =-1.D0 00003120
C                                                                      00003130
C     COMPUTE EQUINOCTIAL ELEMENTS P AND Q                             00003140
C                                                                      00003150
      ELMNTS(4) =  WX/(1.D0 + ELMNTS(13) * WZ)                         00003160
      ELMNTS(5) = -WY/(1.D0 + ELMNTS(13) * WZ)                         00003170
C                                                                      00003180
C     COMPUTE UNIT VECTORS F AND G                                     00003190
C                                                                      00003200
      PQ = 1.D0 + ELMNTS(4)**2 + ELMNTS(5)**2                          00003210
      FX = (1.D0 - ELMNTS(4)**2 + ELMNTS(5)**2)/PQ                     00003220
      FY = 2.D0 * ELMNTS(4) * ELMNTS(5)/PQ                             00003230
      FZ = -2.D0 * ELMNTS(4) * ELMNTS(13)/PQ                           00003240
      GX = 2.D0 * ELMNTS(4) * ELMNTS(5) * ELMNTS(13)/PQ                00003250
      GY = (1.D0 + ELMNTS(4)**2 - ELMNTS(5)**2) * ELMNTS(13)/PQ        00003260
      GZ = 2.D0 * ELMNTS(5)/PQ                                         00003270
C                                                                      00003280
C     COMPUTE EQUINOCTIAL ELEMENTS H AND K                             00003290
C                                                                      00003300
      ELMNTS(2) = EX*GX + EY*GY + EZ*GZ                                00003310
      ELMNTS(3) = EX*FX + EY*FY + EZ*FZ                                00003320
C                                                                      00003330
C     COMPUTE AUXILIARY QUANTITIES XX, YY, XXD, YYD, BETA, AND         00003340
C     A*SQRT(1-H**2-K**2)                                              00003350
C                                                                      00003360
      XX   = RV(1)*FX + RV(2)*FY + RV(3)*FZ                            00003370
      YY   = RV(1)*GX + RV(2)*GY + RV(3)*GZ                            00003380
      XXD  = RV(4)*FX + RV(5)*FY + RV(6)*FZ                            00003390
      YYD  = RV(4)*GX + RV(5)*GY + RV(6)*GZ                            00003400
      BETA = 1.D0/(1.D0+DSQRT(1.D0-ELMNTS(2)**2-ELMNTS(3)**2))         00003410
      AHK  = ELMNTS(1) * DSQRT(1.D0-ELMNTS(2)**2-ELMNTS(3)**2)         00003420
```

```
C                                                                       00003430
C     COMPUTE COS(F), SIN(F), F, LAMDA                                  00003440
C                                                                       00003450
      CSF = ELMNTS(3) + ((1.D0+BETA*ELMNTS(3)**2)*XX-ELMNTS(2)*ELMNTS(3)00003460
     1                        *BETA*YY)/AHK                            00003470
      SNF = ELMNTS(2) + ((1.D0-BETA*ELMNTS(2)**2)*YY-ELMNTS(2)*ELMNTS(3)00003480
     1                        *BETA*XX)/AHK                            00003490
      F = DATAN2(SNF,CSF)                                               00003500
      IF (F.LT.0.D0)  F = 2.D0 * PI + F                                 00003510
      ELMNTS(6)  = F - ELMNTS(3) * SNF + ELMNTS(2) * CSF                00003520
      ELMNTS(22) = F                                                    00003530
C                                                                       00003540
C     COMPUTE ARGUMENT OF PERIAPSIS, LONGITUDE OF THE ASCENDING NODE    00003550
C                                                                       00003560
      IF (ELMNTS(4).EQ.0.D0.AND.ELMNTS(5).EQ.0.D0)  GO TO 60            00003570
      ELMNTS(12) = DATAN2(ELMNTS(4),ELMNTS(5))                          00003580
      IF (ELMNTS(12).LT.0.D0)       ELMNTS(12) = 2.D0 * PI + ELMNTS(12) 00003590
      GO TO 70                                                          00003600
   60 ELMNTS(12) = 0.D0                                                 00003610
   70 IF (ELMNTS(2).EQ.0.D0.AND.ELMNTS(3).EQ.0.D0)  GO TO 80            00003620
      ARCHK      = DATAN2(ELMNTS(2),ELMNTS(3))                          00003630
      IF (ARCHK.LT.0.D0)            ARCHK = 2.D0 * PI + ARCHK           00003640
      GO TO 90                                                          00003650
   80 ARCHK = 0.D0                                                      00003660
   90 ELMNTS(11) = ARCHK - ELMNTS(13) * ELMNTS(12)                      00003670
C                                                                       00003680
C     COMPUTE MEAN ANOMALY, TIME OF PERIAPSIS PASSAGE                   00003690
C                                                                       00003700
      EPM = ELMNTS(6) - ELMNTS(11) - ELMNTS(13) * ELMNTS(12)            00003710
      IF (EPM.GE.2.D0 * PI)             EPM = EPM - 2.D0 * PI           00003720
      IF (EPM.LT.0.D0)                  EPM = 2.D0 * PI + EPM           00003730
      ELMNTS(23) = EPM                                                  00003740
      ELMNTS(25) = (ELMNTS(7) * T0 - EPM)/ELMNTS(7)                     00003750
C                                                                       00003760
C     COMPUTE TRUE LONGITUDE                                            00003770
C                                                                       00003780
      EPE = ELMNTS(22) - ELMNTS(11) - ELMNTS(13) * ELMNTS(12)           00003790
      CF0 = (DCOS(EPE) - ELMNTS(9))/(1.D0 - ELMNTS(9) * DCOS(EPE))      00003800
      SF0 = (DSQRT(1.D0-ELMNTS(9)**2)*DSIN(EPE))/                       00003810
     1                              (1.D0-ELMNTS(9)*DCOS(EPE))          00003820
      EPF = DATAN2(SF0,CF0)                                             00003830
      IF (EPF.LT.0.D0)                  EPF = 2.D0 * PI + EPF           00003840
      ELMNTS(24) = EPF + ELMNTS(11) + ELMNTS(13) * ELMNTS(12)           00003850
      NO2PI = ELMNTS(24)/(2.D0 * PI)                                    00003860
      ELMNTS(24) = ELMNTS(24) - DFLOAT(NO2PI) * 2.D0 * PI               00003870
C                                                                       00003880
C     OUTPUT RESULTS FOR KEPLERIAN CASE                                 00003890
C                                                                       00003900
  100 IF (IPRNT.EQ.0)                   RETURN                          00003910
      PRINT 900,TDAYS,THRS,TMIN,TSEC,                                   00003920
     1          RV(1),RV(4),RV(2),RV(5),RV(3),RV(6)                     00003930
      IF (ISET.EQ.2)  GO TO 110                                         00003940
      EL3DEG  = ELMNTS(3) * DTR                                         00003950
      EL4DEG  = ELMNTS(4) * DTR                                         00003960
      EL5DEG  = ELMNTS(5) * DTR                                         00003970
      EL7DEG  = ELMNTS(7) * DTR                                         00003980
      EL9DEG  = ELMNTS(9) * DTR                                         00003990
```

```
      EL10DG  = ELMNTS(10)* DTR                                          00004000
      EL11DG  = ELMNTS(11)* DTR                                          00004010
      EL12DG  = ELMNTS(12)* DTR                                          00004020
      EL22DG  = ELMNTS(22)* DTR                                          00004030
      EL23DG  = ELMNTS(23)* DTR                                          00004040
      EL24DG  = ELMNTS(24)* DTR                                          00004050
      EL25DG  = ELMNTS(25)* DTR                                          00004060
      EL26DG  = ELMNTS(26)* DTR                                          00004070
      PRINT 905, ELMNTS(1)  , ELMNTS(2)  , EL3DEG    ,                   00004080
     1           EL4DEG     , EL5DEG     , EL23DG                       00004090
      IF (IFRNT.EQ.1)                  RETURN                            00004100
      PRINT 910, ELMNTS(6)  , EL10DG     , EL7DEG    ,                   00004110
     1           EL22DG     , ELMNTS(19) , EL24DG    ,                  00004120
     2           ELMNTS(17) , EL25DG     , ELMNTS(18), EL12DG           00004130
      PRINT 911, ELMNTS(8)  , EL26DG     , ELMNTS(13),                   00004140
     1           EL9DEG     , ELMNTS(16) , EL11DG    ,                  00004150
     2           ELMNTS(20) , ELMNTS(14) , ELMNTS(21), ELMNTS(15)       00004160
      RETURN                                                             00004170
C                                                                        00004180
C     OUTPUT RESULTS FOR EQUINOCTIAL CASE                                00004190
C                                                                        00004200
  110 EL6DEG  = ELMNTS(6) * DTR                                          00004210
      EL7DEG  = ELMNTS(7) * DTR                                          00004220
      EL10DG  = ELMNTS(10)* DTR                                          00004230
      EL11DG  = ELMNTS(11)* DTR                                          00004240
      EL12DG  = ELMNTS(12)* DTR                                          00004250
      EL22DG  = ELMNTS(22)* DTR                                          00004260
      EL23DG  = ELMNTS(23)* DTR                                          00004270
      EL24DG  = ELMNTS(24)* DTR                                          00004280
      IF (ELMNTS(13).GT.0.D0)                                            00004290
     1PRINT 915, ELMNTS(1)  , ELMNTS(2)  , ELMNTS(3)  ,                  00004300
     2           ELMNTS(4)  , ELMNTS(5)  , EL6DEG                        00004310
      IF (ELMNTS(13).LT.0.D0)                                            00004320
     1PRINT 920, ELMNTS(1)  , ELMNTS(2)  , ELMNTS(3)  ,                  00004330
     2           ELMNTS(4)  , ELMNTS(5)  , EL6DEG                        00004340
      IF (IFRNT.EQ.1)                  RETURN                            00004350
      PRINT 925, ELMNTS(25) , EL23DG     , EL7DEG     ,                  00004360
     1           ELMNTS(9)  , ELMNTS(19) , EL10DG     ,                  00004370
     2           ELMNTS(17) , ELMNTS(13) , ELMNTS(18) ,                  00004380
     3           EL24DG     , ELMNTS(8)  , EL22DG     ,                  00004390
     4           ELMNTS(26) , EL12DG     , ELMNTS(16) ,                  00004400
     5           EL11DG     , ELMNTS(20) , ELMNTS(14) ,                  00004410
     6           ELMNTS(21) , ELMNTS(15)                                 00004420
      RETURN                                                             00004430
C                                                                        00004440
C               FORMAT STATEMENTS                                        00004450
C                                                                        00004460
  900 FORMAT(1X,131('*')//10X,'ELAPSED TIME FROM EPOCH        : ',F4.0,  00004470
     1       ' DAYS',F4.0,' HOURS',F4.0,' MINUTES',F7.2,' SECONDS'//     00004480
     2       10X,'CURRENT POSITION VECTOR',12X,'CURRENT VELOCITY',       00004490
     3       ' VECTOR'/10X,'RX =',1PD25.15,3X,'VX =',1PD25.15/10X,       00004500
     4       'RY =',1PD25.15,3X,'VY =',1PD25.15/10X,'RZ =',1PD25.15,3X,  00004510
     5       'VZ =',1PD25.15/)                                           00004520
C                                                                        00004530
  905 FORMAT(10X,'THE KEPLERIAN ELEMENT SET ASSOCIATED WITH THE CURRE',  00004540
     1       'NT STATE IS:'/10X,'SEMIMAJOR AXIS, A',14X,':',1PD25.15/    00004550
     2      10X,'ECCENTRICITY, E',16X,':',1PD25.15/10X,'INCLINATION, ',  00004560
```

```
     3           'I',17X,':',1PD25.15/10X,'ARGUMENT OF PERIAPSIS',10X,   00004570
     4           ':',1PD25.15/10X,'LONGITUDE OF THE ASCENDING NODE:',    00004580
     5           1PD25.15/10X,'MEAN ANOMALY',19X,':',1PD25.15/)          00004590
C                                                                        00004600
  910 FORMAT(10X,'THE FOLLOWING AUXILIARY QUANTITIES ARE ALSO FOUND:'/   00004610
     1    10X,'TIME OF PERIAPSIS PASSAGE',6X,':',1PD25.15,5X,'TRUE ANO',00004620
     2    'MALY',19X,':',1PD25.15/10X,'MEAN MOTION',20X,':',            00004630
     3    1PD25.15,5X,'ECCENTRIC ANOMALY',14X,':',1PD25.15/10X,         00004640
     4    'PERIOD',25X,':',1PD25.15,5X,'RIGHT ASCENSION',16X,':',       00004650
     5    1PD25.15/10X,'PERIFOCAL RADIUS',15X,':',1PD25.15,5X,'DECLINA',00004660
     6    'TION',20X,':',1PD25.15/10X,'APOFOCAL RADIUS',16X,':',        00004670
     7    1PD25.15,5X,'TRUE LONGITUDE',17X,':',1PD25.15)                00004680
C                                                                        00004690
  911 FORMAT(10X,'SEMI-LATUS RECTUM (PARAMETER)  :',1PD25.15,5X,         00004700
     1    'MEAN LONGITUDE',17X,':',1PD25.15/10X,'ANGULAR MOMEN',        00004710
     2    'TUM MAGNITUDE',5X,':',1PD25.15,5X,'LONGITUDE OF PERIAPSIS',  00004720
     3    9X,':',1PD25.15/10X,'Z-COMPONENT OF ANGULAR MOMENTUM:',       00004730
     4    1PD25.15,5X,'ARGUMENT OF LATITUDE',11X,':',1PD25.15/          00004740
     5    10X,'ENERGY (OSCULATING)',12X,':',1PD25.15,5X,'MAGNITUDE OF ',00004750
     6    'POSITION VECTOR   :',1PD25.15/10X,'ENERGY (ZONAL PERTURBED)',00004760
     7    7X,':',1PD25.15,5X,'MAGNITUDE OF VELOCITY VECTOR   :',        00004770
     8    1PD25.15//)                                                   00004780
C                                                                        00004790
  915 FORMAT(10X,'THE DIRECT EQUINOCTIAL ELEMENT SET ASSOCIATED WITH ',  00004800
     1           'THIS STATE IS:'/10X,'A',5X,':',1PD25.15/10X,          00004810
     2           'H',5X,':',1PD25.15/10X,'K',5X,':',1PD25.15/10X,       00004820
     3           'P',5X,':',1PD25.15/10X,'Q',5X,':',1PD25.15/10X,       00004830
     4           'LAMDA :',1PD25.15/)                                   00004840
C                                                                        00004850
  920 FORMAT(10X,'THE RETROGRADE EQUINOCTIAL ELEMENT SET ASSCCIATED ',   00004860
     1           'WITH THIS STATE IS:'/10X,'A',5X,':',1PD25.15/         00004870
     2       10X,'H',5X,':',1PD25.15/10X,'K',5X,':',1PD25.15/10X,       00004880
     3           'P',5X,':',1PD25.15/10X,'Q',5X,':',1PD25.15/10X,       00004890
     4           'LAMDA :',1PD25.15/)                                   00004900
C                                                                        00004910
  925 FORMAT(10X,'THE FOLLOWING AUXILIARY QUANTITIES ARE ALSO FOUND:'/   00004920
     1    10X,'TIME OF PERIAPSIS PASSAGE',6X,':',1PD25.15,5X,'MEAN ANO',00004930
     2    'MALY',19X,':',1PD25.15/10X,'MEAN MOTION',20X,':',            00004940
     3    1PD25.15,5X,'ECCENTRICITY, E           ',5X,':',1PD25.15/10X, 00004950
     4    'PERIOD',25X,':',1PD25.15,5X,'INCLINATION, I ',16X,':',       00004960
     5    1PD25.15/10X,'PERIFOCAL RADIUS',15X,':',1PD25.15,5X,'RETROGR',00004970
     6    'ADE FACTOR',14X,':',1PD25.15/10X,'APOFOCAL RADIUS',16X,':',  00004980
     7    1PD25.15,5X,'TRUE LONGITUDE         ',8X,':',1PD25.15/        00004990
     8    10X,'SEMI-LATUS RECTUM (PARAMETER)  :',1PD25.15,5X,'ECCEN',   00005000
     9    'TRIC LONGITUDE           :',1PD25.15/10X,'ANGULAR MOMENTUM',00005010
     *    ' MAGNITUDE',5X,':',1PD25.15,5X,'LONGITUDE OF THE ASCENDING ',00005020
     *    'NODE:',1PD25.15/10X,'Z-COMPONENT OF ANGULAR MOMENTUM:',      00005030
     *    1PD25.15,5X,'ARGUMENT OF PERIAPSIS       ',2X,':',1PD25.15/  00005040
     *    10X,'ENERGY (OSCULATING)',12X,':',1PD25.15,5X,'MAGNITUDE OF ',00005050
     *    'POSITION VECTOR   :',1PD25.15/10X,'ENERGY (ZONAL PERTURBED)',00005060
     *    7X,':',1PD25.15,5X,'MAGNITUDE OF VELOCITY VECTOR   :',        00005070
     *    1PD25.15//)                                                   00005080
C                                                                        00005090
      END                                                                00005100
```

******** END OF MEMBER ELMRPT 510 RECORDS ********

```
      SUBROUTINE ELRATE(VPELMT,RETRO,GM,ELMDOT)                          00000010
C                                                                        00000020
C     VERSION OF 08/20/79     **** CHECKED OUT ****                      00000030
C                                                                        00000040
C     PURPOSE                                                            00000050
C         TO COMPUTE THE ELEMENT RATES FOR THE VARIATION OF PARAMETERS   00000060
C         FORMULATION OF THE EQUATIONS OF MOTION                         00000070
C                                                                        00000080
C     METHOD                                                             00000090
C         THE GAUSSIAN FORM OF THE VARIATION OF PARAMETERS EQUATIONS     00000100
C         OF MOTION ARE USED.  THE ELEMENT RATES ARE FOUND BY            00000110
C         MULTIPLYING THE MATRIX OF PARTIAL DERIVATIVES OF THE ELEMENTS  00000120
C         WITH RESPECT TO VELOCITY INTO THE DISTURBING ACCELERATION      00000130
C         VECTOR:                                                        00000140
C                                                                        00000150
C                                  III     III                           00000160
C                                  I         I                           00000170
C                        D A       I   ∂A    I                          00000180
C                      -------  =  I  ------  I  *  A                    00000190
C                                  I    .    I      D                   00000200
C                        DT        I   _     I                          00000210
C                                  I  ∂R     I                          00000220
C                                  I         I                          00000230
C                                  III     III                          00000240
C                                                                        00000250
C         DIMENSION:  (6X1)        (6X3)        (3X1)                    00000260
C                                                                        00000270
C     CALLING SEQUENCE                                                   00000280
C         CALL ELRATE(VPELMT,RETRO,GM,ELMDOT)                            00000290
C                                                                        00000300
C     CALLING SEQUENCE PARAMETERS                                        00000310
C         VPELMT   I   ARRAY (6X1) CONTAINING THE CURRENT ORBITAL        00000320
C                      ELEMENTS FOR WHICH THE ELEMENT RATES ARE TO       00000330
C                      BE EVALUATED.                                     00000340
C         RETRO    I   RETROGRADE FACTOR FOR EQUINOCTIAL ELEMENTS        00000350
C         GM       I   GRAVITATIONAL PARAMETER OF THE CENTRAL BODY       00000360
C         ELMDOT   O   ARRAY (6X1) CONTAINING THE ELEMENT RATES          00000370
C                                                                        00000380
C     EXTERNAL REFERENCES                                                00000390
C         AKTOAE, ELMRPT, EQELPV, FORCE, PAKTAE, PEQTX                   00000400
C                                                                        00000410
C     ANALYSIS                                                           00000420
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000430
C                                                                        00000440
C     PROGRAMMING                                                        00000450
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000460
C                                                                        00000470
C * * * * * * * * *          START PROGRAM        * * * * * * * * * * * 00000480
C                                                                        00000490
      IMPLICIT REAL*8(A-H,O-Z)                                           00000500
C                                                                        00000510
C * * * * * * * * *      DIMENSION STATEMENTS     * * * * * * * * * * * 00000520
C                                                                        00000530
      DIMENSION       VPELMT(6)  ,  ELMDOT(6)  ,  POSVEL(6)  ,  ACC(6) ,00000540
     *                DSTFRC(3)  ,  ELDUM(26)  ,  VPDUM(6)   ,          00000550
     *                PKPDV(6,3) ,  PAKDAE(6,6)                         00000560
C                                                                        00000570
```

```
C * * * * * * * * *          COMMON BLOCKS        * * * * * * * * * * * 00000580
C                                                                       00000590
      COMMON/EQPTLX/  PEQDR(6,3) ,  PEQDV(6,3) ,  PEQDX(6,6)            00000600
C                                                                       00000610
      COMMON/NPCNTL/  MORD       ,  NSUB       ,  IEQSET     ,  ITERM   00000620
C                                                                       00000630
C     CONVERT FROM THE INTEGRATION ORBIT ELEMENTS INTO POSITION AND     00000640
C     VELOCITY                                                          00000650
C                                                                       00000660
      IF (IEQSET.EQ.3)        GO TO 10                                  00000670
      CALL AKTOAE(VPELMT,RETRO,VPDUM)                                   00000680
      CALL EQELPV(VPDUM,RETRO,GM,POSVEL)                                00000690
      GO TO 20                                                          00000700
   10 CALL EQELPV(VPELMT,RETRO,GM,POSVEL)                               00000710
C                                                                       00000720
C     COMPUTE THE TOTAL ACCELERATION AT THIS VALUE OF THE STATE VECTOR  00000730
C                                                                       00000740
   20 CALL FORCE(POSVEL,GM,ACC)                                         00000750
C                                                                       00000760
C     REMOVE THE TWO BODY PART OF THE TOTAL ACCELERATION, LEAVING ONLY  00000770
C     THE DISTURBING ACCELERATION IN INERTIAL CARTESIAN COORDINATES     00000780
C                                                                       00000790
      RMAG = DSQRT(POSVEL(1)**2 + POSVEL(2)**2 + POSVEL(3)**2)          00000800
      DO  30   I = 1,3                                                  00000810
      DSTFRC(I) = ACC(I) + (GM/RMAG**3) * POSVEL(I)                     00000820
   30 CONTINUE                                                          00000830
C                                                                       00000840
C                                                                       00000850
C     COMPUTE THE AUXILIARY QUANTITIES NEEDED TO COMPUTE ∂A/∂R           00000860
C                                                                       00000870
      CALL ELMRPT(POSVEL,0.D0,0.D0,GM,2,0,ELDUM)                        00000880
C                                                                       00000890
C                                                                       00000900
C     COMPUTE ∂A/∂R                                                     00000910
C                                                                       00000920
      CALL PEQTX(POSVEL,ELDUM,2)                                        00000930
      IF (IEQSET.EQ.3)                  GO TO 80                        00000940
      CALL PAKTAE(VPDUM,RETRO,PAKDAE)                                   00000950
      DO  50   I = 1,6                                                  00000960
      DO  40   J = 1,3                                                  00000970
   40 PKPDV(I,J) = 0.D0                                                 00000980
   50 CONTINUE                                                          00000990
      DO  80   I = 1,6                                                  00001000
      DO  70   J = 1,3                                                  00001010
      DO  60   K = 1,6                                                  00001020
   60 PKPDV(I,J) = PKPDV(I,J) + PAKDAE(I,K) * PEQDV(K,J)                00001030
   70 CONTINUE                                                          00001040
   80 CONTINUE                                                          00001050
C                                                                       00001060
C     COMPUTE ELEMENT RATES FOR KEPLERIAN ELEMENTS                      00001070
C                                                                       00001080
      XN = DSQRT(GM/VPELMT(1)**3)                                       00001090
      DO  90   I = 1,6                                                  00001100
      ELMDOT(I) = 0.D0                                                  00001110
   90 CONTINUE                                                          00001120
      IF (IEQSET.EQ.3)                  GO TO 120                       00001130
      DO 110   I = 1,6                                                  00001140
```

```
ELRATE              JSS1493.PRLLEL.ORBDET.FORT

      DO 100   J = 1,3                                                  00001150
  100 ELMDOT(I) = ELMDOT(I) + PKPDV(I,J) * DSTFRC(J)                    00001160
  110 CONTINUE                                                          00001170
      ELMDOT(6) = XN + ELMDOT(6)                                        00001180
      RETURN                                                            00001190
C                                                                       00001200
C     COMPUTE ELEMENT RATES FOR EQUINOCTIAL ELEMENTS                    00001210
C                                                                       00001220
  120 DO 140   I = 1,6                                                  00001230
      DO 130   J = 1,3                                                  00001240
  130 ELMDOT(I) = ELMDOT(I) + PEQDV(I,J) * DSTFRC(J)                    00001250
  140 CONTINUE                                                          00001260
      ELMDOT(6) = XN + ELMDOT(6)                                        00001270
C                                                                       00001280
      RETURN                                                            00001290
C                                                                       00001300
      END                                                               00001310

******** END OF MEMBER ELRATE     131 RECORDS ********
```

```
EPELM                JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE EPELM(RV,T0,GM,ISET,IPRNT,ELMNTS)                          00000010
C                                                                           00000020
C     VERSION OF 09/15/79      **** CHECKED OUT ****                        00000030
C                                                                           00000040
C     PURPOSE                                                               00000050
C         TO COMPUTE THE KEPLERIAN OR EQUINOCTIAL ELEMENTS AT SOME          00000060
C         EPOCH TIME T0, GIVEN THE INERTIAL CARTESIAN COMPONENTS OF         00000070
C         POSITION AND VELOCITY AT THAT TIME.  IF ISET=1, THIS SUB-         00000080
C         ROUTINE WILL RETURN THE KEPLERIAN ELEMENTS AND CERTAIN            00000090
C         AUXILIARY QUANTITIES.  IF ISET=2, THIS SUBROUTINE WILL            00000100
C         RETURN THE NONSINGULAR EQUINOCTIAL ELEMENTS AND CERTAIN           00000110
C         AUXILIARY QUANTITIES.                                             00000120
C                                                                           00000130
C     CALLING SEQUENCE                                                      00000140
C         CALL EPELM(RV,T0,GM,ISET,IPRNT,ELMNTS)                            00000150
C                                                                           00000160
C     PARAMETERS                                                            00000170
C         RV         I    ARRAY (6X1) CONTAINING THREE INERTIAL CAR-        00000180
C                         TESIAN COORDINATES OF POSITION AND THREE          00000190
C                         INERTIAL CARTESIAN COORDINATES OF VELOCITY        00000200
C                         AT THE EPOCH TIME.                                00000210
C         T0         I    EPOCH TIME.                                       00000220
C         GM         I    GRAVITATIONAL PARAMETER.                          00000230
C         ISET       I    FLAG TO SELECT DESIRED ELEMENT SET.               00000240
C         IPRNT      I    FLAG TO INDICATE IF THE EPOCH STATE ELEMENT       00000250
C                         REPORT SHOULD BE PRINTED:                         00000260
C                         0 = DO NOT PRINT EPOCH STATE ELEMENT REPORT       00000270
C                         1 = PRINT EPOCH STATE ELEMENT REPORT              00000280
C         ELMNTS     O    ARRAY (26X1) CONTAINING APPROPRIATE ELEMENTS      00000290
C                         AND AUXILIARY QUANTITIES.                         00000300
C                         FOR ISET=1:  KEPLERIAN ELEMENTS                   00000310
C                         ELMNTS(1)  = SEMIMAJOR AXIS, A                    00000320
C                         ELMNTS(2)  = ECCENTRICITY, E                      00000330
C                         ELMNTS(3)  = INCLINATION, I                       00000340
C                         ELMNTS(4)  = ARGUMENT OF PERIAPSIS                00000350
C                         ELMNTS(5)  = LONGITUDE OF THE ASCENDING NODE      00000360
C                         ELMNTS(6)  = TIME OF PERIAPSIS PASSAGE (SEC)      00000370
C                         ELMNTS(7)  = MEAN MOTION (DEG/SEC)                00000380
C                         ELMNTS(8)  = SEMI-LATUS RECTUM (PARAMETER)        00000390
C                         ELMNTS(9)  = LONGITUDE OF PERIAPSIS               00000400
C                         ELMNTS(10) = TRUE ANOMALY AT EPOCH                00000410
C                         ELMNTS(11) = ARGUMENT OF LATITUDE AT EPOCH        00000420
C                         ELMNTS(12) = TRUE LONGITUDE AT EPOCH              00000430
C                         ELMNTS(13) = MAGNITUDE OF ANGULAR MOMENTUM        00000440
C                         ELMNTS(14) = MAGNITUDE OF POSITION VECTOR         00000450
C                         ELMNTS(15) = MAGNITUDE OF VELOCITY VECTOR         00000460
C                         ELMNTS(16) = Z-COMPONENT OF ANGULAR MOMENTUM      00000470
C                         ELMNTS(17) = PERIFOCAL RADIUS                     00000480
C                         ELMNTS(18) = APOFOCAL RADIUS                      00000490
C                         ELMNTS(19) = PERIOD                               00000500
C                         ELMNTS(20) = ENERGY (OSCULATING)                  00000510
C                         ELMNTS(21) = ENERGY (ZONAL PERTURBED)             00000520
C                         ELMNTS(22) = ECCENTRIC ANOMALY AT EPOCH           00000530
C                         ELMNTS(23) = MEAN ANOMALY AT EPOCH                00000540
C                         ELMNTS(24) = RIGHT ASCENSION                      00000550
C                         ELMNTS(25) = DECLINATION                          00000560
C                         ELMNTS(26) = MEAN LONGITUDE AT EPOCH              00000570
```

```
C                                                                       00000580
C                       FOR ISET=2:  EQUINOCTIAL ELEMENTS               00000590
C                       ELMNTS(1)  = SEMIMAJOR AXIS, A                  00000600
C                       ELMNTS(2)  = H                                  00000610
C                       ELMNTS(3)  = K                                  00000620
C                       ELMNTS(4)  = P                                  00000630
C                       ELMNTS(5)  = Q                                  00000640
C                       ELMNTS(6)  = LAMDA, MEAN LONGITUDE AT EPOCH     00000650
C                       ELMNTS(7)  = MEAN MOTION (DEG/SEC)              00000660
C                       ELMNTS(8)  = SEMI-LATUS RECTUM (PARAMETER)      00000670
C                       ELMNTS(9)  = ECCENTRICITY, E                    00000680
C                       ELMNTS(10) = INCLINATION, I                     00000690
C                       ELMNTS(11) = ARGUMENT OF PERIAPSIS              00000700
C                       ELMNTS(12) = LONGITUDE OF THE ASCENDING NODE    00000710
C                       ELMNTS(13) = RETROGRADE FACTOR                  00000720
C                       ELMNTS(14) = MAGNITUDE OF POSITION VECTOR       00000730
C                       ELMNTS(15) = MAGNITUDE OF VELOCITY VECTOR       00000740
C                       ELMNTS(16) = Z-COMPONENT OF ANGULAR MOMENTUM    00000750
C                       ELMNTS(17) = PERIFOCAL RADIUS                   00000760
C                       ELMNTS(18) = APOFOCAL RADIUS                    00000770
C                       ELMNTS(19) = PERIOD                             00000780
C                       ELMNTS(20) = ENERGY (OSCULATING)                00000790
C                       ELMNTS(21) = ENERGY (ZONAL PERTURBED)           00000800
C                       ELMNTS(22) = ECCENTRIC LONGITUDE AT EPOCH       00000810
C                       ELMNTS(23) = MEAN ANOMALY AT EPOCH              00000820
C                       ELMNTS(24) = TRUE LONGITUDE AT EPOCH            00000830
C                       ELMNTS(25) = TIME OF PERIAPSIS PASSAGE          00000840
C                       ELMNTS(26) = MAGNITUDE OF ANGULAR MOMENTUM      00000850
C                                                                       00000860
C     EXTERNAL REFERENCES                                               00000870
C         NONE                                                          00000880
C                                                                       00000890
C     ANALYSIS                                                          00000900
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000910
C                                                                       00000920
C     PROGRAMMER                                                        00000930
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000940
C                                                                       00000950
      IMPLICIT REAL*8(A-H,O-Z)                                          00000960
C                                                                       00000970
C * * * * * * * * *       DIMENSION STATEMENTS     * * * * * * * * * * 00000980
C                                                                       00000990
      DIMENSION       RV(6)     ,      ELMNTS(26)                       00001000
C                                                                       00001010
C * * * * * * * * *           COMMON BLOCKS          * * * * * * * * * 00001020
C                                                                       00001030
C     COMMON/CNSTNT/  PI        ,      TWOPI    ,     DTR    ,     RTD  00001040
C                                                                       00001050
      COMMON/ERTSHP/  REQ       ,      WE                               00001060
C                                                                       00001070
      COMMON/ZNLCOF/  SHJ(8)    ,      IZNLS                            00001080
C                                                                       00001090
      PI    = 3.1415926535897932D0                                      00001100
      DTR   = 180.D0/PI                                                 00001110
C                                                                       00001120
C     COMPUTE MAGNITUDES OF THE POSITION AND VELOCITY VECTORS AND THE   00001130
C     DOT PRODUCT OF THE POSITION VECTOR WITH THE VELOCITY VECTOR       00001140
```

```
C                                                                       00001150
      R0 = DSQRT(RV(1)**2 + RV(2)**2 + RV(3)**2)                        00001160
      ELMNTS(14) = R0                                                   00001170
      V0 = DSQRT(RV(4)**2 + RV(5)**2 + RV(6)**2)                        00001180
      ELMNTS(15) = V0                                                   00001190
      RDOTV = RV(1)*RV(4) + RV(2)*RV(5) + RV(3)*RV(6)                   00001200
C                                                                       00001210
C     COMPUTE ANGULAR MOMENTUM VECTOR COMPONENTS AND MAGNITUDE          00001220
C                                                                       00001230
      HX = RV(2)*RV(6) - RV(3)*RV(5)                                    00001240
      HY = RV(3)*RV(4) - RV(1)*RV(6)                                    00001250
      HZ = RV(1)*RV(5) - RV(2)*RV(4)                                    00001260
      ELMNTS(16) = HZ                                                   00001270
      ELMNTS(13) = DSQRT(HX**2 + HY**2 + HZ**2)                         00001280
C                                                                       00001290
C     COMPUTE ECCENTRICITY VECTOR COMPONENTS AND MAGNITUDE              00001300
C                                                                       00001310
      EX = (RV(5)*HZ - RV(6)*HY)/GM - RV(1)/R0                          00001320
      EY = (RV(6)*HX - RV(4)*HZ)/GM - RV(2)/R0                          00001330
      EZ = (RV(4)*HY - RV(5)*HX)/GM - RV(3)/R0                          00001340
      ELMNTS(2) = DSQRT(EX**2 + EY**2 + EZ**2)                          00001350
C                                                                       00001360
C     COMPUTE SEMI-LATUS RECTUM, SEMIMAJOR AXIS, MEAN MOTION            00001370
C                                                                       00001380
      ELMNTS(8) = ELMNTS(13)**2/GM                                      00001390
      ELMNTS(1) = ELMNTS(8)/(1.D0-ELMNTS(2)**2)                         00001400
      ELMNTS(7) = DSQRT(GM/ELMNTS(1) ** 3)                              00001410
C                                                                       00001420
C     COMPUTE PERIFOCAL AND APOFOCAL RADII, PERIOD, ENERGY (OSCULATING) 00001430
C                                                                       00001440
      ELMNTS(17) = ELMNTS(1) * (1.D0 - ELMNTS(2))                       00001450
      ELMNTS(18) = ELMNTS(1) * (1.D0 + ELMNTS(2))                       00001460
      ELMNTS(19) = (2.D0 * PI/ELMNTS(7))/3.6D03                         00001470
      ELMNTS(20) = -GM/(2.D0 * ELMNTS(1))                               00001480
C                                                                       00001490
C     COMPUTE ENERGY (ZONAL PERTURBED):  THIS IS A CONSTANT OF THE      00001500
C     MOTION WHEN DISTURBING FUNCTION CONSISTS ONLY OF ZONAL HARMONICS  00001510
C                                                                       00001520
      ELMNTS(21) = ELMNTS(20)                                           00001530
      IF (IZNLS.LT.2)                    GO TO 5                        00001540
      ARG   = RV(3)/R0                                                  00001550
      CFAC  = GM/R0                                                     00001560
      PKMN1 = ARG                                                       00001570
      PKMN2 = 1.D0                                                      00001580
      K = 2                                                             00001590
    1 DK = DFLOAT(K)                                                    00001600
      COFK1 = ((2.D0 * DK - 1.D0)/DK) * ARG                             00001610
      COFK2 = (DK - 1.D0)/DK                                            00001620
      PK    = COFK1 * PKMN1 - COFK2 * PKMN2                             00001630
      COF   = SHJ(K) * (REQ/R0)**K                                      00001640
      ELMNTS(21) = ELMNTS(21) + CFAC * COF * PK                         00001650
      IF (K.GE.IZNLS)                    GO TO 5                        00001660
      PKMN2 = PKMN1                                                     00001670
      PKMN1 = PK                                                        00001680
      K = K + 1                                                         00001690
      GO TO 1                                                           00001700
    5 CONTINUE                                                          00001710
```

```
C                                                                       00001720
C     COMPUTE THE ORBITAL INCLINATION                                   00001730
C                                                                       00001740
      ELMNTS(3) = DARCOS(HZ/ELMNTS(13))                                 00001750
      IF (ISET.EQ.2)  GO TO 50                                          00001760
C                                                                       00001770
C     COMPUTE UNIT VECTOR IN NODAL DIRECTION, IF NODE IS DEFINED        00001780
C                                                                       00001790
      IF (ELMNTS(3).LT.1.D-06)  GO TO 20                                00001800
      UNITN  = DSQRT(HY**2 + HX**2)                                     00001810
      UNITNX = -HY/UNITN                                                00001820
      UNITNY =  HX/UNITN                                                00001830
C                                                                       00001840
C     COMPUTE LONGITUDE OF ASCENDING NODE, IF INCLINATION IS NONZERO    00001850
C                                                                       00001860
      ELMNTS(5) = DARCOS(UNITNX)                                        00001870
      IF (UNITNY.LT.0.D0) ELMNTS(5) = 2.D0 * PI - ELMNTS(5)             00001880
C                                                                       00001890
C     COMPUTE ARGUMENT OF PERIAPSIS, I.NE.0 AND E.NE.0                  00001900
C                                                                       00001910
      IF (ELMNTS(2).LT.1.D-06)  GO TO 10                                00001920
      ELMNTS(4) = DARCOS((UNITNX*EX + UNITNY*EY)/ELMNTS(2))             00001930
      IF (EZ.LT.0.D0) ELMNTS(4) = 2.D0 * PI - ELMNTS(4)                 00001940
C                                                                       00001950
C     COMPUTE LONGITUDE OF PERIAPSIS, I.NE.0 AND E.NE.0                 00001960
C                                                                       00001970
      ELMNTS(9) = ELMNTS(4) + ELMNTS(5)                                 00001980
C                                                                       00001990
C     COMPUTE TRUE ANOMALY AT EPOCH, I.NE.0 AND E.NE.0                  00002000
C                                                                       00002010
      ELMNTS(10) = DARCOS((EX*RV(1)+EY*RV(2)+EZ*RV(3))/(R0*ELMNTS(2)))  00002020
      IF (RDOTV.LT.0.D0) ELMNTS(10) = 2.D0 * PI - ELMNTS(10)            00002030
C                                                                       00002040
C     COMPUTE ARGUMENT OF LATITUDE, TRUE LONGITUDE AT EPOCH             00002050
C                                                                       00002060
      ELMNTS(11) = ELMNTS(4) + ELMNTS(10)                               00002070
      ELMNTS(12) = ELMNTS(5) + ELMNTS(11)                               00002080
      GO TO 40                                                          00002090
C                                                                       00002100
C     BEGIN COMPUTATIONS FOR INCLINATION.NE.0 BUT ECCENTRICITY.EQ.0     00002110
C                                                                       00002120
   10 ELMNTS(4)  = 0.D0                                                 00002130
      ELMNTS(9)  = ELMNTS(5)                                            00002140
      ELMNTS(11) = DARCOS((UNITNX*RV(1) + UNITNY*RV(2))/R0)             00002150
      IF (RV(3).LT.0.D0)  ELMNTS(11) = 2.D0 * PI - ELMNTS(11)           00002160
      ELMNTS(10) = ELMNTS(11)                                           00002170
      ELMNTS(12) = ELMNTS(5) + ELMNTS(11)                               00002180
      GO TO 40                                                          00002190
C                                                                       00002200
C     BEGIN COMPUTATIONS FOR INCLINATION.EQ.0 BUT ECCENTRICITY.NE.0     00002210
C                                                                       00002220
   20 IF (ELMNTS(2).LT.1.D-06)  GO TO 30                                00002230
      ELMNTS(5)  = 0.D0                                                 00002240
      ELMNTS(4)  = DARCOS(EX/ELMNTS(2))                                 00002250
      IF (EY.LT.0.D0)  ELMNTS(4) = 2.D0 * PI - ELMNTS(4)                00002260
      ELMNTS(9)  = ELMNTS(4)                                            00002270
      ELMNTS(10) = DARCOS((EX*RV(1)+EY*RV(2)+EZ*RV(3))/(R0*ELMNTS(2)))  00002280
```

```
      IF (RDOTV.LT.0.D0)  ELMNTS(10) = 2.D0 * PI - ELMNTS(10)          00002290
      ELMNTS(11) = ELMNTS(4) + ELMNTS(10)                              00002300
      ELMNTS(12) = ELMNTS(11)                                          00002310
      GO TO 40                                                         00002320
C                                                                      00002330
C     BEGIN COMPUTATIONS FOR INCLINATION.EQ.0 AND ECCENTRICITY.EQ.0    00002340
C                                                                      00002350
   30 ELMNTS(4)  = 0.D0                                                00002360
      ELMNTS(5)  = 0.D0                                                00002370
      ELMNTS(9)  = 0.D0                                                00002380
      ELMNTS(12) = DARCOS(RV(1)/R0)                                    00002390
      IF (RV(2).LT.0.D0)  ELMNTS(12) = 2.D0 * PI - ELMNTS(12)          00002400
      ELMNTS(10) = ELMNTS(12)                                          00002410
      ELMNTS(11) = ELMNTS(12)                                          00002420
C                                                                      00002430
C     COMPUTE TIME OF PERIAPSIS PASSAGE                                00002440
C                                                                      00002450
   40 CE0 = (ELMNTS(2) + DCOS(ELMNTS(10)))/                            00002460
     1                       (1.D0 + ELMNTS(2) * DCOS(ELMNTS(10)))     00002470
      SE0 = DSQRT(1.D0 - ELMNTS(2)**2) * DSIN(ELMNTS(10))/             00002480
     1                       (1.D0 + ELMNTS(2) * DCOS(ELMNTS(10)))     00002490
      E0  = DATAN2(SE0,CE0)                                            00002500
      IF (E0.LT.0.D0) E0 = 2.D0 * PI + E0                              00002510
      ELMNTS(22) = E0                                                  00002520
      EPM = E0 - ELMNTS(2) * DSIN(E0)                                  00002530
      IF (EPM.GE.2.D0 * PI)  EPM = EPM - 2.D0 * PI                     00002540
      ELMNTS(23) = EPM                                                 00002550
      ELMNTS(6) = (ELMNTS(7) * T0 - EPM)/ELMNTS(7)                     00002560
C                                                                      00002570
C     COMPUTE RIGHT ASCENSION, DECLINATION, MEAN LONGITUDE             00002580
C                                                                      00002590
      ELMNTS(24) = DARCOS(RV(1)/DSQRT(RV(1)**2 + RV(2)**2))            00002600
      IF (RV(2).LT.0.D0)  ELMNTS(24) = 2.D0 * PI - ELMNTS(24)          00002610
      ELMNTS(25) = DARSIN(RV(3)/R0)                                    00002620
      ELMNTS(26) = ELMNTS(9) + ELMNTS(23)                              00002630
      IF  (ELMNTS(9).GE.2.D0*PI)  ELMNTS(9)  = ELMNTS(9)  - 2.D0 * PI  00002640
      IF (ELMNTS(11).GE.2.D0*PI)  ELMNTS(11) = ELMNTS(11) - 2.D0 * PI  00002650
      IF (ELMNTS(12).GE.2.D0*PI)  ELMNTS(12) = ELMNTS(12) - 2.D0 * PI  00002660
      IF (ELMNTS(26).GE.2.D0*PI)  ELMNTS(26) = ELMNTS(26) - 2.D0 * PI  00002670
      GO TO 100                                                        00002680
C                                                                      00002690
C     IF ISET=2, COMPUTE THE EQUINOCTIAL ELEMENTS                      00002700
C     COMPUTE UNIT VECTOR W                                            00002710
C                                                                      00002720
   50 ELMNTS(26) = ELMNTS(13)                                          00002730
      ELMNTS(9)  = ELMNTS(2)                                           00002740
      ELMNTS(10) = ELMNTS(3)                                           00002750
      WX = HX/ELMNTS(13)                                               00002760
      WY = HY/ELMNTS(13)                                               00002770
      WZ = HZ/ELMNTS(13)                                               00002780
C                                                                      00002790
C     COMPUTE RETROGRADE FACTOR                                        00002800
C                                                                      00002810
      IF (ELMNTS(3).GE.0.D0.AND.ELMNTS(3).LE.PI/2.D0) ELMNTS(13) = 1.D0 00002820
      IF (ELMNTS(3).GT.PI/2.D0.AND.ELMNTS(3).LE.PI)   ELMNTS(13) =-1.D0 00002830
C                                                                      00002840
C     COMPUTE EQUINOCTIAL ELEMENTS P AND Q                             00002850
```

```
C                                                                       00002860
      ELMNTS(4) =  WX/(1.D0 + ELMNTS(13) * WZ)                          00002870
      ELMNTS(5) = -WY/(1.D0 + ELMNTS(13) * WZ)                          00002880
C                                                                       00002890
C     COMPUTE UNIT VECTORS F AND G                                      00002900
C                                                                       00002910
      PQ = 1.D0 + ELMNTS(4)**2 + ELMNTS(5)**2                           00002920
      FX = (1.D0 - ELMNTS(4)**2 + ELMNTS(5)**2)/PQ                      00002930
      FY = 2.D0 * ELMNTS(4) * ELMNTS(5)/PQ                              00002940
      FZ = -2.D0 * ELMNTS(4) * ELMNTS(13)/PQ                            00002950
      GX = 2.D0 * ELMNTS(4) * ELMNTS(5) * ELMNTS(13)/PQ                 00002960
      GY = (1.D0 + ELMNTS(4)**2 - ELMNTS(5)**2) * ELMNTS(13)/PQ         00002970
      GZ = 2.D0 * ELMNTS(5)/PQ                                          00002980
C                                                                       00002990
C     COMPUTE EQUINOCTIAL ELEMENTS H AND K                              00003000
C                                                                       00003010
      ELMNTS(2) = EX*GX + EY*GY + EZ*GZ                                 00003020
      ELMNTS(3) = EX*FX + EY*FY + EZ*FZ                                 00003030
C                                                                       00003040
C     COMPUTE AUXILIARY QUANTITIES X, Y, BETA, A*SQRT(1-H**2-K**2)      00003050
C                                                                       00003060
      X    = RV(1)*FX + RV(2)*FY + RV(3)*FZ                             00003070
      Y    = RV(1)*GX + RV(2)*GY + RV(3)*GZ                             00003080
      BETA = 1.D0/(1.D0+DSQRT(1.D0-ELMNTS(2)**2-ELMNTS(3)**2))          00003090
      AHK  = ELMNTS(1) * DSQRT(1.D0-ELMNTS(2)**2-ELMNTS(3)**2)          00003100
C                                                                       00003110
C     COMPUTE COS(F), SIN(F), F, LAMDA                                  00003120
C                                                                       00003130
      CSF = ELMNTS(3) + ((1.D0+BETA*ELMNTS(3)**2)*X-ELMNTS(2)*ELMNTS(3) 00003140
     1                          *BETA*Y)/AHK                            00003150
      SNF = ELMNTS(2) + ((1.D0-BETA*ELMNTS(2)**2)*Y-ELMNTS(2)*ELMNTS(3) 00003160
     1                          *BETA*X)/AHK                            00003170
      F = DATAN2(SNF,CSF)                                               00003180
      IF (F.LT.0.D0)  F = 2.D0 * PI + F                                 00003190
      ELMNTS(6)  = F - ELMNTS(3) * SNF + ELMNTS(2) * CSF                00003200
      ELMNTS(22) = F                                                    00003210
C                                                                       00003220
C     COMPUTE ARGUMENT OF PERIAPSIS, LONGITUDE OF THE ASCENDING NODE    00003230
C                                                                       00003240
      ELMNTS(12) = DATAN2(ELMNTS(4),ELMNTS(5))                          00003250
      IF (ELMNTS(12).LT.0.D0)       ELMNTS(12) = 2.D0 * PI + ELMNTS(12) 00003260
      ARCHK      = DATAN2(ELMNTS(2),ELMNTS(3))                          00003270
      IF (ARCHK.LT.0.D0)            ARCHK      = 2.D0 * PI + ARCHK      00003280
      ELMNTS(11) = ARCHK - ELMNTS(13) * ELMNTS(12)                      00003290
C                                                                       00003300
C     COMPUTE MEAN ANOMALY AT EPOCH, TIME OF PERIAPSIS PASSAGE          00003310
C                                                                       00003320
      EPM = ELMNTS(6) - ELMNTS(11) - ELMNTS(13) * ELMNTS(12)            00003330
      IF (EPM.GE.2.D0 * PI)             EPM = EPM - 2.D0 * PI           00003340
      IF (EPM.LT.0.D0)                  EPM = 2.D0 * PI + EPM           00003350
      ELMNTS(23) = EPM                                                  00003360
      ELMNTS(25) = (ELMNTS(7) * T0 - EPM)/ELMNTS(7)                     00003370
C                                                                       00003380
C     COMPUTE TRUE LONGITUDE AT EPOCH                                   00003390
C                                                                       00003400
      EPE = ELMNTS(22) - ELMNTS(11) - ELMNTS(13) * ELMNTS(12)           00003410
      CF0 = (DCOS(EPE) - ELMNTS(9))/(1.D0 - ELMNTS(9) * DCOS(EPE))      00003420
```

```
      SF0 = (DSQRT(1.D0-ELMNTS(9)**2)*DSIN(EPE))/                       00003430
     1                                (1.D0-ELMNTS(9)*DCOS(EPE))       00003440
      EPF = DATAN2(SF0,CF0)                                             00003450
      IF (EPF.LT.0.D0)                EPF = 2.D0 * PI + EPF             00003460
      ELMNTS(24) = EPF + ELMNTS(11) + ELMNTS(13) * ELMNTS(12)           00003470
      NO2PI = ELMNTS(24)/(2.D0 * PI)                                    00003480
      ELMNTS(24) = ELMNTS(24) - (DFLOAT(NO2PI) * 2.D0 * PI)             00003490
C                                                                       00003500
C     OUTPUT RESULTS FOR KEPLERIAN CASE                                 00003510
C                                                                       00003520
  100 IF (IPRNT.EQ.0)                   RETURN                          00003530
      PRINT 900,T0,RV(1),RV(4),RV(2),RV(5),RV(3),RV(6)                  00003540
      IF (ISET.EQ.2)  GO TO 110                                         00003550
      EL3DEG  = ELMNTS(3) * DTR                                         00003560
      EL4DEG  = ELMNTS(4) * DTR                                         00003570
      EL5DEG  = ELMNTS(5) * DTR                                         00003580
      EL7DEG  = ELMNTS(7) * DTR                                         00003590
      EL9DEG  = ELMNTS(9) * DTR                                         00003600
      EL10DG  = ELMNTS(10)* DTR                                         00003610
      EL11DG  = ELMNTS(11)* DTR                                         00003620
      EL12DG  = ELMNTS(12)* DTR                                         00003630
      EL22DG  = ELMNTS(22)* DTR                                         00003640
      EL23DG  = ELMNTS(23)* DTR                                         00003650
      EL24DG  = ELMNTS(24)* DTR                                         00003660
      EL25DG  = ELMNTS(25)* DTR                                         00003670
      EL26DG  = ELMNTS(26)* DTR                                         00003680
      PRINT 905, ELMNTS(1)  , ELMNTS(2)  , EL3DEG    ,                  00003690
     1           EL4DEG     , EL5DEG     , EL23DG                       00003700
      PRINT 910, ELMNTS(6)  , EL10DG     , EL7DEG    ,                  00003710
     1           EL22DG     , ELMNTS(19) , EL24DG    ,                  00003720
     2           ELMNTS(17) , EL25DG     , ELMNTS(18), EL12DG          00003730
      PRINT 911, ELMNTS(8)  , EL26DG     , ELMNTS(13),                  00003740
     1           EL9DEG     , ELMNTS(16) , EL11DG    ,                  00003750
     2           ELMNTS(20) , ELMNTS(14) , ELMNTS(21), ELMNTS(15)     00003760
      RETURN                                                            00003770
C                                                                       00003780
C     OUTPUT RESULTS FOR EQUINOCTIAL CASE                               00003790
C                                                                       00003800
  110 EL6DEG  = ELMNTS(6) * DTR                                         00003810
      EL7DEG  = ELMNTS(7) * DTR                                         00003820
      EL10DG  = ELMNTS(10)* DTR                                         00003830
      EL11DG  = ELMNTS(11)* DTR                                         00003840
      EL12DG  = ELMNTS(12)* DTR                                         00003850
      EL22DG  = ELMNTS(22)* DTR                                         00003860
      EL23DG  = ELMNTS(23)* DTR                                         00003870
      EL24DG  = ELMNTS(24)* DTR                                         00003880
      IF (ELMNTS(13).GT.0.D0)                                           00003890
     1PRINT 915, ELMNTS(1)  , ELMNTS(2)  , ELMNTS(3)  ,                 00003900
     2           ELMNTS(4)  , ELMNTS(5)  , EL6DEG                       00003910
      IF (ELMNTS(13).LT.0.D0)                                           00003920
     1PRINT 920, ELMNTS(1)  , ELMNTS(2)  , ELMNTS(3)  ,                 00003930
     2           ELMNTS(4)  , ELMNTS(5)  , EL6DEG                       00003940
      PRINT 925, ELMNTS(25) , EL23DG     , EL7DEG     ,                 00003950
     1           ELMNTS(9)  , ELMNTS(19) , EL10DG     ,                 00003960
     2           ELMNTS(17) , ELMNTS(13) , ELMNTS(18) ,                 00003970
     3           EL24DG     , ELMNTS(8)  , EL22DG     ,                 00003980
     4           ELMNTS(26) , EL12DG     , ELMNTS(16) ,                 00003990
```

```
     5           ELLIDG     ,  ELMNTS(20) ,  ELMNTS(14) ,                  00004000
     6           ELMNTS(21) ,  ELMNTS(15)                                  00004010
      RETURN                                                               00004020
C                                                                          00004030
C               FORMAT STATEMENTS                                          00004040
C                                                                          00004050
  900 FORMAT(52X,'INITIAL STATE ELEMENT REPORT'//10X,'INITIAL TIME =',     00004060
     1   1PD25.15//10X,'INITIAL POSITION VECTOR',12X,'INITIAL VELOCITY',00004070
     2        ' VECTOR'/10X,'RX =',1PD25.15,3X,'VX =',1PD25.15/10X,       00004080
     3       'RY =',1PD25.15,3X,'VY =',1PD25.15/10X,'RZ =',1PD25.15,3X,00004090
     4       'VZ =',1PD25.15//)                                          00004100
C                                                                          00004110
  905 FORMAT(10X,'THE KEPLERIAN ELEMENT SET ASSOCIATED WITH THIS INITI',00004120
     1        'AL STATE IS:'/10X,'SEMIMAJOR AXIS, A',14X,':',1PD25.15/ 00004130
     2      10X,'ECCENTRICITY, E',16X,':',1PD25.15/10X,'INCLINATION, ',00004140
     3          'I',17X,':',1PD25.15/10X,'ARGUMENT OF PERIAPSIS',10X,   00004150
     4          ':',1PD25.15/10X,'LONGITUDE OF THE ASCENDING NODE:',    00004160
     5          1PD25.15/10X,'MEAN ANOMALY AT EPOCH',10X,':',           00004170
     6          1PD25.15//)                                             00004180
C                                                                          00004190
  910 FORMAT(10X,'THE FOLLOWING AUXILIARY QUANTITIES ARE ALSO FOUND:'/  00004200
     1    10X,'TIME OF PERIAPSIS PASSAGE',6X,':',1PD25.15,5X,'TRUE ANO',00004210
     2    'MALY AT EPOCH',10X,':',1PD25.15/10X,'MEAN MOTION',20X,':',   00004220
     3    1PD25.15,5X,'ECCENTRIC ANOMALY AT EPOCH',5X,':',1PD25.15/10X, 00004230
     4    'PERIOD',25X,':',1PD25.15,5X,'RIGHT ASCENSION',16X,':',       00004240
     5    1PD25.15/10X,'PERIFOCAL RADIUS',15X,':',1PD25.15,5X,'DECLINA',00004250
     6    'TION',20X,':',1PD25.15/10X,'APOFOCAL RADIUS',16X,':',        00004260
     7    1PD25.15,5X,'TRUE LONGITUDE AT EPOCH',8X,':',1PD25.15)        00004270
C                                                                          00004280
  911 FORMAT(10X,'SEMI-LATUS RECTUM (PARAMETER)  :',1PD25.15,5X,         00004290
     1    'MEAN LONGITUDE AT EPOCH',8X,':',1PD25.15/10X,'ANGULAR MOMEN',00004300
     2    'TUM MAGNITUDE',5X,':',1PD25.15,5X,'LONGITUDE OF PERIAPSIS',  00004310
     3    9X,':',1PD25.15/10X,'Z-COMPONENT OF ANGULAR MOMENTUM:',       00004320
     4    1PD25.15,5X,'ARGUMENT OF LATITUDE AT EPOCH',2X,':',1PD25.15/  00004330
     5    10X,'ENERGY (OSCULATING)',12X,':',1PD25.15,5X,'MAGNITUDE OF ',00004340
     6    'POSITION VECTOR   :',1PD25.15/10X,'ENERGY (ZONAL PERTURBED)',00004350
     7    7X,':',1PD25.15,5X,'MAGNITUDE OF VELOCITY VECTOR   :',        00004360
     8    1PD25.15///)                                                  00004370
C                                                                          00004380
  915 FORMAT(10X,'THE DIRECT EQUINOCTIAL ELEMENT SET ASSOCIATED WITH ', 00004390
     1          'THIS INITIAL STATE IS:'/10X,'A',5X,':',1PD25.15/10X,  00004400
     2          'H',5X,':',1PD25.15/10X,'K',5X,':',1PD25.15/10X,       00004410
     3          'P',5X,':',1PD25.15/10X,'Q',5X,':',1PD25.15/10X,       00004420
     4          'LAMDA :',1PD25.15//)                                  00004430
C                                                                          00004440
  920 FORMAT(10X,'THE RETROGRADE EQUINOCTIAL ELEMENT SET ASSOCIATED ',  00004450
     1          'WITH THIS INITIAL STATE IS:'/10X,'A',5X,':',1PD25.15/ 00004460
     2      10X,'H',5X,':',1PD25.15/10X,'K',5X,':',1PD25.15/10X,       00004470
     3          'P',5X,':',1PD25.15/10X,'Q',5X,':',1PD25.15/10X,       00004480
     4          'LAMDA :',1PD25.15//)                                  00004490
C                                                                          00004500
  925 FORMAT(10X,'THE FOLLOWING AUXILIARY QUANTITIES ARE ALSO FOUND:'/  00004510
     1    10X,'TIME OF PERIAPSIS PASSAGE',6X,':',1PD25.15,5X,'MEAN ANO',00004520
     2    'MALY AT EPOCH',10X,':',1PD25.15/10X,'MEAN MOTION',20X,':',   00004530
     3    1PD25.15,5X,'ECCENTRICITY, E          ',5X,':',1PD25.15/10X, 00004540
     4    'PERIOD',25X,':',1PD25.15,5X,'INCLINATION, I ',16X,':',       00004550
     5    1PD25.15/10X,'PERIFOCAL RADIUS',15X,':',1PD25.15,5X,'RETROGR',00004560
```

```
EPELM                JSS1493.PRLLEL.ORBDET.FORT

     6    'ADE FACTOR',14X,':',1PD25.15/10X,'APOFOCAL RADIUS',16X,':',  00004570
     7    1PD25.15,5X,'TRUE LONGITUDE AT EPOCH',8X,':',1PD25.15/        00004580
     8    10X,'SEMI-LATUS RECTUM (PARAMETER)  :',1PD25.15,5X,'ECCEN',   00004590
     9    'TRIC LONGITUDE AT EPOCH    :',1PD25.15/10X,'ANGULAR MOMENTUM',00004600
     *    ' MAGNITUDE',5X,':',1PD25.15,5X,'LONGITUDE OF THE ASCENDING ',00004610
     *    'NODE:',1PD25.15/10X,'Z-COMPONENT OF ANGULAR MOMENTUM:',      00004620
     *    1PD25.15,5X,'ARGUMENT OF PERIAPSIS          ',2X,':',1PD25.15/  00004630
     *    10X,'ENERGY (OSCULATING)',12X,':',1PD25.15,5X,'MAGNITUDE OF ',00004640
     *    'POSITION VECTOR    :',1PD25.15/10X,'ENERGY (ZONAL PERTURBED)',00004650
     *    7X,':',1PD25.15,5X,'MAGNITUDE OF VELOCITY VECTOR    :',       00004660
     *    1PD25.15///)                                                  00004670
C                                                                       00004680
      END                                                               00004690

******** END OF MEMBER EPELM      469 RECORDS ********
```

```
EPHEX                JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE EPHEX                                                  00000010
C                                                                       00000020
C     VERSION OF 09/15/79      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO CONTROL THE EXECUTION OF AN EPHEMERIS GENERATION RUN,      00000060
C         USING THE PICARD-CHEBYSHEV INTEGRATOR                         00000070
C                                                                       00000080
C     CALLING SEQUENCE                                                  00000090
C         CALL EPHEX                                                    00000100
C                                                                       00000110
C     CALLING SEQUENCE PARAMETERS                                       00000120
C         NONE                                                          00000130
C                                                                       00000140
C     EXTERNAL REFERENCES                                               00000150
C         SUBROUTINES                                                   00000160
C              AKTOAE                                                   00000170
C              CSRSTA                                                   00000180
C              ELMRPT                                                   00000190
C              EQELPV                                                   00000200
C              PXTSLV                                                   00000210
C              VTEST                                                    00000220
C                                                                       00000230
C         COMMON BLOCKS                                                 00000240
C              /ERTSHP/                                                 00000250
C              /ICNTRL/                                                 00000260
C              /NPCNTL/                                                 00000270
C              /STAPAR/                                                 00000280
C              /STATE /                                                 00000290
C              /TIMER /                                                 00000300
C                                                                       00000310
C     ANALYSIS                                                          00000320
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000330
C                                                                       00000340
C     PROGRAMMING                                                       00000350
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000360
C                                                                       00000370
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000380
C                                                                       00000390
      IMPLICIT REAL*8(A-H,O-Z)                                          00000400
C                                                                       00000410
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000420
C                                                                       00000430
      DIMENSION       STA(6)        ,  POSVEL(6)    ,  ELDUM(6)       , 00000440
     1                EL(26)                                            00000450
C                                                                       00000460
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000470
C                                                                       00000480
      COMMON/ERTSHP/  REQ           ,  WE           ,  FINV           , 00000490
     1                SPHINF        ,  GM                               00000500
C                                                                       00000510
      COMMON/ICNTRL/  IRNTYP        ,  IVINIT       ,  IVINTG         , 00000520
     1                ISLVFR        ,  NSLVFR       ,  IPTLS          , 00000530
     2                IJ2RTS        ,  IRVPAR       ,  IPARRV         , 00000540
     3                IPRNT1        ,  IPRNT2       ,  IPRNT3         , 00000550
     4                IPRNT4        ,  IPRNT5       ,  IOUT           , 00000560
     5                ICGCMP        ,  ISCID        ,  NPAR             00000570
```

```
C                                                                       00000580
      COMMON/NPCNTL/  MORD          ,  NSUB          ,  IEQSET          , 00000590
     1                ITERM                                             00000600
C                                                                       00000610
      COMMON/STAPAR/  PXDAEO(6,6)   ,  PXDAKO(6,6)   ,  PXDXO(6,6)     00000620
C                                                                       00000630
      COMMON/STATE /  SO(6)         ,  STK(70,6)     ,  STK1(70,6)     00000640
C                                                                       00000650
      COMMON/TIMER /  TO            ,  TSPAN         ,  TOUT           , 00000660
     1                H             ,  TINC                            00000670
C                                                                       00000680
C     INITIALIZE TIMES FOR FIRST INTEGRATION STEP                       00000690
C                                                                       00000700
      TO   = 0.DO                                                       00000710
      TL   = TO                                                         00000720
      TU   = H                                                          00000730
      TOUT = TL                                                         00000740
      N1   = NSUB + 1                                                   00000750
C                                                                       00000760
C     CALL THE PICARD-CHEBYSHEV INTEGRATOR FOR THIS INTEGRATION STEP    00000770
C                                                                       00000780
   10 CALL VTEST(TL,TU,GM)                                              00000790
C                                                                       00000800
C     INTERPOLATE FOR THE STATE AT ALL OUTPUT TIMES WITHIN THIS         00000810
C     INTEGRATION STEP                                                  00000820
C                                                                       00000830
   20 CALL CSRSTA(STA,TU,TOUT,H,1)                                      00000840
      GO TO (30,40,50) , IEQSET                                         00000850
C                                                                       00000860
C     OUTPUT FOR COWELL EQUATIONS                                       00000870
C                                                                       00000880
   30 CALL ELMRPT(STA,TO,TOUT,GM,ICUT,IPRNT1,EL)                        00000890
      GO TO 60                                                          00000900
C                                                                       00000910
C     OUTPUT FOR KEPLERIAN VARIATION OF PARAMETERS EQUATIONS            00000920
C                                                                       00000930
   40 CALL AKTOAE(STA,RETRO,GM,ELDUM)                                   00000940
      CALL EQELPV(ELDUM,RETRO,GM,POSVEL)                                00000950
      CALL ELMRPT(POSVEL,TO,TOUT,GM,IOUT,IPRNT1,EL)                     00000960
      GO TO 60                                                          00000970
C                                                                       00000980
C     OUTPUT FOR EQUINOCTIAL VARIATION OF PARAMETERS EQUATIONS          00000990
C                                                                       00001000
   50 RETRO = 1.DO                                                      00001010
      CALL EQELPV(STA,RETRO,GM,POSVEL)                                  00001020
      CALL ELMRPT(POSVEL,TO,TOUT,GM,IOUT,IPRNT1,EL)                     00001030
C                                                                       00001040
C     COMPUTE AND OUTPUT ANALYTICAL PARTIALS, IF REQUIRED               00001050
C                                                                       00001060
   60 IF (IPTLS.EQ.0)                    GO TO 70                       00001070
      IF (IEQSET.NE.1)                   GO TO 65                       00001080
      IF (IOUT.NE.2)          CALL ELMRPT(STA,TO,TOUT,GM,2,0,EL)        00001090
      CALL PXTSLV(STA,EL,TOUT,TO)                                       00001100
      GO TO 70                                                          00001110
   65 IF (IOUT.NE.2)          CALL ELMRPT(POSVEL,TO,TOUT,GM,2,0,EL)     00001120
      CALL PXTSLV(POSVEL,EL,TOUT,TO)                                    00001130
   70 CONTINUE                                                          00001140
```

```
EPHEX            .  JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00001150
C     UPDATE OUTPUT TIME; RETURN TO INTERPOLATOR IF THE END OF THE      00001160
C     INTEGRATION STEP HAS NOT BEEN REACHED                             00001170
C                                                                       00001180
      TOUT = TOUT + TINC                                                00001190
      IF (TOUT.LE.TU)                   GO TO 20                        00001200
C                                                                       00001210
C     IF THE END OF THE SPAN HAS NOT BEEN REACHED, TAKE ANOTHER STEP    00001220
C                                                                       00001230
      IF (TSPAN - TU)                   120,120,80                      00001240
   80 IF (DABS(TSPAN - TU).GE.H)        GO TO 90                        00001250
      TL = TU                                                           00001260
      TU = TSPAN                                                        00001270
      GO TO 100                                                         00001280
   90 TL = TU                                                           00001290
      TU = TU + H                                                       00001300
  100 DO  110  I = 1,6                                                  00001310
      S0(I) = STK1(N1,I)                                                00001320
  110 CONTINUE                                                          00001330
      GO TO 10                                                          00001340
C                                                                       00001350
  120 RETURN                                                            00001360
C                                                                       00001370
      END                                                               00001380

******** END OF MEMBER EPHEX      138 RECORDS ********
```

EQELPV JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE EQELPV(AE,RETRO,GM,ST)                                 00000010
C                                                                       00000020
C     VERSION OF 08/27/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO CONVERT A SET OF EQUINOCTIAL ORBITAL ELEMENTS TO INERTIAL  00000060
C         CARTESIAN POSITION AND VELOCITY                               00000070
C                                                                       00000080
C     CALLING SEQUENCE                                                  00000090
C         CALL EQELPV(AE,RETRO,GM,ST)                                   00000100
C                                                                       00000110
C     CALLING SEQUENCE PARAMETERS                                       00000120
C         AE        I   ARRAY (6X1) OF EQUINOCTIAL ORBITAL ELEMENTS     00000130
C                         AE(1) = A, SEMIMAJOR AXIS                     00000140
C                         AE(2) = H                                     00000150
C                         AE(3) = K                                     00000160
C                         AE(4) = P                                     00000170
C                         AE(5) = Q                                     00000180
C                         AE(6) = LAMDA, MEAN LONGITUDE                 00000190
C         RETRO     I   RETROGRADE FACTOR                               00000200
C         GM        I   GRAVITATIONAL PARAMETER FOR CENTRAL BODY        00000210
C         ST        O   ARRAY (6X1) CONTAINING THE CORRESPONDING        00000220
C                       POSITION AND VELOCITY COMPONENTS                00000230
C                                                                       00000240
C     EXTERNAL REFERENCES                                               00000250
C         KEPLR                                                         00000260
C                                                                       00000270
C     ANALYSIS                                                          00000280
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000290
C                                                                       00000300
C     PROGRAMMING                                                       00000310
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000320
C                                                                       00000330
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000340
C                                                                       00000350
      IMPLICIT REAL*8(A-H,O-Z)                                          00000360
C                                                                       00000370
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000380
C                                                                       00000390
      DIMENSION       AE(6)   ,   ST(6)   ,   F(3)     ,   G(3)         00000400
C                                                                       00000410
C     COMPUTE THE TRUE LONGITUDE FROM H AND K                           00000420
C                                                                       00000430
      PI = 3.141592653589793200                                         00000440
      IF (AE(2).EQ.0.D0.AND.AE(3).EQ.0.D0)        GO TO 10              00000450
      WPIOM = DATAN2(AE(2),AE(3))                                       00000460
      IF (WPIOM.LT.0.D0)                WPIOM = 2.D0 * PI + WPIOM       00000470
      GO TO 20                                                          00000480
   10 WPIOM = 0.D0                                                      00000490
   20 ANOMM = AE(6) - WPIOM                                             00000500
      IF (ANOMM.LT.0.D0)                ANOMM = 2.D0 * PI + ANOMM       00000510
      ECC   = DSQRT(AE(2)**2 + AE(3)**2)                                00000520
      CALL KEPLR(ANOMM,ECC,1.D-10,ANOME,ANOMF)                          00000530
      TRULNG = ANOMF + WPIOM                                            00000540
      IF (TRULNG.GE.2.D0*PI)            TRULNG = TRULNG - 2.D0 * PI     00000550
C                                                                       00000560
C     COMPUTE THE MAGNITUDE OF THE POSITION VECTOR                      00000570
```

```
EQELPV              JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000580
      H2K2 = 1.D0 - AE(2)**2 - AE(3)**2                                 00000590
      RMAG = (AE(1)*H2K2)/(1.D0+AE(3)*DCOS(TRULNG)+AE(2)*DSIN(TRULNG))  00000600
C                                                                       00000610
C                                                                       00000620
C     COMPUTE THE UNIT VECTORS F AND G, EXPRESSED IN INERTIAL           00000630
C     CARTESIAN COORDINATES                                             00000640
C                                                                       00000650
      PQ = 1.D0 + AE(4)**2 + AE(5)**2                                   00000660
      F(1) =  (1.D0 - AE(4)**2 + AE(5)**2)/PQ                           00000670
      F(2) =  (2.D0 * AE(4) * AE(5))/PQ                                 00000680
      F(3) = -(2.D0 * AE(4) * RETRO)/PQ                                 00000690
      G(1) =  (2.D0 * AE(4) * AE(5) * RETRO)/PQ                         00000700
      G(2) =  ((1.D0 + AE(4)**2 - AE(5)**2) * RETRO)/PQ                 00000710
      G(3) =  (2.D0 * AE(5))/PQ                                         00000720
C                                                                       00000730
C                    . .                                                00000740
C     COMPUTE X, Y, X, Y                                                00000750
C                                                                       00000760
      XN =  DSQRT(GM/AE(1)**3)                                          00000770
      X  =  RMAG * DCOS(TRULNG)                                         00000780
      Y  =  RMAG * DSIN(TRULNG)                                         00000790
      XD = -(XN * AE(1) * (AE(2) + DSIN(TRULNG)))/DSQRT(H2K2)           00000800
      YD =  (XN * AE(1) * (AE(3) + DCOS(TRULNG)))/DSQRT(H2K2)           00000810
C                                                                       00000820
C     COMPUTE POSITION AND VELOCITY VECTORS IN INERTIAL COORDINATES     00000830
C                                                                       00000840
      DO  30   I = 1,3                                                  00000850
      ST(I)   = X  * F(I) + Y  * G(I)                                   00000860
      ST(I+3) = XD * F(I) + YD * G(I)                                   00000870
   30 CONTINUE                                                          00000880
      RETURN                                                            00000890
      END                                                               00000900

******** END OF MEMBER EQELPV      90 RECORDS ********
```

```
EQPARM              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK EQPARM   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/EQPARM/   FGW(3,3)   ,   YAUX(3)   ,   PQ   ,   BETA   ,  00000120
     *                 XX         ,   YY        ,   XXD  ,   YYD    ,  00000122
     *                 AHK        ,   GG                               00000124
C                                                                       00000130
C                      #LOCA-                                           00000140
C             DIMEN-    TION                                           00000150
C  VARIABLE    SION     (REAL)              DESCRIPTION                 00000160
C  ________   _____    _______  ____________________________________   00000170
C                                                                       00000180
C       FGW      3,3         1  ARRAY (3X3) IN WHICH THE COLUMNS CON-  00000190
C                               TAIN THE COMPONENTS OF THE UNIT VECTORS 00000200
C                               F, G AND W.                             00000210
C      YAUX        3         10  ARRAY (3X1) CONTAINING THE AUXILIARY   00000215
C                               VECTOR Y NEEDED FOR THE COMPUTATION OF  00000220
C                               CERTAIN PARTIAL DERIVATIVES:  SEE       00000230
C                               CHAPTER 3 OF "FORMULATION AND EVALUA-   00000240
C                               TION OF PARALLEL ALGORITHMS FOR THE     00000245
C                               ORBIT DETERMINATION PROBLEM."           00000250
C        PQ         1         13  1 + P**2 + Q**2                      00000260
C      BETA         1         14  1/(1 + SQRT(1 - H**2 - K**2) )       00000270
C        XX         1         15  COMPONENT OF THE POSITION VECTOR IN THE00000280
C                               DIRECTION OF ORBIT PLANE UNIT VECTOR F. 00000290
C        YY         1         16  COMPONENT OF THE POSITION VECTOR IN THE00000300
C                               DIRECTION OF ORBIT PLANE UNIT VECTOR G. 00000310
C       XXD         1         17  COMPONENT OF THE VELOCITY VECTOR IN THE00000320
C                               DIRECTION OF ORBIT PLANE UNIT VECTOR F. 00000330
C       YYD         1         18  COMPONENT OF THE VELOCITY VECTOR IN THE00000340
C                               DIRECTION OF ORBIT PLANE UNIT VECTOR G. 00000350
C       AHK         1         19  A * SQRT(1 - H**2 - K**2)             00000360
C        GG         1         20  MAGNITUDE OF ANGULAR MOMENTUM VECTOR. 00000370
C                                                                       00000380
      DATA FGW    / 9*0.D0      /                                       00000390
C                                                                       00000400
      DATA YAUX   / 3*0.D0      /                                       00000410
C                                                                       00000420
      DATA PQ     / 0.D0        /                                       00000430
C                                                                       00000440
      DATA BETA   / 0.D0        /                                       00000450
C                                                                       00000460
      DATA XX     / 0.D0        /                                       00000470
C                                                                       00000480
      DATA YY     / 0.D0        /                                       00000490
C                                                                       00000500
      DATA XXD    / 0.D0        /                                       00000510
C                                                                       00000520
      DATA YYD    / 0.D0        /                                       00000530
```

```
EQPARM              JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000540
      DATA AHK    / 0.D0          /                                     00000550
C                                                                       00000560
      DATA GG     / 0.D0          /                                     00000570
C                                                                       00000580
      END                                                               00000590

******** END OF MEMBER EQPARM      63 RECORDS ********
```

```
EQPTLX              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK EQPARM   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/EQPARM/   PEQDR(6,3)   ,   PEQDV(6,3)   ,   PEQDX(6,6)    00000120
C                                                                       00000130
C                      #LOCA-                                           00000140
C             DIMEN-     TION                                           00000150
C  VARIABLE    SION     (REAL)                 DESCRIPTION              00000160
C  ________   ______   _______   ____________________________________   00000170
C                                                                       00000180
C     PEQDR      6,3         1   ARRAY (6X3) CONTAINING THE PARTIAL DE- 00000190
C                                RIVATIVES OF THE EQUINOCTIAL ORBITAL   00000200
C                                ELEMENTS WITH RESPECT TO THE POSITION  00000210
C                                VECTOR.                                00000215
C     PEQDV      6,3        19   ARRAY (6X3) CONTAINING THE PARTIAL DE- 00000220
C                                RIVATIVES OF THE EQUINOCTIAL ORBITAL   00000230
C                                ELEMENTS WITH RESPECT TO THE VELOCITY  00000240
C                                VECTOR.                                00000250
C     PEQDX      6,6        37   ARRAY (6,6) CONTAINING THE PARTIAL DE- 00000260
C                                RIVATIVES OF THE EQUINOCTIAL ORBITAL   00000270
C                                ELEMENTS WITH RESPECT TO THE SPACECRAFT00000280
C                                STATE VECTOR X.                        00000290
C                                                                       00000300
      DATA PEQDR  / 18*0.D0      /                                      00000310
C                                                                       00000320
      DATA PEQDV  / 18*0.D0      /                                      00000330
C                                                                       00000340
      DATA PEQDX  / 36*0.D0      /                                      00000350
C                                                                       00000360
      END                                                               00000370

******** END OF MEMBER EQPTLX      38 RECORDS ********
```

```
ERTSHP              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 08/07/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK ERTSHP  00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
C * * * * * * * * *            /ERTSHP/          * * * * * * * * * * * 00000110
C                                                                       00000120
      IMPLICIT REAL*8(A-H,O-Z)                                          00000130
      COMMON/ERTSHP/  REQ     ,  WE      ,  FINV    ,  SPHINF  ,  GM    00000140
C                                                                       00000150
C                              #LOCA-                                   00000160
C                 DIMEN-        TION                                    00000170
C  VARIABLE        SION        (REAL)              DESCRIPTION          00000180
C  ________       _____       _______   ____________________________   00000190
C                                                                       00000200
C  REQ                1           1    MEAN EQUATORIAL RADIUS OF EARTH (KM)   00000210
C  WE                 1           2    EARTH ROTATION RATE (RAD/SEC)          00000220
C  FINV               1           3    INVERSE FLATTENING COEFFICIENT         00000230
C  SPHINF             1           4    RADIUS OF THE SPHERE OF INFLUENCE (KM) 00000240
C  GM                 1           5    GRAVITATIONAL PARAMETER (KM**3/SEC**2) 00000250
C                                                                       00000260
      DATA REQ      /0.63781400D 04/                                    00000270
C                                                                       00000280
      DATA WE       /0.72921159D-04/                                    00000290
C                                                                       00000300
      DATA FINV     /0.29825000D 03/                                    00000310
C                                                                       00000320
      DATA SPHINF   /0.92500000D 06/                                    00000330
C                                                                       00000340
      DATA GM       /0.39860080D 06/                                    00000350
C                                                                       00000360
      END                                                               00000370

******** END OF MEMBER ERTSHP      37 RECORDS ********
```

```
ESTEX                JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE ESTEX                                                  00000010
C                                                                       00000020
C     VERSION OF 09/15/79      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO ESTIMATE, USING A NONLINEAR FUNCTION MINIMIZATION          00000060
C         TECHNIQUE, THE VECTOR 'XMIN' WHICH MINIMIZES SOME SCALAR      00000070
C         FUNCTIONAL OF THE VECTOR 'X'.  THIS SUBROUTINE USES A         00000080
C         PARALLEL VARIABLE METRIC (PVM) OPTIMIZATION ALGORITHM         00000090
C         (GRADIENT DEPENDENT) DUE TO STRAETER.  THIS METHOD IS         00000100
C         AMENABLE TO EXECUTION USING AN ARRAY OF PARALLEL PROCESSORS,  00000110
C         SUCH AS A SINGLE INSTRUCTION STREAM/MULTIPLE DATA STREAM       00000120
C         (SIMD) COMPUTER ARCHITECTURE.                                 00000130
C                                                                       00000140
C     METHOD                                                            00000150
C         FOR PARALLEL VARIABLE METRIC (PVM) ALGORITHM SEE REFERENCE:   00000160
C                                                                       00000170
C              STRAETER, T., "A PARALLEL VARIABLE METRIC OPTIMIZATION    00000180
C              ALGORITHM," NASA TN D-7329, DECEMBER 1973.               00000190
C                                                                       00000200
C     CALLING SEQUENCE                                                  00000210
C         CALL ESTEX                                                    00000220
C                                                                       00000230
C     CALLING SEQUENCE PARAMETERS                                       00000240
C         NONE                                                          00000250
C                                                                       00000260
C         INPUT VIA COMMON/XESTMT/:                                     00000270
C         XAP       I         VECTOR (6X1) CONTAINING THE 'A PRIORI'    00000280
C                             ESTIMATE OF THE SOLVE-FOR VECTOR X.       00000290
C         SCLFAC    I         VECTOR (6X1) CONTAINING THE SET OF        00000300
C                             SCALE FACTORS TO BE APPLIED TO EACH COM-  00000310
C                             PONENT OF THE SCHAUDER BASIS SPANNING THE 00000320
C                             PARAMETER SPACE.                          00000330
C         EPSILN    I         CONVERGENCE CRITERION FOR THE PVM METHOD. 00000340
C                                                                       00000350
C         INPUT VIA COMMON/ICNTRL/:                                     00000360
C         NSLV      I         NUMBER OF SOLVE-FOR PARAMETERS/DIMENSION  00000370
C                             OF THE PARAMETER SPACE.                   00000380
C         NPAR      I         DEGREE OF PARALLELISM IN THE ASSUMED      00000390
C                             COMPUTER ARCHITECTURE.                    00000400
C                                                                       00000410
C         OUTPUT VIA COMMON/XESTMT/:                                    00000420
C         XMIN      O         VECTOR (6X1) CONTAINING THE ESTIMATE OF   00000430
C                             THE VECTOR X THAT MINIMIZES THE GIVEN COST00000440
C                             FUNCTIONAL.                               00000450
C                                                                       00000460
C     EXTERNAL REFERENCES                                               00000470
C         SUBROUTINES                                                   00000480
C              CGINTG     PTLOUT                                        00000490
C              CGLINE     SERCH1                                        00000500
C              COVMTX                                                   00000510
C              INPROD                                                   00000520
C              OBSOUT                                                   00000530
C              OUTPRD                                                   00000540
C                                                                       00000550
C                                                                       00000560
C         COMMON BLOCKS                                                 00000570
```

```
C             /ICNTRL/                                                  00000580
C             /ISTSTC/                                                  00000590
C             /XESTMT/                                                  00000600
C                                                                       00000610
C     ANALYSIS                                                          00000620
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000630
C                                                                       00000640
C     PROGRAMMING                                                       00000650
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000660
C                                                                       00000670
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000680
C                                                                       00000690
      IMPLICIT REAL*8(A-H,O-Z)                                          00000700
C                                                                       00000710
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000720
C                                                                       00000730
      DIMENSION       BASIS(6,6)      , VMTRIC(6,6)     , YMAT(6,6)     , 00000740
     *                XGNTH(6)        , XJATXJ(6)       , BMAT(6,6)     , 00000750
     *                XGATXJ(6,6)     , SRCH(6)         , RMAT(6,6)     , 00000760
     *                RDOTY(6,6)      , TAU(6)          , XMAT(6,6)     , 00000770
     *                XN(6)                                               00000780
C                                                                       00000790
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000800
C                                                                       00000810
      COMMON/ICNTRL/  IRNTYP          , IVINIT          , IVINTG        , 00000820
     *                ISLVFR          , NSLVFR          , IPTLS         , 00000830
     *                IJ2RTS          , IRVPAR          , IPARRV        , 00000840
     *                IPRNT1          , IPRNT2          , IPRNT3        , 00000850
     *                IPRNT4          , IPRNT5          , ICUT          , 00000860
     *                ICGCMP          , ISCID           , NPAR          , 00000870
     *                ICOV                                                00000880
C                                                                       00000890
      COMMON/ISTSTC/  ITCNT           , IF1CNT          , IG1CNT        , 00000900
     *                IF2CNT          , IG2CNT          , ICCNT         , 00000910
     *                IFG1WO          , IFG2WO          , IFG1W         , 00000920
     *                IFG2W           , IT1WO           , IT2WO         , 00000930
     *                IT1W            , IT2W                              00000940
C                                                                       00000950
      COMMON/XESTMT/  XAP(6)          , SCLFAC(6)       , EPSILN        , 00000960
     *                ALFA1           , STPSZ           , C1DIM         , 00000970
     *                CMTXAP(6,6)     , XMIN(6)                           00000980
C                                                                       00000990
C * * * * * * * * *    EQUIVALENCE STATEMENTS       * * * * * * * * * * 00001000
C                                                                       00001010
C     FROM COMMON/ICNTRL/:                                              00001020
      EQUIVALENCE     ( NSLVFR  ,  NSLV    )                            00001030
C                                                                       00001040
C     ESTABLISH SET OF BASIS VECTORS FOR THE SPACE OF SOLVE-FOR PARA-   00001050
C     METERS; SET THE ZEROTH ITERATION OF THE METRIC MATRIX EQUAL TO    00001060
C     THE IDENTITY MATRIX.                                              00001070
C                                                                       00001080
      DO  20   JCOL = 1,NSLV                                            00001090
      DO  10   IROW = 1,NSLV                                            00001100
      BASIS(IROW,JCOL)  = 0.D0                                          00001110
   10 VMTRIC(IROW,JCOL) = 0.D0                                          00001120
      VMTRIC(JCOL,JCOL) = 1.D0                                          00001130
      BASIS(JCOL,JCOL) = SCLFAC(JCOL) * EPSILN                          00001140
```

```
   20 CONTINUE                                                          00001150
      WRITE(6,700)                                                      00001160
      WRITE(6,710)                                                      00001170
      DO  2000 JCOL = 1,NSLV                                            00001180
      WRITE(6,720)  JCOL                                                00001190
      WRITE(6,730)  (BASIS(IROW,JCOL),IROW = 1,NSLV)                    00001200
 2000 CONTINUE                                                          00001210
      WRITE(6,750)  NSLV                                                00001220
      WRITE(6,760)  NPAR                                                00001230
      WRITE(6,770)  EPSILN                                              00001240
      WRITE(6,780)                                                      00001250
      DO  2500 IROW = 1,NSLV                                            00001260
      WRITE(6,1310) IROW,SCLFAC(IROW)                                   00001270
 2500 CONTINUE                                                          00001280
C                                                                       00001290
C     INITIALIZE CERTAIN VARIABLES FROM COMMON/ISTSTC/.                 00001300
C                                                                       00001310
      ITCNT = 0                                                         00001320
      ICCNT = 0                                                         00001330
      IT1WO = 0                                                         00001340
      IT2WO = 0                                                         00001350
      IT1W  = 0                                                         00001360
      IT2W  = 0                                                         00001370
C                                                                       00001380
C     BEGIN PARALLEL VARIABLE METRIC (PVM) ALGORITHM.                   00001390
C                                                                       00001400
C     STEP 1(A):                                                        00001410
C     COMPUTE THE COST FUNCTIONAL, ITS GRADIENT (G) AND G DOT G AT THE  00001420
C     CURRENT ITERATE OF THE SOLVE-FOR VECTOR X; CHECK IF GDOTG.LE.     00001430
C     CONVERGENCE CRITERION SQUARED; IF SO, CONVERGENCE HAS BEEN        00001440
C     ACHIEVED AND METHOD IS TERMINATED.                                00001450
C                                                                       00001460
      IF1CNT = 0                                                        00001470
      IF2CNT = 0                                                        00001480
      IG1CNT = 0                                                        00001490
      IG2CNT = 0                                                        00001500
      CSTLST = 0.D0                                                     00001510
   30 JSAVE = 100                                                       00001520
      CALL CGINTG(XAP,COST,XGNTH)                                       00001530
      IF1CNT = IF1CNT + 1                                               00001540
      IF2CNT = IF2CNT + 1                                               00001550
      IG1CNT = IG1CNT + 1                                               00001560
      IG2CNT = IG2CNT + 1                                               00001570
      CALL INPRD1(XGNTH,XGNTH,GDOTG,BASIS)                              00001580
      WRITE(6,790)  ITCNT                                               00001590
      WRITE(6,800)                                                      00001600
      WRITE(6,730)  (XAP(IROW),IROW = 1,NSLV)                           00001610
      WRITE(6,810)  COST                                                00001620
      WRITE(6,820)                                                      00001630
      WRITE(6,730)  (XGNTH(IROW),IROW = 1,NSLV)                         00001640
      IF (IPRNT4.EQ.1)                  CALL OBSOUT                     00001650
      IF (IPRNT5.EQ.1)                  CALL PTLOUT                     00001660
      WRITE(6,830)  GDOTG                                               00001670
      WRITE(6,840)                                                      00001680
      DO  3000 IROW = 1,NSLV                                            00001690
      WRITE(6,850)  IROW                                                00001700
      WRITE(6,730)  (VMTRIC(IROW,JCOL),JCOL = 1,NSLV)                   00001710
```

```
 3000 CONTINUE                                                          00001720
C                                                                       00001730
C     UPDATE AND PRINT STATISTICS FOR THIS ITERATION; UPDATE THE        00001740
C     OVERALL STATISTICS FOR THE CONVERGENCE REPORT.                    00001750
C                                                                       00001760
      IFG1WO = IF1CNT + IG1CNT                                          00001770
      IFG2WO = IF2CNT + IG2CNT                                          00001780
      IFG1W  = IF1CNT + IG1CNT + ICCNT                                  00001790
      IFG2W  = IF2CNT + IG2CNT + ICCNT                                  00001800
      WRITE(6,1220)  ITCNT,ICCNT                                        00001810
      WRITE(6,1200)  ITCNT,IFG1WO                                       00001820
      WRITE(6,1210)  ITCNT,IFG1W                                        00001830
      WRITE(6,1230)  ITCNT,IFG2WO                                       00001840
      WRITE(6,1240)  ITCNT,IFG2W                                        00001850
      IT1WO = IT1WO + IFG1WO                                            00001860
      IT1W  = IT1W  + IFG1W                                             00001870
      IT2WO = IT2WO + IFG2WO                                            00001880
      IT2W  = IT2W  + IFG2W                                             00001890
      IF (GDOTG.LE.EPSILN**2)           GO TO 500                       00001900
      IF (DABS(COST-CSTLST).LE.EPSILN)  GO TO 500                       00001910
      CSTLST = COST                                                     00001920
      WRITE(6,860)                                                      00001930
      ITCNT = ITCNT + 1                                                 00001940
C                                                                       00001950
C     STEP 1(B):                                                        00001960
C     COMPUTE THE VECTORS AT WHICH THE COST FUNCTIONAL AND ITS GRADIENT  00001970
C     MUST BE COMPUTED AS FOLLOWS:                                      00001980
C                                                                       00001990
C                                N                                      00002000
C                           X  = X  + SIGMA     FOR J = 1,2,...,NSLV    00002010
C                            J      J                                   00002020
C                                                                       00002030
C                                 N                                     00002040
C                       WHERE:   X  = NTH ITERATE OF SOLVE-FOR VECTOR X 00002050
C                          AND SIGMA  = JTH BASIS VECTOR                00002060
C                                 J                                     00002070
C                                                                       00002080
      WRITE(6,870)  NSLV                                                00002090
      DO  50   JCOL = 1,NSLV                                            00002100
      DO  40   IROW = 1,NSLV                                            00002110
   40 XMAT(IROW,JCOL) = XAP(IROW) + BASIS(IROW,JCOL)                    00002120
      WRITE(6,880)  JCOL                                                00002130
      WRITE(6,730)  (XMAT(IROW,JCOL),IROW = 1,NSLV)                     00002140
   50 CONTINUE                                                          00002150
C                                                                       00002160
C     STEP 1(C):                                                        00002170
C     COMPUTE THE VALUE OF THE COST FUNCTIONAL AND ITS GRADIENT AT EACH 00002180
C     OF THE VECTORS X .                                                00002190
C                     J                                                 00002200
C                                                                       00002210
      WRITE(6,890)  NSLV                                                00002220
      IF1CNT = 0                                                        00002230
      IF2CNT = 0                                                        00002240
      IG1CNT = 0                                                        00002250
      IG2CNT = 0                                                        00002260
      DO  60   JCOL = 1,NSLV                                            00002270
      IF (ICGCMP.EQ.1)                                                  00002280
```

```
     *    CALL CGINTG(XMAT(1,JCOL),XJATXJ(JCOL),XGATXJ(1,JCOL))          00002290
      IF (ICGCMP.GT.1)                                                   00002300
     *    CALL CGLINE(XMAT(1,JCOL),XJATXJ(JCOL),XGATXJ(1,JCOL))          00002310
      IF1CNT = IF1CNT + 1                                                00002320
      IG1CNT = IG1CNT + 1                                                00002330
      WRITE(6,900)  JCOL,XJATXJ(JCOL)                                    00002340
      WRITE(6,920)  JCOL                                                 00002350
      WRITE(6,730)  (XGATXJ(IROW,JCOL),IROW = 1,NSLV)                    00002360
      IF (IPRNT4.EQ.1)                  CALL OBSOUT                      00002370
      IF (IPRNT5.EQ.1)                  CALL PTLOUT                      00002380
   60 CONTINUE                                                           00002390
      IF2CNT = IF2CNT + 1                                                00002400
      IG2CNT = IG2CNT + 1                                                00002410
C                                                                        00002420
C     STEP 1(D):                                                         00002430
C     COMPUTE THE DOT PRODUCT OF EACH GRADIENT VECTOR WITH ITSELF;       00002440
C     COMPARE WITH THE CONVERGENCE CRITERION SQUARED TO SEE IF CONVER-   00002450
C     GENCE WAS ACHIEVED AT ANY OF THE SEARCH VECTORS X .                00002460
C                                                      J                 00002470
C                                                                        00002480
      WRITE(6,930)  NSLV                                                 00002490
      DO  70   JCOL = 1,NSLV                                             00002500
      CALL INPRD1(XGATXJ(1,JCOL),XGATXJ(1,JCOL),GDOTG,BASIS)             00002510
      WRITE(6,940)  JCOL,GDOTG                                           00002520
      IF (GDOTG.GT.EPSILN**2)           GO TO 70                         00002530
      JSAVE = JCOL                                                       00002540
      GO TO 500                                                          00002550
   70 CONTINUE                                                           00002560
C                                                                        00002570
C     STEP 2(A):                                                         00002580
C     COMPUTE THE Y VECTORS AS THE GRADIENT VECTORS COMPUTED AT EACH OF  00002590
C     THE SEARCH VECTORS MINUS THE GRADIENT VECTOR COMPUTED AT THE       00002600
C     CURRENT ITERATE OF THE SOLVE-FOR VECTOR, AS FOLLOWS:               00002610
C                                                                        00002620
C                                       N                                00002630
C                        Y  = G(X ) - G(X )                              00002640
C                         J      J                                       00002650
C                                                                        00002660
      WRITE(6,950)  NSLV                                                 00002670
      DO  90   JCOL = 1,NSLV                                             00002680
      DO  80   IROW = 1,NSLV                                             00002690
   80 YMAT(IROW,JCOL) = XGATXJ(IROW,JCOL) - XGNTH(IROW)                  00002700
      WRITE(6,960)  JCOL                                                 00002710
      WRITE(6,730)  (YMAT(IROW,JCOL),IROW = 1,NSLV)                      00002720
   90 CONTINUE                                                           00002730
C                                                                        00002740
C     STEP 2(B):                                                         00002750
C     COMPUTE THE RESIDUAL VECTORS AS FOLLOWS:                           00002760
C                                                                        00002770
C              (N-1)                                                     00002780
C         R  = V       * Y  - SIGMA                                      00002790
C          1             1        1                                     00002800
C                                                                        00002810
C     AND FOR J = 2,3,...,NSLV                                           00002820
C                                                                        00002830
C              (N-1)                      J-1                           00002840
C         R  = V       * Y  - SIGMA  - SUM/  (<R ,Y >/<R ,Y >) * R       00002850
```

```
C          J             J        J     K=1    K  J    K  K       K       00002860
C                                                                          00002870
C              WHERE:  <*,*> = INNER PRODUCT                               00002880
C                                                                          00002890
C     FIRST, FORM THE FIRST TWO TERMS IN EACH VECTOR.                      00002900
C                                                                          00002910
      DO  120  JCOL = 1,NSLV                                               00002920
      DO  110  IROW = 1,NSLV                                               00002930
      RMAT(IROW,JCOL) = 0.D0                                               00002940
      DO  100  INDX = 1,NSLV                                               00002950
  100 RMAT(IROW,JCOL) = RMAT(IROW,JCOL) + VMTRIC(IROW,INDX)                00002960
     1                                    * YMAT(INDX,JCOL)                00002970
  110 RMAT(IROW,JCOL) = RMAT(IROW,JCOL) - BASIS(IROW,JCOL)                 00002980
  120 CONTINUE                                                             00002990
C                                                                          00003000
C     SECOND, FORM THE NECESSARY FACTORS FOR THE SUMMATIONS FOR            00003010
C     J = 2,3,...,NSLV AND APPLY TO THE PREVIOUS FACTORS.                  00003020
C                                                                          00003030
C                             -1           -1                            00003040
C     NOTE:  TAU  = -<Y ,R >   = -<R ,Y >    FOR <Y ,R >.NE.0              00003050
C               J      J  J         J  J           J  J                   00003060
C                                                                          00003070
C            TAU  = 0.D0                        FOR <Y ,R >.EQ.0           00003080
C               J                                     J  J                 00003090
C                                                                          00003100
      JCOL = 2                                                             00003110
  130 JMIN1 = JCOL - 1                                                     00003120
      DO  140  INDX = JMIN1,NSLV                                           00003130
      CALL INPROD(RMAT(1,JMIN1),YMAT(1,INDX),RDOTY(JMIN1,INDX),NSLV)       00003140
  140 CONTINUE                                                             00003150
      IF (DABS(RDOTY(JMIN1,JMIN1)).LT.1.D-40)                              00003160
     1                        TAU(JMIN1) = 0.D0                           00003170
      IF (DABS(RDOTY(JMIN1,JMIN1)).GE.1.D-40)                              00003180
     1                        TAU(JMIN1) = -1.D0/RDOTY(JMIN1,JMIN1)       00003190
      DO  160  INDX = 1,JMIN1                                              00003200
      DO  150  IROW = 1,NSLV                                               00003210
  150 RMAT(IROW,JCOL) = RMAT(IROW,JCOL) + TAU(INDX) * RDOTY(INDX,JCOL)     00003220
     1                        * RMAT(IROW,INDX)                           00003230
  160 CONTINUE                                                             00003240
      IF (JCOL.GE.NSLV)                 GO TO 170                        00003250
      JCOL = JCOL + 1                                                      00003260
      GO TO 130                                                            00003270
C                                                                          00003280
C     STEP 2(C):                                                           00003290
C     UPDATE THE METRIC MATRIX AS FOLLOWS:                                 00003300
C                                                                          00003310
C                  (N)    (N-1)      NSLV         (J)                     00003320
C                 V    = V      + SUM /   TAU  * B                       00003330
C                                   J=1     J                            00003340
C                                                                          00003350
C                                  (J)          T                        00003360
C                        WHERE:  B    = R  * R                           00003370
C                                        J    J                           00003380
C                                                                          00003390
  170 WRITE(6,970)  NSLV                                                   00003400
      DO  3500 JCOL = 1,NSLV                                               00003410
      WRITE(6,980)  JCOL                                                   00003420
```

```
      WRITE(6,730)  (RMAT(IROW,JCOL),IROW = 1,NSLV)                          00003430
 3500 CONTINUE                                                               00003440
      CALL INPROD(YMAT(1,NSLV),RMAT(1,NSLV),TAUINV,NSLV)                     00003450
      IF (DABS(TAUINV).LT.1.D-40)  TAU(NSLV) = 0.D0                          00003460
      IF (DABS(TAUINV).GE.1.D-40)  TAU(NSLV) = -1.D0/TAUINV                  00003470
      WRITE(6,990)  NSLV                                                     00003480
      DO  200  INDX = 1,NSLV                                                 00003490
      CALL OUTPRD(RMAT(1,INDX),RMAT(1,INDX),BMAT,NSLV)                       00003500
      WRITE(6,1000) INDX,TAU(INDX)                                           00003510
      DO  4000 IROW = 1,NSLV                                                 00003520
      WRITE(6,1010) INDX,IROW                                                00003530
      WRITE(6,730)  (BMAT(IROW,JCOL),JCOL = 1,NSLV)                          00003540
 4000 CONTINUE                                                               00003550
      DO  190  JCOL = 1,NSLV                                                 00003560
      DO  180  IROW = 1,NSLV                                                 00003570
  180 VMTRIC(IROW,JCOL) = VMTRIC(IROW,JCOL) + TAU(INDX)                      00003580
     1                                        * BMAT(IROW,JCOL)             00003590
  190 CONTINUE                                                               00003600
  200 CONTINUE                                                               00003610
      WRITE(6,1020)                                                          00003620
      DO  5000 IROW = 1,NSLV                                                 00003630
      WRITE(6,650)  IROW                                                     00003640
      WRITE(6,730)  (VMTRIC(IROW,JCOL),JCOL = 1,NSLV)                        00003650
 5000 CONTINUE                                                               00003660
C                                                                            00003670
C     STEP 3(A):                                                             00003680
C     DETERMINE THE NEWTON-LIKE DIRECTION OF SEARCH AS FOLLOWS:              00003690
C                                                                            00003700
C                                (N)      N                                  00003710
C                       S  = -V    * G(X )                                   00003720
C                        N                                                   00003730
C                                                                            00003740
      DO  220  IROW = 1,NSLV                                                 00003750
      SRCH(IROW) = 0.D0                                                      00003760
      DO  210  JCOL = 1,NSLV                                                 00003770
  210 SRCH(IROW) = SRCH(IROW) - VMTRIC(IROW,JCOL) * XGNTH(JCOL)              00003780
  220 CONTINUE                                                               00003790
      WRITE(6,1030)                                                          00003800
      WRITE(6,730)  (SRCH(IROW),IROW = 1,NSLV)                               00003810
C                                                                            00003820
C     STEP 3(B):                                                             00003830
C     PERFORM A UNIVARIATE MINIMIZATION ALONG THE NEWTON-LIKE SEARCH         00003840
C     DIRECTION TO DETERMINE THE STEPSIZE ALPHA, AS FOLLOWS:                 00003850
C                                                                            00003860
C                       N                                                    00003870
C          MINIMIZE J(X  + ALPHA * S  ) WITH RESPECT TO ALPHA                00003880
C                                   N                                        00003890
C     THIS MINIMIZATION IS DONE BY SUBROUTINE 'SERCH1', WHICH REQUIRES       00003900
C     AN INITIAL ESTIMATE OF THE SEARCH STEPSIZE. THIS INITIAL ESTIMATE      00003910
C     WILL BE INPUT ON THE "1-D SRCH" CARD IN THE USER INPUT DECK.           00003920
C                                                                            00003930
      ICGSV = ICGCMP                                                         00003940
      IF (ICGCMP.EQ.1)                  ICG = 1                              00003950
      IF (ICGCMP.GT.1)                  ICG = 2                              00003960
      ICGCMP = ICG + 3                                                       00003970
      DO  230  IROW = 1,NSLV                                                 00003980
      XN(IROW) = XAP(IROW)                                                   00003990
```

```
  230 CONTINUE                                                          00004000
      CALL SERCH1(XN,SRCH,NSLV,ALFA1,STPSZ,C1DIM,IPRNT4,ICG,IPRNT5,     00004010
     *                    ALFAN,XMIN,COST)                             00004020
      ICGCMP = ICGSV                                                    00004030
C                                                                       00004040
C     STEP 3(C):                                                        00004050
C     DETERMINE NEW ITERATE FOR THE SOLVE-FOR VECTOR X AND RESTART      00004060
C     THE PROCESS:                                                      00004070
C                                                                       00004080
C                    N+1    N                                          00004090
C                   X    = X  + ALPHA  * S                            00004100
C                                      N    N                         00004110
C                                                                       00004120
      DO  240  IROW = 1,NSLV                                            00004130
      XAP(IROW) = XMIN(IROW)                                            00004140
  240 CONTINUE                                                          00004150
      GO TO 30                                                          00004160
C                                                                       00004170
C     OUTPUT CONVERGENCE REPORT FOR THE PARALLEL VARIABLE METRIC METHOD.00004180
C                                                                       00004190
  500 WRITE(6,1040)                                                     00004200
      IF (JSAVE.EQ.100)                 GO TO 520                       00004210
      DO  510  IROW = 1,NSLV                                            00004220
      XMIN(IROW)  = XMAT(IROW,JSAVE)                                    00004230
      XGNTH(IROW) = XGATXJ(IROW,JSAVE)                                  00004240
  510 CONTINUE                                                          00004250
      COST = XJATXJ(JSAVE)                                              00004260
      GO TO 530                                                         00004270
  520 DO  530  IROW = 1,NSLV                                            00004280
      XMIN(IROW) = XAP(IROW)                                            00004290
  530 CONTINUE                                                          00004300
      WRITE(6,1050)                                                     00004310
      IF (ISLVFR.EQ.1)                  WRITE(6,1100)                   00004320
      IF (ISLVFR.EQ.2)                  WRITE(6,1110)                   00004330
      IF (ISLVFR.EQ.3)                  WRITE(6,1120)                   00004340
      WRITE(6,1060)                                                     00004350
      DO  540  IROW = 1,NSLV                                            00004360
      WRITE(6,1070) IROW,XMIN(IROW),XGNTH(IROW),SRCH(IROW)              00004370
  540 CONTINUE                                                          00004380
      WRITE(6,1080) COST                                                00004390
      WRITE(6,1130)                                                     00004400
      WRITE(6,1320)                                                     00004410
      DO  550  IROW = 1,6                                               00004420
      WRITE(6,1340) (CMTXAP(IROW,JCOL),JCOL=1,6)                        00004430
  550 CONTINUE                                                          00004440
      IF (ICOV.NE.1)                    GO TO 560                       00004450
      CALL COVMTX(XMIN,CMTXAP,RSSPOS,RSSVEL,RMSPOS,RMSVEL)              00004460
      WRITE(6,1330)                                                     00004470
      DO  560  IROW = 1,6                                               00004480
      WRITE(6,1340) (CMTXAP(IROW,JCOL),JCOL=1,6)                        00004490
  560 CONTINUE                                                          00004500
      IF (ISLVFR.NE.1)                  GO TO 570                       00004510
      WRITE(6,1350) RSSPOS                                              00004520
      WRITE(6,1360) RSSVEL                                              00004530
      WRITE(6,1370) RMSPOS                                              00004540
      WRITE(6,1380) RMSVEL                                              00004550
  570 CONTINUE                                                          00004560
```

```
      WRITE(6,1090) ITCNT                                                   00004570
      WRITE(6,1250) ITCNT,IT1WO                                             00004580
      WRITE(6,1260) ITCNT,IT1W                                              00004590
      WRITE(6,1270) ITCNT,IT2WO                                             00004600
      WRITE(6,1280) ITCNT,IT2W                                              00004610
      SURWO = DFLOAT(IT1WO)/DFLOAT(IT2WO)                                   00004620
      SURW  = DFLOAT(IT1W)/DFLOAT(IT2W)                                     00004630
      WRITE(6,1290) SURWO,SURW                                              00004640
      WRITE(6,1300)                                                         00004650
      RETURN                                                                00004660
C                                                                           00004670
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * * * 00004680
C                                                                           00004690
  700 FORMAT('1',131('*')/1X,32('*'),67X,32('*')/1X,32('*'),'    PAR',      00004700
     1       'ALLEL VARIABLE METRIC (PVM) FUNCTION MINIMIZATION REPOR',     00004710
     2       'T    ',32('*')/1X,32('*'),67X,32('*')/1X,131('*')/////)       00004720
C                                                                           00004730
  710 FORMAT(32X,69('*')/32X,'*',67X,'*'/32X,'*',19X,'BASIS VECTORS ',      00004740
     1       'FOR THIS METHOD',19X,'*'/32X,'*',67X,'*'/32X,69('*')//)       00004750
C                                                                           00004760
  720 FORMAT(2X,'SIGMA(',I2,'):')                                           00004770
C                                                                           00004780
  730 FORMAT(12X,6(1PD20.10))                                               00004790
C                                                                           00004800
  740 FORMAT(12X,1PD20.10)                                                  00004810
C                                                                           00004820
  750 FORMAT(/////33X,'NUMBER OF SOLVE-FOR PARAMETERS/DIMENSION OF ',       00004830
     1       'PARAMETER SPACE:  ',I2,'.')                                   00004840
C                                                                           00004850
  760 FORMAT(///24X,'DEGREE OF PARALLELISM IN THE ASSUMED PARALLEL ',       00004860
     1       'PROCESSOR COMPUTER ARCHITIECTURE:  ',I2,'.')                  00004870
C                                                                           00004880
  770 FORMAT(///37X,'CONVERGENCE CRITERION FOR THE METHOD:',1PD20.10,       00004890
     1       '.')                                                           00004900
C                                                                           00004910
  780 FORMAT(///28X,'SCALE FACTOR FCR COMPUTATION OF THE SCHAUDER ',        00004920
     1       'BASIS SPANNING PARAMETER SPACE:')                             00004930
C                                                                           00004940
  790 FORMAT('1',9X,112('*')/10X,23('*'),23X,'ITERATION NUMBER ',I4,        00004950
     1       23X,22('*')/10X,112('*')///)                                   00004960
C                                                                           00004970
  800 FORMAT(5X,'CURRENT SOLVE-FOR VECTOR:'/10X,'X:')                       00004980
C                                                                           00004990
  810 FORMAT(///5X,'COST FUNCTIONAL EVALUATED AT THE CURRENT ITERATE',      00005000
     1       ' OF THE SOLVE-FOR VECTOR:'/7X,'J(X):',1PD20.10///)            00005010
C                                                                           00005020
  820 FORMAT(5X,'GRADIENT OF THE COST FUNCTIONAL EVALUATED AT THE CUR',     00005030
     1       'RENT ITERATE OF THE SOLVE-FOR VECTOR:'/7X,'G(X):')            00005040
C                                                                           00005050
  830 FORMAT(///5X,'SQUARE OF THE MAGNITUDE OF THE CURRENT GRADIENT ',      00005060
     1       'VECTOR:'/4X,'G DOT G:',1PD20.10///)                           00005070
C                                                                           00005080
  840 FORMAT(5X,'CURRENT ITERATE OF THE METRIC MATRIX V(N):')               00005090
C                                                                           00005100
  850 FORMAT(4X,'ROW(',I2,'):')                                             00005110
C                                                                           00005120
  860 FORMAT(///32X,69('*')/32X,'*',67X,'*'/32X,'*',12X,'CONVERGENCE N',   00005130
```

```
     1       'OT ACHIEVED -- CONTINUE SEARCH',12X,'*'/32X,'*',67X,'*'/  00005140
     2       32X,69('*')//)                                             00005150
C                                                                       00005160
  870 FORMAT(5X,'THE ',I2,' VECTORS AT WHICH THE COST AND GRADIENT ',    00005170
     1       'MUST BE COMPUTED (IN PARALLEL):')                         00005180
C                                                                       00005190
  880 FORMAT(6X,'X(',I2,'):')                                           00005200
C                                                                       00005210
  890 FORMAT(//5X,'THE ',I2,' EVALUATIONS OF THE COST FUNCTIONAL AND ', 00005220
     1       'THE GRADIENT VECTOR:')                                    00005230
C                                                                       00005240
  900 FORMAT(/5X,'J(X',I2,'):',1PD20.10)                                00005250
C                                                                       00005260
  920 FORMAT(5X,'G(X',I2,'):')                                          00005270
C                                                                       00005280
  930 FORMAT(//5X,'THE DOT PRODUCTS OF EACH OF THE ',I2,' GRADIENT ',   00005290
     1       'VECTORS WITH ITSELF:')                                    00005300
C                                                                       00005310
  940 FORMAT(2X,'GDOTG(',I2,'):',1PD20.10)                              00005320
C                                                                       00005330
  950 FORMAT(//5X,'THE ',I2,' Y VECTORS:')                              00005340
C                                                                       00005350
  960 FORMAT(6X,'Y(',I2,'):')                                           00005360
C                                                                       00005370
  970 FORMAT(//5X,'THE ',I2,' RESIDUAL VECTORS:')                       00005380
C                                                                       00005390
  980 FORMAT(6X,'R(',I2,'):')                                           00005400
C                                                                       00005410
  990 FORMAT(//5X,'THE ',I2,' VALUES OF THE SCALARS TAU(J) AND MATRI',  00005420
     1       'CES B(J):')                                               00005430
C                                                                       00005440
 1000 FORMAT(/4X,'TAU(',I2,'):',1PD20.10)                               00005450
C                                                                       00005460
 1010 FORMAT(/6X,'B(',I2,'):  ROW(',I2,')')                             00005470
C                                                                       00005480
 1020 FORMAT(//5X,'UPDATED ITERATE OF THE METRIC MATRIX V(N+1):')       00005490
C                                                                       00005500
 1030 FORMAT(//5X,'NEWTON-LIKE SEARCH DIRECTION VECTOR:'/9X,'SN:')      00005510
C                                                                       00005520
 1040 FORMAT(//32X,69('*')/32X,'*',67X,'*'/32X,'*',9X,'CONVERGENCE AC', 00005530
     1       'HIEVED -- OUTPUT CONVERGENCE REPORT',9X,'*'/32X,'*',67X,  00005540
     2       '*'/32X,69('*'))                                           00005550
C                                                                       00005560
 1050 FORMAT('1',131('*')/1X,32('*'),67X,32('*')/1X,32('*'),9X,'PARA',  00005570
     1       'LLEL VARIABLE METRIC (PVM) CONVERGENCE REPORT',9X,         00005580
     2       32('*')/1X,32('*'),67X,32('*')/1X,131('*')/////)           00005590
C                                                                       00005600
 1060 FORMAT(50X,'OPTIMUM',19X,'OPTIMUM',18X,'OPTIMUM'/25X,'COMPONENT', 00005610
     1       12X,'SOLVE-FOR VECTOR',11X,'GRADIENT VECTOR',11X,'SEARCH', 00005620
     2       ' VECTOR'/19X,4(20('_'),5X)/)                              00005630
C                                                                       00005640
 1070 FORMAT(28X,I2,12X,3(1PD20.10,5X))                                 00005650
C                                                                       00005660
 1080 FORMAT(///22X,'THE MINIMUM VALUE OF THE ASSOCIATED COST FUNC',    00005670
     1       'TIONAL AT CONVERGENCE:',1PD20.10)                         00005680
C                                                                       00005690
 1090 FORMAT('1'///45X,'THE NUMBER OF ITERATIONS TO CONVERGENCE: ',I4)  00005700
```

```
C                                                                       00005710
 1100 FORMAT(///27X,'THE SOLVE-FOR VECTOR COMPONENTS ARE THE EPOCH ',   00005720
     1       'POSITION AND VELOCITY COMPONENTS'//53X,'COMPONENT 1:  ',  00005730
     2       'RX0 (KM)'/53X,'COMPONENT 2:  RY0 (KM)'/53X,'COMPONENT',   00005740
     3       ' 3:  RZ0 (KM)'/53X,'COMPONENT 4:  VX0 (KM/SEC)'/53X,      00005750
     4       'COMPONENT 5:  VY0 (KM/SEC)'/53X,'COMPONENT 6:  VZ0 ',     00005760
     5       '(KM/SEC)'//////)                                          00005770
C                                                                       00005780
 1110 FORMAT(///34X,'THE SOLVE-FOR VECTOR COMPONENTS ARE THE EPOCH ',   00005790
     1       'KEPLERIAN ELEMENTS'//45X,'COMPONENT 1:  A (KM)'/45X,      00005800
     2       'COMPONENT 2:  E'/45X,'COMPONENT 3:  I (RADIANS)'/45X,     00005810
     3       'COMPONENT 4:  ARG. OF PERIAPSIS (RADIANS)'/45X,           00005820
     4       'COMPONENT 5:  LONG. OF ASC. NODE (RADIANS)'/45X,          00005830
     5       'COMPONENT 6:  MEAN ANOMALY (RADIANS)'//////)              00005840
C                                                                       00005850
 1120 FORMAT(///33X,'THE SOLVE-FOR VECTOR COMPONENTS ARE THE EPOCH ',   00005860
     1       'EQUINOCTIAL ELEMENTS'//47X,'COMPONENT 1:  A (KM)'/47X,    00005870
     2       'COMPONENT 2:  H'/47X,'COMPONENT 3:  K'/47X,'COMPONENT ',  00005880
     3       '4:  P'/47X,'COMPONENT 5:  Q'/47X,'COMPONENT 6:  MEAN ',   00005890
     4       'LONGITUDE (RADIANS)'//////)                               00005900
C                                                                       00005910
 1130 FORMAT('1    ******',16X,'ORBIT ESTIMATION STATISTICAL SUMMARY:', 00005920
     1       '  A PRIORI STATISTICS FOR INFORMATION ONLY',16X,          00005930
     2       '**********'///)                                          00005940
C                                                                       00005950
 1200 FORMAT(///29X,'NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (EX',  00005960
     1       'CLUDING UNIVARIATE SEARCH)'/29X,'IN SEQUENTIAL EXECUTI',  00005970
     2       'ON OF PVM METHOD FOR ITERATION ',I2,': ',I3,'.')          00005980
C                                                                       00005990
 1210 FORMAT(///29X,'NUMBER OF FUNCTION AND GRADIENT EVALUATIONS (IN',  00006000
     1       'CLUDING UNIVARIATE SEARCH)'/29X,'IN SEQUENTIAL EXECUTI',  00006010
     2       'ON OF PVM METHOD FOR ITERATION ',I2,': ',I3,'.')          00006020
C                                                                       00006030
 1220 FORMAT(///18X,'NUMBER OF FUNCTION EVALUATIONS (SEQUENTIAL) IN ',  00006040
     1       'THE UNIVARIATE SEARCH METHOD FOR ITERATION ',I2,': ',     00006050
     2       I3,'.')                                                    00006060
C                                                                       00006070
 1230 FORMAT(///28X,'EFFECTIVE NUMBER OF FUNCTION AND GRADIENT EVALU',  00006080
     1       'ATIONS (EXCLUDING UNIVARIATE'/28X,'SEARCH) IN PARALLEL',  00006090
     2       ' EXECUTION OF PVM METHOD FOR ITERATION ',I2,': ',I3,'.')  00006100
C                                                                       00006110
 1240 FORMAT(///28X,'EFFECTIVE NUMBER OF FUNCTION AND GRADIENT EVALU',  00006120
     1       'ATIONS (INCLUDING UNIVARIATE'/28X,'SEARCH) IN PARALLEL',  00006130
     2       ' EXECUTION OF PVM METHOD FOR ITERATION ',I2,': ',I3,'.')  00006140
C                                                                       00006150
 1250 FORMAT(///33X,'TOTAL NUMBER OF FUNCTION AND GRADIENT EVALUATI',   00006160
     1       'ONS (EXCLUDING UNI-'/33X,'VARIATE SEARCH) IN ',I2,' SE',  00006170
     2       'QUENTIAL ITERATIONS OF THE PVM METHOD:  ',I3)             00006180
C                                                                       00006190
 1260 FORMAT(///33X,'TOTAL NUMBER OF FUNCTION AND GRADIENT EVALUATI',   00006200
     1       'ONS (INCLUDING UNI-'/33X,'VARIATE SEARCH) IN ',I2,' SE',  00006210
     2       'QUENTIAL ITERATIONS OF THE PVM METHOD:  ',I3)             00006220
C                                                                       00006230
 1270 FORMAT(///31X,'EFFECTIVE TOTAL NUMBER OF FUNCTION AND GRADIENT ', 00006240
     1       'EVALUATIONS (EXCLUDING'/31X,'UNIVARIATE SEARCH) IN ',I2,  00006250
     2       ' PARALLEL ITERATIONS OF THE PVM METHOD:  ',I3)            00006260
C                                                                       00006270
```

```
ESTEX                JSS1493.PRLLEL.ORBDET.FORT

 1280 FORMAT(///31X,'EFFECTIVE TOTAL NUMBER OF FUNCTION AND GRADIENT ', 00006280
     1       'EVALUATIONS (INCLUDING'/31X,'UNIVARIATE SEARCH) IN ',I2,  00006290
     2       ' PARALLEL ITERATIONS OF THE PVM METHOD:  ',I3)            00006300
C                                                                       00006310
 1290 FORMAT(///28X,'THE SPEED-UP RATIO (SR) IS DEFINED AS THE NUMBER', 00006320
     1       ' OF EVALUATIONS IN THE SEQUENTIAL'/28X,'MODE DIVIDED ',   00006330
     2       'BY THE NUMBER OF EVALUATIONS IN THE PARALLEL MODE.'//37X, 00006340
     3       'EXCLUDING UNIVARIATE SEARCH:  SR = ',1PD17.10,' TO 1.'//  00006350
     4       37X,'INCLUDING UNIVARIATE SEARCH:  SR = ',1PD17.10,        00006360
     5       ' TO 1.')                                                  00006370
C                                                                       00006380
 1300 FORMAT(///1X,131('*')/1X,32('*'),67X,32('*')/1X,32('*'),'   E',   00006390
     1       'ND OF PARALLEL VARIABLE METRIC FUNCTION MINIMIZATION ',   00006400
     2       'REPORT    ',32('*')/1X,32('*'),67X,32('*')/1X,131('*'))   00006410
C                                                                       00006420
 1310 FORMAT(48X,'SCALE FACTOR ',I1,' =',1PD20.10)                      00006430
C                                                                       00006440
 1320 FORMAT(///5X,30('*'),10X,'A PRIORI COVARIANCE MATRIX OF ESTIMA',  00006450
     1       'TION ERRORS',10X,30('*'))                                 00006460
C                                                                       00006470
 1330 FORMAT(///5X,30('*'),11X,'UPDATED COVARIANCE MATRIX OF ESTIMA',   00006480
     1       'TION ERRORS',10X,30('*'))                                 00006490
C                                                                       00006500
 1340 FORMAT(/6(2X,1PD20.10))                                           00006510
C                                                                       00006520
 1350 FORMAT(///34X,'MEAN SQUARED POSITION ERROR    :',1PD20.10,        00006530
     *       ' KM**2')                                                  00006540
C                                                                       00006550
 1360 FORMAT(///34X,'MEAN SQUARED VELOCITY ERROR    :',1PD20.10,        00006560
     *       ' KM**2/SEC**2')                                           00006570
C                                                                       00006580
 1370 FORMAT(///34X,'ROOT MEAN SQUARE POSITION ERROR:',1PD20.10,' KM')  00006590
C                                                                       00006600
 1380 FORMAT(///34X,'ROOT MEAN SQUARE VELOCITY ERROR:',1PD20.10,        00006610
     *       ' KM/SEC')                                                 00006620
C                                                                       00006630
      END                                                               00006640

******** END OF MEMBER ESTEX      664 RECORDS ********
```

```
FORCE                JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE FORCE(ST,GM,ACCEL)                                     00000010
C                                                                       00000020
C     VERSION OF 08/09/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE ACCELERATION ACTING ON A SPACECRAFT, GIVEN THE 00000060
C         SPACECRAFT'S POSITION AND VELOCITY VECTORS.                   00000070
C                                                                       00000080
C     METHOD                                                            00000090
C         GIVEN THE SPACECRAFT POSITION AND VELOCITY VECTORS IN INERTIAL00000100
C         CARTESIAN COORDINATES (VIA THE ARRAY 'ST') AND THE GRAVITA-   00000110
C         TIONAL PARAMETER OF THE CENTRAL BODY, THE TWO-BODY GRAVITA-   00000120
C         TIONAL ACCELERATION CONTRIBUTION IS COMPUTED FROM:            00000130
C                                                                       00000140
C                                                                       00000150
C                    2                                                  00000160
C                   D  R                GM                            00000170
C                   ------  =  -  ------------  *  R                   00000180
C                     2                3                              00000190
C                   DT                R                               00000200
C                                                                       00000210
C         IF THE FORCE MODEL IS TO INCLUDE THE ZONAL HARMONICS IN THE   00000220
C         SPHERICAL HARMONIC REPRESENTATION OF THE GRAVITATIONAL PO-     00000230
C         TENTIAL, THEIR CONTRIBUTION TO THE SPACECRAFT ACCELERATION    00000240
C         IS GIVEN BY:                                                  00000250
C                                                                       00000260
C                                                                       00000270
C                        GM       INF                  K     '         00000280
C              A    =  ------  * SUM/  J  * (R   / R) * (P    (COS ∂) I  -00000290
C               D          2     K=2   K     EQ          K+1         R  00000300
C                         R                                             00000310
C                                                                       00000320
C                                                                       00000330
C                                                 '                     00000340
C                                          P (COS ∂) I   )              00000350
C                                           K         Z                 00000360
C                                                                       00000370
C                   WHERE GM = GRAVITATIONAL PARAMETER OF CENTRAL BODY  00000380
C                          R = MAGNITUDE OF SPACECRAFT POSITION VECTOR  00000390
C                         J  = KTH ZONAL HARMONIC COEFFICIENT           00000400
C                          K                                            00000410
C                        R   = EQUATORIAL RADIUS OF THE CENTRAL BODY    00000420
C                         EQ                                            00000430
C                       '                                               00000440
C                     P (X) = DERIVATIVE OF THE NTH LEGENDRE POLYNOMIAL 00000450
C                      N                                                00000460
C                     COS ∂ = COSINE OF THE COLATITUDE OF THE SATELLITE 00000470
C                             POSITION VECTOR                           00000480
C                                                                       00000490
C                        I  = UNIT VECTOR IN THE DIRECTION OF THE       00000500
C                         R   SATELLITE POSITION VECTOR                 00000510
C                                                                       00000520
C                        I  = UNIT VECTOR IN THE INERTIAL CARTESIAN Z   00000530
C                         Z   DIRECTION                                 00000540
C         IF DRAG IS TO BE INCLUDED IN THE FORCE MODEL, THE DRAG CONTRI-00000550
C         BUTION TO THE SPACECRAFT ACCELERATION IS COMPUTED BY THE SUB- 00000560
C         ROUTINE 'AERO', AND ADDED TO THE PREVIOUSLY COMPUTED VEHICLE  00000570
```

```
C         ACCELERATION, AS FOLLOWS:                                          00000580
C                                                                            00000590
C                                                                            00000600
C                    A  = -(1/2) * C  * (A/M) * RHO * V  * V                 00000610
C                     D              D                   A    A               00000620
C                                                                            00000630
C               WHERE  C  = DRAG COEFFICIENT                                 00000640
C                       D                                                    00000650
C                     A/M = SATELLITE AREA-TO-MASS RATIO                     00000660
C                     RHO = ATMOSPHERIC DENSITY AT R                         00000670
C                                                                            00000680
C                      V  = SPACECRAFT VELOCITY VECTOR RELATIVE TO THE       00000690
C                       A         ROTATING ATMOSPHERE                        00000700
C                                                                            00000710
C         THE DENSITY IS COMPUTED USING A SIMPLE EXPONENTIAL DENSITY         00000720
C         MODEL:                                                             00000730
C                                                                            00000740
C                                                                            00000750
C                    RHO = RHO  * EXP ( -(R - R )/LAMDA )                    00000760
C                              0               0                             00000770
C                                                                            00000780
C               WHERE RHO  = ATMOSPHERIC DENSITY AT THE REFERENCE R          00000790
C                         0                                          0       00000800
C                       R  = REFERENCE RADIUS (TAKEN TO BE THE MEAN          00000810
C                        0       RADIUS OF THE CENTRAL BODY                  00000820
C                     LAMDA = ATMOSPHERIC SCALE HEIGHT                       00000830
C                                                                            00000840
C     CALLING SEQUENCE                                                       00000850
C         CALL FORCE(ST,GM,ACCEL)                                            00000860
C                                                                            00000870
C     CALLING SEQUENCE PARAMETERS                                            00000880
C         ST    I       ARRAY (6X1) CONTAINING THE SPACECRAFT POSITION       00000890
C                       AND VELOCITY VECTOR COMPONENTS                       00000900
C         GM    I       GRAVITATIONAL PARAMETER OF THE CENTRAL BODY          00000910
C         ACCEL O       ARRAY (6X1) IN WHICH THE FIRST THREE COMPONENTS      00000920
C                       CONTAIN THE SPACECRAFT ACCELERATION VECTOR            00000930
C                                                                            00000940
C     EXTERNAL REFERENCES                                                    00000950
C         AERO, LEGPRM                                                       00000960
C                                                                            00000970
C     ANALYSIS                                                               00000980
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00000990
C                                                                            00001000
C     PROGRAMMING                                                            00001010
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00001020
C                                                                            00001030
      IMPLICIT REAL*8(A-H,O-Z)                                               00001040
C                                                                            00001050
C * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * * * 00001060
C                                                                            00001070
      DIMENSION       ST(6)    ,   ACCEL(6)  ,   DRGACC(3)                   00001080
C                                                                            00001090
C * * * * * * * * *           COMMON BLOCKS          * * * * * * * * * * * * 00001100
C                                                                            00001110
      COMMON/DRGPAR/  CSUBD    ,   SCAREA    ,   SCMASS    ,   RHOZRO    ,00001120
     *                RZERO    ,   SCLHGT    ,   IDRAG                     00001130
C                                                                            00001140
```

```
FORCE                JSS1493.PRLLEL.ORBDET.FORT

      COMMON/ERTSHP/  REQ      ,   WE       ,   FINV     ,   SPHINF    00001150
C                                                                       00001160
      COMMON/ZNLCOF/  SHJ(8)   ,   IZNLS                               00001170
C                                                                       00001180
C     COMPUTE THE TWO-BODY ACCELERATION                                 00001190
C                                                                       00001200
      RSQ = ST(1)*ST(1) + ST(2)*ST(2) + ST(3)*ST(3)                     00001210
      R   = DSQRT(RSQ)                                                  00001220
      FAC1 = GM/RSQ                                                     00001230
      DO  10   I = 1,3                                                  00001240
      ACCEL(I) = -FAC1 * (ST(I)/R)                                      00001250
   10 CONTINUE                                                          00001260
C                                                                       00001270
C     COMPUTE CONTRIBUTION DUE TO ZONAL HARMONICS, IF REQUIRED          00001280
C                                                                       00001290
      IF (IZNLS.LE.1)                  GO TO 50                         00001300
      CSPHI = ST(3)/R                                                   00001310
      KZONAL = 2                                                        00001320
   20 KPOLY  = KZONAL + 1                                               00001330
      CALL LEGPRM(CSPHI,KPOLY,PPRMK1,PPRMK,PPKMN1)                      00001340
      FAC1R  = SHJ(KZONAL) * ((REQ/R)**KZONAL) * PPRMK1                 00001350
      FAC1Z  = SHJ(KZONAL) * ((REQ/R)**KZONAL) * PPRMK                  00001360
      DO  30   I = 1,3                                                  00001370
      ACCEL(I) = ACCEL(I) + FAC1 * FAC1R * (ST(I)/R)                    00001380
   30 CONTINUE                                                          00001390
      ACCEL(3) = ACCEL(3) - FAC1 * FAC1Z                                00001400
      IF (KZONAL.GE.IZNLS)             GO TO 50                         00001410
      KZONAL = KZONAL + 1                                               00001420
      GO TO 20                                                          00001430
C                                                                       00001440
C     COMPUTE THE CONTRIBUTION DUE TO ATMOSPHERIC DRAG, ASSUMING A RIGID00001450
C     BODY ROTATING ATMOSPHERE AND EXPONENTIAL DENSITY MODEL            00001460
C                                                                       00001470
   50 IF (IDRAG.NE.1)                  RETURN                           00001480
      CALL AERO(ST,DRGACC)                                              00001490
      DO  60   I = 1,3                                                  00001500
      ACCEL(I) = ACCEL(I) + DRGACC(I)                                   00001510
   60 CONTINUE                                                          00001520
      RETURN                                                            00001530
      END                                                               00001540

******** END OF MEMBER FORCE      154 RECORDS ********
```

F2BDY JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE F2BDY(ST,GM,ACCEL)                                     00000010
C                                                                       00000020
C     VERSION OF 04/04/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO COMPUTE THE TWO BODY ACCELERATION ACTING ON THE SATELLITE   00000060
C        FOR USE IN THE PICARD-CHEBYSHEV INTEGRATION PROCESS.           00000070
C                                                                       00000080
C     CALLING SEQUENCE                                                  00000090
C        CALL F2BDY(ST,GM,ACCEL)                                        00000100
C                                                                       00000110
C     CALLING SEQUENCE PARAMETERS                                       00000120
C        ST      I    ARRAY (6X1) CONTAINING THE CURRENT SATELLITE     00000130
C                     POSITION AND VELOCITY VECTORS                     00000140
C        GM      I    CENTRAL BODY GRAVITATIONAL CONSTANT              00000150
C        ACCEL   O    TWO BODY ACCELERATION ACTING ON SATELLITE        00000160
C                                                                       00000170
C     EXTERNAL REFERENCES                                               00000180
C        NONE                                                           00000190
C                                                                       00000200
C     ANALYSIS                                                          00000210
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000220
C                                                                       00000230
C     PROGRAMMING                                                       00000240
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000250
C                                                                       00000260
      IMPLICIT REAL*8(A-H,O-Z)                                          00000270
      DIMENSION     ST(6),ACCEL(6)                                      00000280
      R = DSQRT(ST(1)**2 + ST(2)**2 + ST(3)**2)                         00000290
      DO  10   I = 1,3                                                  00000300
      ACCEL(I) = -(GM/R**3) * ST(I)                                     00000310
   10 CONTINUE                                                          00000320
      RETURN                                                            00000330
      END                                                               00000340
```

******** END OF MEMBER F2BDY 34 RECORDS ********

```
ICNTRL              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK ICNTRL  00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/ICNTRL/  IRNTYP     ,   IVINIT     ,   IVINTG     ,        00000120
     *                ISLVFR     ,   NSLVFR     ,   IPTLS      ,        00000130
     *                IJ2RTS     ,   IRVPAR     ,   IPARRV     ,        00000140
     *                IPRNT1     ,   IPRNT2     ,   IPRNT3     ,        00000150
     *                IPRNT4     ,   IPRNT5     ,   IOUT       ,        00000160
     *                ICGCMP     ,   ISCID      ,   NPAR       ,        00000170
     *                ICOV                                              00000180
C                                                                       00000190
C                          #LOCA-                                       00000200
C             DIMEN-        TION                                        00000210
C  VARIABLE    SION      INTEGER              DESCRIPTION               00000220
C  ________   _____     _______   ____________________________________  00000230
C                                                                       00000240
C    IRNTYP       1           1   RUN-TYPE INDICATOR:                   00000250
C                                 1=EPHEMERIS GENERATION RUN            00000260
C                                 2=ORBIT ESTIMATION RUN                00000270
C    IVINIT       1           2   INPUT COORDINATE SYSTEM TYPE:         00000280
C                                 1=CARTESIAN                           00000290
C                                 2=KEPLERIAN                           00000300
C                                 3=EQUINOCTIAL                         00000310
C    IVINTG       1           3   EQUATION SET BEING INTEGRATED:        00000320
C                                 1=COWELL (CARTESIAN)                  00000330
C                                 2=VOP (KEPLERIAN)                     00000340
C                                 3=VOP (EQUINOCTIAL)                   00000350
C    ISLVFR       1           4   TYPE OF ESTIMATION SOLVE-FOR PARAMETERS00000360
C                                 1=EPOCH CARTESIAN                     00000370
C                                 2=EPOCH KEPLERIAN                     00000380
C                                 3=EPOCH EQUINOCTIAL                   00000390
C    NSLVFR       1           5   NUMBER OF SOLVE-FOR PARAMETERS        00000400
C     IPTLS       1           6   FLAG INDICATING THAT ANALYTICAL PARTIAL00000410
C                                 DERIVATIVES ARE TO BE COMPUTED:       00000420
C                                 0=DO NOT COMPUTE PARTIALS             00000430
C                                 1=COMPUTE ANALYTICAL PARTIALS         00000440
C    IJ2RTS       1           7   FLAG INDICATING THAT J2 SECULAR RATES 00000450
C                                 ARE TO BE INCLUDED IN THE PARTIAL DE- 00000460
C                                 RIVATIVE COMPUTATION                  00000470
C    IRVPAR       1           8   FLAG TO INDICATE WHICH POSITION AND   00000480
C                                 VELOCITY PARTIALS WRT EQUINOCTIAL     00000490
C                                 ELEMENTS ARE TO BE COMPUTED:          00000500
C                                 0=COMPUTE BOTH POSITION AND VELOCITY  00000510
C                                   PARTIALS                            00000520
C                                 1=COMPUTE ONLY POSITION PARTIALS      00000530
C                                 2=COMPUTE ONLY VELOCITY PARTIALS      00000540
C    IPARRV       1           9   FLAG TO INDICATE WHICH PARTIALS OF THE 00000550
C                                 EQUINOCTIAL ELEMENTS WRT POSITION AND 00000560
C                                 VELOCITY ARE TO BE TAKEN:             00000570
```

```
C                                    0=COMPUTE PARTIALS WITH RESPECT TO BOTH00000580
C                                      POSITION AND VELOCITY               00000590
C                                    1=COMPUTE PARTIALS WITH RESPECT TO    00000600
C                                      POSITION ONLY                       00000610
C                                    2=COMPUTE PARTIALS WITH RESPECT TO    00000620
C                                      VELOCITY ONLY                       00000630
C     IPRNT1        1        10      CONTROL FLAG FOR INTEGRATION PRINTOUT 00000640
C                                    AT EACH OUTPUT TIME:                  00000650
C                                    0=DO NOT PRINT INTEGRATION REPORT     00000660
C                                    1=PRINT ONLY POSITION, VELOCITY AND   00000670
C                                      ORBITAL ELEMENTS AT EACH OUTPUT TIME 00000680
C                                    2=PRINT POSITION, VELOCITY, ORBITAL   00000690
C                                      ELEMENTS AND 20 AUXILIARY QUANTITIES 00000700
C                                      AT EACH OUTPUT TIME                 00000710
C     IPRNT2        1        11      CONTROL FLAG FOR PARTIAL DERIVATIVE   00000720
C                                    PRINTOUT AT EACH OUTPUT TIME:         00000730
C                                    0=DO NOT PRINT ANALYTICAL PARTIALS    00000740
C                                    1=PRINT ANALYTICAL PARTIAL DERIVATIVES 00000750
C                                      OF CURRENT STATE WITH RESPECT TO THE 00000760
C                                      SOLVE-FOR PARAMETERS AT EACH OUTPUT  00000770
C                                      TIME                                00000780
C     IPRNT3        1        12      CONTROL FLAG FOR THE PICARD-CHEBYSHEV 00000790
C                                    INTEGRATION CONVERGENCE REPORT PRINTOUT00000800
C                                    0=DO NOT PRINT PICARD-CHEBYSHEV REPORT 00000810
C                                    1=PRINT PICARD-CHEBYSHEV REPORT AT THE 00000820
C                                      BEGINNING OF EACH INTEGRATION STEP IN00000830
C                                      THE INTEGRATION ARC                 00000840
C     IPRNT4        1        13      CONTROL FLAG FOR OBSERVATION PROCESSING00000850
C                                    REPORT PRINTOUT:                      00000860
C                                    0=DO NOT PRINT OBSERVATION PROCESSING 00000870
C                                      REPORT AT EACH EVALUATION OF THE COST00000880
C                                      AND GRADIENT                        00000890
C                                    1=PRINT OBSERVATION PROCESSING REPORT 00000900
C                                      AT EACH EVALUATION OF THE COST AND  00000910
C                                      GRADIENT                            00000920
C     IPRNT5        1        14      CONTROL FLAG FOR OBSERVATION PARTIAL  00000930
C                                    DERIVATIVE REPORT PRINTOUT:           00000940
C                                    0=DO NOT PRINT OBSERVATION PARTIALS    00000950
C                                      AT EACH EVALUATION OF THE COST AND  00000960
C                                      GRADIENT                            00000970
C                                    1=PRINT OBSERVATION PARTIALS AT EACH  00000980
C                                      EVALUATION OF THE COST AND GRADIENT 00000990
C       IOUT        1        15      OUTPUT ELEMENT SET FOR EPHEMERIS GEN- 00001000
C                                    ERATION REPORT:                       00001010
C                                    1=KEPLERIAN                           00001020
C                                    2=EQUINOCTIAL                         00001030
C     ICGCMP        1        16      METHOD FOR EVALUATION OF THE COST     00001040
C                                    FUNCTIONAL AND ITS GRADIENT:          00001050
C                                    1=DIRECT NUMERICAL INTEGRATION METHOD 00001060
C                                    2=LINEARIZATION METHOD WITHOUT RE-     00001070
C                                      COMPUTATION OF THE OBSERVATION      00001080
C                                      PARTIAL DERIVATIVES                 00001090
C                                    3=LINEARIZATION METHOD WITH RECOMPU-   00001100
C                                      TATION OF THE OBSERVATION PARTIAL    00001110
C                                      DERIVATIVES                         00001120
C      ISCID        1        17      SPACECRAFT IDENTIFIER FOR TABLE LOOKUP 00001130
C       NPAR        1        18      NUMBER OF PARALLEL PROCESSOR ELEMENTS  00001140
```

```
ICNTRL              JSS1493.PRLLEL.ORBDET.FORT

C                                   IN THE POSTULATED HARDWARE ARCHITECTURE00001150
C      ICOV         1        19     FLAG TO INDICATE THATTHE ESTIMATION    00001160
C                                   ERROR COVARIANCE MATRIX IS TO BE UP-   00001170
C                                   DATED AFTER THE ESTIMATION RUN:        00001180
C                                   0=DO NOT UPDATE COVARIANCE MATRIX      00001190
C                                   1=UPDATE THE COVARIANCE MATRIX         00001200
C                                                                          00001210
      DATA IRNTYP / 1           /                                          00001220
C                                                                          00001230
      DATA IVINIT / 1           /                                          00001240
C                                                                          00001250
      DATA IVINTG / 1           /                                          00001260
C                                                                          00001270
      DATA ISLVFR / 3           /                                          00001280
C                                                                          00001290
      DATA NSLVFR / 6           /                                          00001300
C                                                                          00001310
      DATA IPTLS  / 0           /                                          00001320
C                                                                          00001330
      DATA IJ2RTS / 0           /                                          00001340
C                                                                          00001350
      DATA IRVPAR / 0           /                                          00001360
C                                                                          00001370
      DATA IPARRV / 0           /                                          00001380
C                                                                          00001390
      DATA IPRNT1 / 2           /                                          00001400
C                                                                          00001410
      DATA IPRNT2 / 0           /                                          00001420
C                                                                          00001430
      DATA IPRNT3 / 1           /                                          00001440
C                                                                          00001450
      DATA IPRNT4 / 0           /                                          00001460
C                                                                          00001470
      DATA IPRNT5 / 0           /                                          00001480
C                                                                          00001490
      DATA IOUT   / 2           /                                          00001500
C                                                                          00001510
      DATA ICGCMP / 2           /                                          00001520
C                                                                          00001530
      DATA ISCID  / 0           /                                          00001540
C                                                                          00001550
      DATA NPAR   / 6           /                                          00001560
C                                                                          00001570
      DATA ICOV   / 1           /                                          00001580
C                                                                          00001590
      END                                                                  00001600

******** END OF MEMBER ICNTRL     160 RECORDS ********
```

INPRD1 JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE INPRD1(X,Y,Z,B)                                        00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE INNER (DOT) PRODUCT OF TWO VECTORS X AND Y,    00000060
C         DEFINED ON THE SET OF BASIS VECTORS B.                        00000065
C                                                                       00000070
C     CALLING SEQUENCE                                                  00000080
C         CALL INPRD1(X,Y,Z,B)                                          00000090
C                                                                       00000100
C     CALLING SEQUENCE PARAMETERS                                       00000110
C         X,Y         I       VECTORS (UP TO 6X1) FOR WHICH THE INNER   00000120
C                             PRODUCT IS TO BE FOUND.                   00000130
C         B           I       SET OF BASIS VECTORS ON WHICH X AND Y ARE 00000140
C                             DEFINED.                                  00000145
C         Z           O       SCALAR REPRESENTING THE DOT PRODUCT OF    00000150
C                             THE INPUT VECTORS X AND Y:                00000160
C                                                        T       T      00000170
C                                 Z = <X,Y> = <Y,X> = X * Y = Y * X     00000180
C                                                                       00000190
C     EXTERNAL REFERENCES                                               00000200
C         NONE                                                          00000210
C                                                                       00000220
C     ANALYSIS                                                          00000230
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000240
C                                                                       00000250
C     PROGRAMMING                                                       00000260
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000270
C                                                                       00000280
      IMPLICIT REAL*8(A-H,O-Z)                                          00000290
      DIMENSION       X(6)        ,  Y(6)     ,  B(6,6)                 00000300
      Z = 0.D0                                                          00000310
      DO  10   I = 1,6                                                  00000320
      Z = Z + X(I) * Y(I) * B(I,I)**2                                   00000330
   10 CONTINUE                                                          00000340
      RETURN                                                            00000350
      END                                                               00000360

******** END OF MEMBER INPRD1      38 RECORDS ********
```

```
INPROD              JSS1493.PRLLEL.ORBDET.FORT

      SUBROUTINE INPROD(X,Y,Z,N)                                          00000010
C                                                                         00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                       00000030
C                                                                         00000040
C     PURPOSE                                                             00000050
C        TO COMPUTE THE INNER (DOT) PRODUCT OF TWO VECTORS X AND Y.       00000060
C                                                                         00000070
C     CALLING SEQUENCE                                                    00000080
C        CALL INPROD(X,Y,Z,N)                                             00000090
C                                                                         00000100
C     CALLING SEQUENCE PARAMETERS                                         00000110
C        X,Y         I        VECTORS (UP TO 6X1) FOR WHICH THE INNER    00000120
C                             PRODUCT IS TO BE FOUND.                    00000130
C        N           I        DIMENSION OF THE VECTORS X AND Y.          00000140
C        Z           O        SCALAR REPRESENTING THE DOT PRODUCT OF     00000150
C                             THE INPUT VECTORS X AND Y:                 00000160
C                                                      T       T          00000170
C                                 Z = <X,Y> = <Y,X> = X * Y = Y * X       00000180
C                                                                         00000190
C     EXTERNAL REFERENCES                                                 00000200
C        NONE                                                             00000210
C                                                                         00000220
C     ANALYSIS                                                            00000230
C        CAPTAIN JEFFREY S. SHAVER, USAF                                  00000240
C                                                                         00000250
C     PROGRAMMING                                                         00000260
C        CAPTAIN JEFFREY S. SHAVER, USAF                                  00000270
C                                                                         00000280
      IMPLICIT REAL*8(A-H,O-Z)                                            00000290
      DIMENSION       X(6)        ,  Y(6)                                 00000300
      Z = 0.D0                                                            00000310
      DO  10   I = 1,N                                                    00000320
      Z = Z + X(I) * Y(I)                                                 00000330
   10 CONTINUE                                                            00000340
      RETURN                                                              00000350
      END                                                                 00000360

******** END OF MEMBER INPROD      36 RECORDS ********
```

```
      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK ISTSTC  00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/ISTSTC/   ITCNT     ,   IF1CNT     ,   IG1CNT     ,       00000120
     *                 IF2CNT     ,   IG2CNT     ,   ICCNT      ,       00000130
     *                 IFG1WO     ,   IFG2WO     ,   IFG1W      ,       00000140
     *                 IFG2W      ,   IT1WO      ,   IT2WO      ,       00000150
     *                 IT1W       ,   IT2W                              00000160
C                                                                       00000170
C                      #LOCA-                                           00000180
C             DIMEN-     TION                                           00000190
C  VARIABLE    SION    INTEGER                DESCRIPTION               00000200
C  ________   ______   _______   ____________________________________   00000210
C                                                                       00000220
C     ITCNT        1         1   PVM ITERATION COUNTER                  00000230
C    IF1CNT        1         2   NUMBER OF COST FUNCTIONAL EVALUATIONS  00000240
C                                 PERFORMED IN A SEQUENTIAL EXECUTION   00000250
C                                 OF A SINGLE ITERATION OF THE PVM      00000260
C                                 ESTIMATION ALGORITHM.                 00000270
C    IG1CNT        1         3   NUMBER OF GRADIENT EVALUATIONS PER-    00000280
C                                 FORMED IN A SEQUENTIAL EXECUTION OF   00000290
C                                 A SINGLE ITERATION OF THE PVM         00000300
C                                 ESTIMATION ALGORITHM.                 00000310
C    IF2CNT        1         4   EFFECTIVE NUMBER OF COST FUNCTIONAL    00000320
C                                 EVALUATIONS PERFORMED IN A PARALLEL   00000330
C                                 EXECUTION OF A SINGLE ITERATION OF THE00000340
C                                 PVM ESTIMATION ALGORITHM.             00000350
C    IG2CNT        1         5   EFFECTIVE NUMBER OF GRADIENT EVALUA-   00000360
C                                 TIONS PERFORMED IN A PARALLEL EXECU-  00000370
C                                 TION OF A SINGLE ITERATION OF THE PVM 00000380
C                                 ESTIMATION ALGORITHM.                 00000390
C     ICCNT        1         6   NUMBER OF COST FUNCTIONAL EVALUATIONS  00000400
C                                 REQUIRED FOR THE UNIVARIATE SEARCH IN 00000410
C                                 ONE ITERATION OF THE PVM ALGORITHM.   00000420
C    IFG1WO        1         7   NUMBER OF COST AND GRADIENT EVALUATIONS00000430
C                                 IN A SEQUENTIAL EXECUTION OF ONE ITER-00000440
C                                 ATION OF THE PVM ALGORITHM, EXCLUDING 00000450
C                                 THE UNIVARIATE SEARCH.                00000460
C    IFG2WO        1         8   EFFECTIVE NUMBER OF COST AND GRADIENT  00000470
C                                 EVALUATIONS IN A PARALLEL EXECUTION OF00000480
C                                 ONE ITERATION OF THE PVM ALGORITHM,   00000490
C                                 EXCLUDING THE UNIVARIATE SEARCH.      00000500
C     IFG1W        1         9   NUMBER OF COST AND GRADIENT EVALUATIONS00000510
C                                 IN A SEQUENTIAL EXECUTION OF ONE ITER-00000520
C                                 ATION OF THE PVM ALGORITHM, INCLUDING 00000530
C                                 THE UNIVARIATE SEARCH.                00000540
C     IFG2W        1        10   EFFECTIVE NUMBER OF COST AND GRADIENT  00000550
C                                 EVALUATIONS IN A PARALLEL EXECUTION   00000560
C                                 OF ONE ITERATION OF THE PVM ALGORITHM, 00000570
```

```
ISTSTC              JSS1493.PRLLEL.ORBDET.FORT

C                                    INCLUDING THE UNIVARIATE SEARCH.        00000580
C     IT1WO        1        11     TOTAL NUMBER OF COST AND GRADIENT       00000590
C                                    EVALUATIONS IN A SEQUENTIAL EXECUTION   00000600
C                                    OF THE PVM ALGORITHM, EXCLUDING THE     00000610
C                                    UNIVARIATE SEARCH.                      00000620
C     IT2WO        1        12     TOTAL NUMBER OF EFFECTIVE COST AND      00000630
C                                    GRADIENT EVALUATIONS IN A PARALLEL      00000640
C                                    EXECUTION OF THE PVM ALGORITHM, EX-     00000650
C                                    CLUDING THE UNIVARIATE SEARCH.          00000660
C      IT1W        1        13     TOTAL NUMBER OF COST AND GRADIENT       00000670
C                                    EVALUATIONS IN A SEQUENTIAL EXECUTION   00000680
C                                    OF THE PVM ALGORITHM, INCLUDING THE     00000690
C                                    UNIVARIATE SEARCH.                      00000700
C      IT2W        1        14     TOTAL NUMBER OF EFFECTIVE COST AND      00000710
C                                    GRADIENT EVALUATIONS IN A PARALLEL      00000720
C                                    EXECUTION OF THE PVM ALGORITHM, IN-     00000730
C                                    CLUDING THE UNIVARIATE SEARCH.          00000740
C                                                                            00000750
      DATA ITCNT  / 0              /                                         00000760
C                                                                            00000770
      DATA IF1CNT / 0              /                                         00000780
C                                                                            00000790
      DATA IG1CNT / 0              /                                         00000800
C                                                                            00000810
      DATA IF2CNT / 0              /                                         00000820
C                                                                            00000830
      DATA IG2CNT / 0              /                                         00000840
C                                                                            00000850
      DATA ICCNT  / 0              /                                         00000860
C                                                                            00000870
      DATA IFG1WO / 0              /                                         00000880
C                                                                            00000890
      DATA IFG2WO / 0              /                                         00000900
C                                                                            00000910
      DATA IFG1W  / 0              /                                         00000920
C                                                                            00000930
      DATA IFG2W  / 0              /                                         00000940
C                                                                            00000950
      DATA IT1WO  / 0              /                                         00000960
C                                                                            00000970
      DATA IT2WO  / 0              /                                         00000980
C                                                                            00000990
      DATA IT1W   / 0              /                                         00001000
C                                                                            00001010
      DATA IT2W   / 0              /                                         00001020
C                                                                            00001030
      END                                                                    00001040

******** END OF MEMBER ISTSTC     104 RECORDS ********
```

KEPLR JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE KEPLR(ANOMM,ECC,CONV,ANOME,ANOMF)                      00000010
C                                                                       00000020
C     VERSION OF 3/21/79      **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE ECCENTRIC AND TRUE ANOMALIES, GIVEN THE MEAN   00000060
C         ANOMALY AND THE ECCENTRICITY FOR A KEPLERIAN ORBIT BY         00000070
C         SOLVING KEPLER'S EQUATION:                                    00000080
C                                                                       00000090
C                          M = E - ECC*SIN(E)                           00000100
C                                                                       00000110
C         THIS TRANCENDENTAL EQUATION IS SOLVED BY FORMING THE          00000120
C         FUNCTION:                                                     00000130
C                                                                       00000140
C                      F(E) = M - E + ECC*SIN(E)                        00000150
C                                                                       00000160
C         AND THEN USING NEWTON'S METHOD TO FIND THE VALUE OF E FOR     00000170
C         WHICH F(E) = 0, GIVEN M AND ECC.  NEWTON'S METHOD INVOLVES    00000180
C         THE ITERATIVE SOLUTION OF THE FOLLOWING EQUATION:             00000190
C                                                                       00000200
C                        E    = E  - F(E )/F'(E )                     00000210
C                         N+1   N      N      N                        00000220
C                                                                       00000230
C         WHERE       F(E ) = M - E  + ECC*SIN(E )                    00000240
C                       N         N            N                      00000250
C                                                                       00000260
C              AND  F'(E ) = ECC*COS(E ) - 1.0                          00000270
C                     N             N                                 00000280
C                                                                       00000290
C         THE INITIAL GUESS FOR THE ECCENTRIC ANOMALY IS TAKEN TO BE    00000300
C         THE KNOWN MEAN ANOMALY (E  = M), SINCE THE DIFFERENCE IS      00000310
C                                 0                                     00000320
C         OF THE ORDER OF THE ECCENTRICITY.                             00000330
C                                                                       00000340
C     CALLING SEQUENCE                                                  00000350
C         CALL KEPLR(ANOMM,ECC,CONV,ANOME,ANOMF)                        00000360
C                                                                       00000370
C     PARAMETERS                                                        00000380
C         ANOMM     I     VALUE OF THE MEAN ANOMALY                   00000390
C         ECC       I     VALUE OF THE ECCENTRICITY OF THE ORBIT      00000400
C         CONV      I     CONVERGENCE CRITERION FOR THE ITERATIVE     00000410
C                         PROCESS                                     00000420
C         ANOME     O     COMPUTED VALUE OF THE ECCENTRIC ANOMALY     00000430
C         ANOMF     O     COMPUTED VALUE OF THE TRUE ANOMALY          00000440
C                                                                       00000450
C     EXTERNAL REFERENCES                                               00000460
C         NONE                                                          00000470
C                                                                       00000480
C     ANALYSIS                                                          00000490
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000500
C                                                                       00000510
C     PROGRAMMER                                                        00000520
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000530
C                                                                       00000540
      IMPLICIT REAL*8(A-H,O-Z)                                          00000550
      LOGICAL       CON1,CON2                                           00000560
      PI = 3.141592653589793D0                                          00000570
```

```
C                                                                       00000580
C     INITIALIZE FOR ITERATIVE PROCESS                                  00000590
C                                                                       00000600
      ANOME = ANOMM                                                     00000610
      CON1  = .FALSE.                                                   00000620
      CON2  = .FALSE.                                                   00000630
      ICNT  = 0                                                         00000640
C                                                                       00000650
C     BEGIN ITERATION                                                   00000660
C                                                                       00000670
   10 ICNT  = ICNT + 1                                                  00000680
      F = ANOMM - ANOME + ECC * DSIN(ANOME)                             00000690
      FPRIME = ECC * DCOS(ANOME) - 1.D0                                 00000700
      ANOME1 = ANOME - F/FPRIME                                         00000710
      ERR = DABS(ANOME1 - ANOME)                                        00000720
C                                                                       00000730
C     CHECK FOR CONVERGENCE ON THIS ITERATION                           00000740
C                                                                       00000750
      CON2 = ERR.LT.CONV                                                00000760
      IF (CON1.AND.CON2)  GO TO 20                                      00000770
      CON1 = CON2                                                       00000780
      ANOME = ANOME1                                                    00000790
      GO TO 10                                                          00000800
C                                                                       00000810
C     CONVERT TO TRUE ANOMALY AND OUTPUT RESULTS                        00000820
C                                                                       00000830
   20 CF = (DCOS(ANOME1)-ECC)/(1.D0-ECC*DCOS(ANOME1))                   00000840
      SF = (DSQRT(1.D0-ECC**2)*DSIN(ANOME1))/(1.D0-ECC*DCOS(ANOME1))    00000850
      ANOMF = DATAN2(SF,CF)                                             00000860
      IF (ANOMF.LT.0.D0) ANOMF = 2.D0 * PI + ANOMF                      00000870
C     AMDEG = ANOMM  * 180.D0/PI                                        00000880
C     AEDEG = ANOME1 * 180.D0/PI                                        00000890
C     AFDEG = ANOMF  * 180.D0/PI                                        00000900
C     PRINT  100,ICNT,ECC,AMDEG,AEDEG,AFDEG                             00000910
      RETURN                                                            00000920
C                                                                       00000930
C               FORMAT STATEMENT                                        00000940
C                                                                       00000950
  100 FORMAT(10X,'THE ITERATIVE SOLUTION CONVERGED IN ',I2,' ITERA',    00000960
     1           'TIONS.'/19X,'ECCENTRICITY',19X,'MEAN ANOMALY',19X,    00000970
     2           'ECCENTRIC ANOMALY',19X,'TRUE ANOMALY'/12X,1PD25.15,   00000980
     3           6X,1PD25.15,6X,1PD25.15,6X,1PD25.15//)                 00000990
C                                                                       00001000
      END                                                               00001010
```

******** END OF MEMBER KEPLR 101 RECORDS ********

```
      SUBROUTINE LEGPRM(ARG,K,PPK,PPKMN1,PPKMN2)                        00000010
C                                                                       00000020
C     VERSION OF 08/10/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO COMPUTE THE DERIVATIVES OF THE LEGENDRE POLYNOMIALS BY      00000060
C        USING A RECURSION RELATION                                     00000070
C                                                                       00000080
C     METHOD                                                            00000090
C        THE RECURSION USED IN THIS PROGRAM IS:                         00000100
C                                                                       00000110
C              '                '              '                       00000120
C        (K-1)*P (X) - (2K-1)*X*P   (X) + K*P   (X) = 0   FOR K.GE.2    00000130
C              K                K-1           K-2                      00000140
C                                                                       00000150
C                                              '                        00000160
C                        WITH STARTING VALUES  P (X) = 0                00000170
C                                               0                       00000180
C                                                                       00000190
C                                              '                        00000200
C                                             P (X) = 1                 00000210
C                                              1                        00000220
C                                                                       00000230
C     CALLING SEQUENCE                                                  00000240
C        CALL LEGPRM(ARG,K,PPK,PPKMN1,PPKMN2)                           00000250
C                                                                       00000260
C     CALLING SEQUENCE PARAMETERS                                       00000270
C        ARG      I    VALUE OF THE ARGUMENT OF THE LEGENDRE POLYNOMIALS00000280
C        K        I    ORDER OF THE LEGENDRE POLYNOMIAL WHOSE DERIVATIVE00000290
C                      IS BEING COMPUTED (K.GE.2)                       00000300
C        PPK      O    VALUE OF THE DERIVATIVE OF THE KTH LEGENDRE POLY-00000310
C                      NOMIAL                                           00000320
C        PPKMN1   O    VALUE OF THE DERIVATIVE OF THE (K-1)ST LEGENDRE  00000330
C                      POLYNOMIAL                                       00000340
C        PPKMN2   O    VALUE OF THE DERIVATIVE OF THE (K-2)ND LEGENDRE  00000350
C                      POLYNOMIAL                                       00000360
C                                                                       00000370
C     EXTERNAL REFERENCES                                               00000380
C        NONE                                                           00000390
C                                                                       00000400
C     ANALYSIS                                                          00000410
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000420
C                                                                       00000430
C     PROGRAMMING                                                       00000440
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000450
C                                                                       00000460
      IMPLICIT REAL*8(A-H,O-Z)                                          00000470
C                                                                       00000480
C     INITIALIZE THE RECURSION                                          00000490
C                                                                       00000500
      KSAVE = K                                                         00000510
      K = 2                                                             00000520
      PPKMN1 = 1.D0                                                     00000530
      PPKMN2 = 0.D0                                                     00000540
C                                                                       00000550
C     BEGIN RECURSIVE COMPUTATION                                       00000560
C                                                                       00000570
```

```
LEGPRM              JSS1493.PRLLEL.ORBDET.FORT

   10 DK    = DFLOAT(K)                                                  00000580
      COF1 = (2.D0 * DK - 1.D0)/(DK - 1.D0)                             00000590
      COF2 = DK/(DK - 1.D0)                                             00000600
      PPK  = COF1 * ARG * PPKMN1  -  COF2 * PPKMN2                      00000610
      IF (K.GE.KSAVE)                  RETURN                           00000620
      PPKMN2 = PPKMN1                                                   00000630
      PPKMN1 = PPK                                                      00000640
      K = K + 1                                                         00000650
      GO TO 10                                                          00000660
      END                                                               00000670

******** END OF.MEMBER LEGPRM      67 RECORDS ********
```

```
      SUBROUTINE LGRNGE(S0,TA0,GM,TA,SLR,AM,ST)                          00000010
C                                                                        00000020
C     VERSION OF 3/21/79      **** CHECKED OUT ****                      00000030
C                                                                        00000040
C     PURPOSE                                                            00000050
C         TO COMPUTE THE COMPONENTS OF THE INERTIAL CARTESIAN POSITION   00000060
C         AND VELOCITY VECTORS AT SOME GIVEN VALUE OF TRUE ANOMALY FOR   00000070
C         THE UNPERTURBED TWO BODY PROBLEM, GIVEN SOME EPOCH STATE       00000080
C         VECTOR AND EPOCH TRUE ANOMALY.                                 00000090
C                                                                        00000100
C         THE METHOD USED IS THAT OF COMPUTING THE LAGRANGE COEFFICIENTS 00000110
C         AND USING THEM TO DETERMINE THE NEW POSITION AND VELOCITY      00000120
C         VECTORS FROM THE GIVEN EPOCH STATE.                            00000130
C                                                                        00000140
C     CALLING SEQUENCE                                                   00000150
C         CALL LGRNGE(S0,TA0,GM,TA,SLR,AM,ST)                            00000160
C                                                                        00000170
C     PARAMETERS                                                         00000180
C         S0   I    ARRAY (6X1) CONTAINING THREE COMPONENTS OF EPOCH     00000190
C                   POSITION VECTOR AND THREE COMPONENTS OF EPOCH         00000200
C                   VELOCITY VECTOR.                                     00000210
C         TA0  I    TRUE ANOMALY AT EPOCH.                               00000220
C         GM   I    GRAVITATIONAL PARAMETER.                             00000230
C         TA   I    VALUE OF TRUE ANOMALY FOR WHICH WE WANT TO COMPUTE   00000240
C                   POSITION AND VELOCITY VECTORS.                       00000250
C         SLR  I    SEMI-LATUS RECTUM (PARAMETER).                       00000260
C         AM   I    MAGNITUDE OF ANGULAR MOMENTUM VECTOR.                00000270
C         ST   O    ARRAY (6X1) CONTAINING THREE COMPONENTS OF POSITION  00000280
C                   AND THREE COMPONENTS OF VELOCITY ASSOCIATED WITH     00000290
C                   THE GIVEN TRUE ANOMALY.                              00000300
C                                                                        00000310
C     EXTERNAL REFERENCES                                                00000320
C         NONE                                                           00000330
C                                                                        00000340
C     ANALYSIS                                                           00000350
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000360
C                                                                        00000370
C     PROGRAMMER                                                         00000380
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000390
C                                                                        00000400
      IMPLICIT REAL*8(A-H,O-Z)                                           00000410
      DIMENSION     S0(6),ST(6)                                          00000420
      PI = 3.14159265358979320                                           00000430
      DELF = TA - TA0                                                    00000440
      RDOTV = S0(1)*S0(4) + S0(2)*S0(5) + S0(3)*S0(6)                    00000450
      R0 = DSQRT(S0(1)**2 + S0(2)**2 + S0(3)**2)                         00000460
C                                                                        00000470
C     COMPUTE MAGNITUDE OF THE POSITION VECTOR                           00000480
C                                                                        00000490
      R=(SLR*R0)/(R0+(SLR-R0)*DCOS(DELF)-(AM/GM)*RDOTV*DSIN(DELF))       00000500
C                                                                        00000510
C     COMPUTE THE LAGRANGE COEFFICIENTS                                  00000520
C                                                                        00000530
      F  = 1.D0 - (R/SLR) * (1.D0 - DCOS(DELF))                          00000540
      G  = (R * R0/AM) * DSIN(DELF)                                      00000550
      FT = (RDOTV/(SLR*R0))*(1.D0-DCOS(DELF))-(AM/(SLR*R0))*DSIN(DELF)   00000560
      GT = 1.D0 - (R0/SLR) * (1.D0 - DCOS(DELF))                         00000570
```

```
LGRNGE              JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000580
C     COMPUTE POSITION AND VELOCITY VECTORS                             00000590
C                                                                       00000600
      DO 10    I = 1,3                                                  00000610
      ST(I)   = F * S0(I) + G * S0(I+3)                                 00000620
      ST(I+3) = FT* S0(I) + GT* S0(I+3)                                 00000630
   10 CONTINUE                                                          00000640
C                                                                       00000650
C     OUTPUT RESULTS                                                    00000660
C                                                                       00000670
C     TA0DEG = TA0 * 180.D0/PI                                          00000680
C     TADEG  = TA * 180.D0/PI                                           00000690
C     PRINT 100,TA0DEG,TADEG,ST(1),ST(2),ST(3),ST(4),ST(5),ST(6)        00000700
      RETURN                                                            00000710
C                                                                       00000720
C                  FORMAT STATEMENT                                     00000730
C                                                                       00000740
  100 FORMAT(10X,'EPOCH TRUE ANOMALY =',1PD25.15/10X,'CURRENT ',        00000750
     1       'TRUE ANOMALY =',1PD25.15/10X,'RX =',1PD25.15/10X,         00000760
     2       'RY =',1PD25.15/10X,'RZ =',1PD25.15/10X,'VX =',1PD25.15/   00000770
     3       10X,'VY =',1PD25.15/10X,'VZ =',1PD25.15////)               00000780
C                                                                       00000790
      END                                                               00000800

******** END OF MEMBER LGRNGE      80 RECORDS ********
```

```
MAIN                JSS1493.PRLLEL.ORBDET.FORT

C     MAIN.FORT                                                          00000010
C                                                                        00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                      00000030
C                                                                        00000040
C     PURPOSE                                                            00000050
C         TO ACT AS THE OVERALL DRIVER OF THE PARALLEL PROCESSING ORBIT  00000060
C         DETERMINATION SYSTEM                                           00000070
C                                                                        00000080
C     EXTERNAL REFERENCES                                                00000090
C         SUBROUTINES                                                    00000100
C              EPHEX                                                     00000110
C              ESTEX                                                     00000120
C              RUNSET                                                    00000140
C                                                                        00000150
C         COMMON BLOCKS                                                  00000160
C              /ICNTRL/                                                  00000170
C                                                                        00000180
C     ANALYSIS                                                           00000190
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000200
C                                                                        00000210
C     PROGRAMMING                                                        00000220
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00000230
C                                                                        00000240
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000250
C                                                                        00000260
      IMPLICIT REAL*8(A-H,O-Z)                                           00000270
C                                                                        00000280
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000290
C                                                                        00000300
      COMMON/ICNTRL/  IRNTYP     ,  IVINIT     ,  IVINTG     ,         00000310
     1                ISLVFR     ,  NSLVFR     ,  IPTLS      ,         00000320
     2                IJ2RTS     ,  IRVPAR     ,  IPARRV     ,         00000330
     3                IPRNT1     ,  IPRNT2     ,  IPRNT3     ,         00000340
     4                IPRNT4     ,  IPRNT5     ,  IOUT       ,         00000350
     5                ICGCMP     ,  ISCID      ,  NPAR                 00000360
C                                                                        00000370
C************************************************************************00000380
C******************                                  ******************00000390
C******************      RUN SET-UP PROCESSING       ******************00000400
C******************                                  ******************00000410
C************************************************************************00000420
C                                                                        00000430
      CALL RUNSET                                                        00000440
C                                                                        00000490
C************************************************************************00000500
C******************                                  ******************00000510
C******************    RUN EXECUTION PROCESSING      ******************00000520
C******************                                  ******************00000530
C************************************************************************00000540
C                                                                        00000550
      GO TO (10,20) , IRNTYP                                             00000560
C                                                                        00000570
C     PARALLEL EPHEMERIS GENERATION PROCESSING                           00000580
C                                                                        00000590
   10 CALL EPHEX                                                         00000600
      STOP                                                               00000610
C                                                                        00000620
```

```
MAIN                JSS1493.PRLLEL.ORBDET.FORT

C     PARALLEL VARIABLE METRIC ESTIMATION PROCESSING                    00000630
C                                                                       00000640
   20 CALL ESTEX                                                        00000650
      STOP                                                              00000660
C                                                                       00000670
      END                                                               00000680

******** END OF MEMBER MAIN        63 RECORDS ********
```

```
MATINV              JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000100
C=======================================================================00000200
C    WARNING:  ACCORDING TO STANFORD UNIVERSITY, THIS ROUTINE CONTAINS  00000300
C              A DEFECT OR DEFICIENCY.  PLEASE REFER TO DCD "DIGITAL    00000400
C              ENTOMOLOGIST" NUMBER 45, OR SEE THE PROGRAM LIBRARIAN.   00000500
C=======================================================================00000600
C     .................................................................00000700
C                                                                       00000800
C        SUBROUTINE MINV                                                00000900
C                                                                       00001000
C        PURPOSE                                                        00001100
C           INVERT A MATRIX                                             00001200
C                                                                       00001300
C        USAGE                                                          00001400
C           CALL MINV(A,N,D,L,M)                                        00001500
C                                                                       00001600
C        DESCRIPTION OF PARAMETERS                                      00001700
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY  00001800
C               RESULTANT INVERSE.                                      00001900
C           N - ORDER OF MATRIX A                                       00002000
C           D - RESULTANT DETERMINANT                                   00002100
C           L - WORK VECTOR OF LENGTH N                                 00002200
C           M - WORK VECTOR OF LENGTH N                                 00002300
C                                                                       00002400
C        REMARKS                                                        00002500
C           MATRIX A MUST BE A GENERAL MATRIX                           00002600
C                                                                       00002700
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  00002800
C           NONE                                                        00002900
C                                                                       00003000
C        METHOD                                                         00003100
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT   00003200
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT    00003300
C           THE MATRIX IS SINGULAR.                                     00003400
C                                                                       00003500
C     .................................................................00003600
C                                                                       00003700
      SUBROUTINE MATINV(A,N,D,L,M)                                      00003800
      DIMENSION A(1),L(1),M(1)                                          00003900
C                                                                       00004000
C        .............................................................00004100
C                                                                       00004200
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE  00004300
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION      00004400
C        STATEMENT WHICH FOLLOWS.                                       00004500
C                                                                       00004600
C     DOUBLE PRECISION A,D,BIGA,HOLD                                    00004700
C                                                                       00004800
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS    00004900
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS      00005000
C        ROUTINE.                                                       00005100
C                                                                       00005200
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO      00005300
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT  00005400
C        10 MUST BE CHANGED TO DABS.                                    00005500
C                                                                       00005600
C        .............................................................00005700
```

```
C                                                                       00005800
C                                                                       00005900
C  THIS CODE MAY BE COMMENTED OUT BY UNDAUNTED USERS 7-78 PDM           00006000
C     LOGICAL*1  $FIRST/.FALSE./                                        00006100
C     IF ($FIRST)  GO TO 1                                              00006200
C        WRITE (6,9999)                                                 00006300
C9999    FORMAT ('0-----------------------------------------' /         00006400
C    .           ' MINV/DMINV DEFECTIVE, SEE PROGRAM LIBRARIAN' /       00006500
C    .           ' -----------------------------------------')         00006600
C        $FIRST = .TRUE.                                                00006700
C   1 CONTINUE                                                          00006800
C                                                                       00006900
C        SEARCH FOR LARGEST ELEMENT                                     00007000
C                                                                       00007100
      D=1.0                                                             00007200
      NK=-N                                                             00007300
      DO 80 K=1,N                                                       00007400
      NK=NK+N                                                           00007500
      L(K)=K                                                            00007600
      M(K)=K                                                            00007700
      KK=NK+K                                                           00007800
      BIGA=A(KK)                                                        00007900
      DO 20 J=K,N                                                       00008000
      IZ=N*(J-1)                                                        00008100
      DO 20 I=K,N                                                       00008200
      IJ=IZ+I                                                           00008300
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20                               00008400
   15 BIGA=A(IJ)                                                        00008500
      L(K)=I                                                            00008600
      M(K)=J                                                            00008700
   20 CONTINUE                                                          00008800
C                                                                       00008900
C        INTERCHANGE ROWS                                               00009000
C                                                                       00009100
      J=L(K)                                                            00009200
      IF(J-K) 35,35,25                                                  00009300
   25 KI=K-N                                                            00009400
      DO 30 I=1,N                                                       00009500
      KI=KI+N                                                           00009600
      HOLD=-A(KI)                                                       00009700
      JI=KI-K+J                                                         00009800
      A(KI)=A(JI)                                                       00009900
   30 A(JI) =HOLD                                                       00010000
C                                                                       00010100
C        INTERCHANGE COLUMNS                                            00010200
C                                                                       00010300
   35 I=M(K)                                                            00010400
      IF(I-K) 45,45,38                                                  00010500
   38 JP=N*(I-1)                                                        00010600
      DO 40 J=1,N                                                       00010700
      JK=NK+J                                                           00010800
      JI=JP+J                                                           00010900
      HOLD=-A(JK)                                                       00011000
      A(JK)=A(JI)                                                       00011100
   40 A(JI) =HOLD                                                       00011200
C                                                                       00011300
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS        00011400
```

```
C        CONTAINED IN BIGA)                                          00011500
C                                                                    00011600
   45 IF(BIGA) 48,46,48                                              00011700
   46 D=0.0                                                          00011800
      RETURN                                                         00011900
   48 DO 55 I=1,N                                                    00012000
      IF(I-K) 50,55,50                                               00012100
   50 IK=NK+I                                                        00012200
      A(IK)=A(IK)/(-BIGA)                                            00012300
   55 CONTINUE                                                       00012400
C                                                                    00012500
C        REDUCE MATRIX                                               00012600
C                                                                    00012700
      DO 65 I=1,N                                                    00012800
      IK=NK+I                                                        00012900
      HOLD=A(IK)                                                     00013000
      IJ=I-N                                                         00013100
      DO 65 J=1,N                                                    00013200
      IJ=IJ+N                                                        00013300
      IF(I-K) 60,65,60                                               00013400
   60 IF(J-K) 62,65,62                                               00013500
   62 KJ=IJ-I+K                                                      00013600
      A(IJ)=HOLD*A(KJ)+A(IJ)                                         00013700
   65 CONTINUE                                                       00013800
C                                                                    00013900
C        DIVIDE ROW BY PIVOT                                         00014000
C                                                                    00014100
      KJ=K-N                                                         00014200
      DO 75 J=1,N                                                    00014300
      KJ=KJ+N                                                        00014400
      IF(J-K) 70,75,70                                               00014500
   70 A(KJ)=A(KJ)/BIGA                                               00014600
   75 CONTINUE                                                       00014700
C                                                                    00014800
C        PRODUCT OF PIVOTS                                           00014900
C                                                                    00015000
      D=D*BIGA                                                       00015100
C                                                                    00015200
C        REPLACE PIVOT BY RECIPROCAL                                 00015300
C                                                                    00015400
      A(KK)=1.0/BIGA                                                 00015500
   80 CONTINUE                                                       00015600
C                                                                    00015700
C        FINAL ROW AND COLUMN INTERCHANGE                            00015800
C                                                                    00015900
      K=N                                                            00016000
  100 K=(K-1)                                                        00016100
      IF(K) 150,150,105                                              00016200
  105 I=L(K)                                                         00016300
      IF(I-K) 120,120,108                                            00016400
  108 JQ=N*(K-1)                                                     00016500
      JR=N*(I-1)                                                     00016600
      DO 110 J=1,N                                                   00016700
      JK=JQ+J                                                        00016800
      HOLD=A(JK)                                                     00016900
      JI=JR+J                                                        00017000
      A(JK)=-A(JI)                                                   00017100
```

```
MATINV              JSS1493.PRLLEL.ORBDET.FORT

  110 A(JI) =HOLD                                                    00017200
  120 J=M(K)                                                         00017300
      IF(J-K) 100,100,125                                            00017400
  125 KI=K-N                                                         00017500
      DO 130 I=1,N                                                   00017600
      KI=KI+N                                                        00017700
      HOLD=A(KI)                                                     00017800
      JI=KI-K+J                                                      00017900
      A(KI)=-A(JI)                                                   00018000
  130 A(JI) =HOLD                                                    00018100
      GO TO 100                                                      00018200
  150 RETURN                                                         00018300
      END                                                            00018400

******** END OF MEMBER MATINV     184 RECORDS ********
```

M6666 JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE M6666(A,B,C)                                           00000010
C                                                                       00000020
C     VERSION OF 08/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO MULTIPLY TWO 6X6 MATRICES:  A * B = C                      00000060
C                                                                       00000070
      IMPLICIT REAL*8(A-H,O-Z)                                          00000080
      DIMENSION       A(6,6)  ,  B(6,6)  ,  C(6,6)                      00000090
C                                                                       00000100
C     INITIALIZE ELEMENTS IN RESULTS MATRIX TO ZERO                     00000110
C                                                                       00000120
      DO  20   I = 1,6                                                  00000130
      DO  10   J = 1,6                                                  00000140
   10 C(I,J) = 0.D0                                                     00000150
   20 CONTINUE                                                          00000160
C                                                                       00000170
C     COMPUTE ELEMENT C(I,J) BY ACCUMULATING THE PRODUCT OF THE ITH ROW 00000180
C     OF A WITH THE JTH COLUMN OF B                                     00000190
C                                                                       00000200
      DO  50   I = 1,6                                                  00000210
      DO  40   J = 1,6                                                  00000220
      DO  30   K = 1,6                                                  00000230
   30 C(I,J) = C(I,J) + A(I,K) * B(K,J)                                 00000240
   40 CONTINUE                                                          00000250
   50 CONTINUE                                                          00000260
      RETURN                                                            00000270
      END                                                               00000280

******** END OF MEMBER M6666       28 RECORDS ********
```

```
NPCNTL              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK NPCNTL   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/NPCNTL/   MORD     ,     NSUB     ,     IEQSET     ,      00000120
     *                 ITERM                                          00000130
C                                                                       00000140
C                       #LOCA-                                          00000150
C              DIMEN-     TION                                         00000160
C  VARIABLE     SION    INTEGER                 DESCRIPTION            00000170
C  ________    _____    _______  ____________________________________  00000180
C                                                                       00000190
C       MORD         1         1  ORDER OF THE CHEBYSHEV SERIES APPROXI- 00000200
C                                 MATING THE INTEGRANDS IN THE PICARD-  00000210
C                                 CHEBYSHEV INTEGRATION ALGORITHM (<68) 00000220
C       NSUB         1         2  SUBSCRIPT OF THE FINAL CHEBYSHEV OR-  00000230
C                                 THOGONALITY POINT (<70)               00000240
C     IEQSET         1         3  FLAG INDICATING THE SET OF DIFFERENTIAL00000250
C                                 EQUATIONS TO BE INTEGRATED:           00000260
C                                 1=COWELL EQUATIONS                    00000270
C                                 2=KEPLERIAN VOP EQUATIONS             00000280
C                                 3=EQUINOCTIAL VOP EQUATIONS           00000290
C      ITERM         1         4  TERMINAL ITERATION NUMBER, SPECIFYING 00000300
C                                 THE MAXIMUM NUMBER OF ITERATIONS      00000310
C                                 ALLOWED BEFORE THE ALGORITHM WILL     00000320
C                                 TERMINATE                             00000330
C                                                                       00000340
      DATA MORD   / 36          /                                       00000350
C                                                                       00000360
      DATA NSUB   / 49          /                                       00000370
C                                                                       00000380
      DATA IEQSET / 1           /                                       00000390
C                                                                       00000400
      DATA ITERM  / 99          /                                       00000410
C                                                                       00000420
      END                                                               00000430

******** END OF MEMBER NPCNTL      43 RECORDS ********
```

OBSOUT JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE OBSOUT                                                 00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO OUTPUT THE OBSERVATION PROCESSING REPORT                   00000060
C                                                                       00000070
C     CALLING SEQUENCE                                                  00000080
C         CALL OBSOUT                                                   00000090
C                                                                       00000100
C     CALLING SEQUENCE PARAMETERS                                       00000110
C         NONE                                                          00000120
C                                                                       00000130
C     EXTERNAL REFERENCES                                               00000140
C         COMMON BLOCKS                                                 00000150
C              /OBSDAT/                                                 00000160
C                                                                       00000170
C     ANALYSIS                                                          00000180
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000190
C                                                                       00000200
C     PROGRAMMING                                                       00000210
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000220
C                                                                       00000230
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000240
C                                                                       00000250
      IMPLICIT REAL*8(A-H,O-Z)                                          00000260
C                                                                       00000270
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000280
C                                                                       00000290
      DIMENSION       OBSTYP(6)                                         00000300
C                                                                       00000310
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000320
C                                                                       00000330
      COMMON/OBSDAT/  OBSFIL(100,6)         ,         OBSPAR(100,6)   , 00000340
     1                OBSCMP(100,6)         ,         RES(100)        , 00000350
     2                CMTXMS(100)           ,         NOBS              00000360
C                                                                       00000370
C * * * * * * * * *         DATA STATEMENTS         * * * * * * * * * * 00000380
C                                                                       00000390
      DATA OBSTYP  /  'X'  ,  'Y'  ,  'Z'  ,  'X'  ,  'Y'  ,  'Z'    / 00000400
C                                                                       00000410
C     PRINT OUT OBSERVATION PROCESSING REPORT                           00000420
C                                                                       00000430
      WRITE(6,100)                                                      00000440
      DO  10   I = 1,NOBS                                               00000450
      TOBS   = OBSFIL(I,1)                                              00000460
      IOTYPE = OBSFIL(I,2)                                              00000470
      OO     = OBSFIL(I,6)                                              00000480
      OC     = OBSCMP(I,IOTYPE)                                         00000490
      R      = RES(I)                                                   00000500
      WF     = CMTXMS(I)                                                00000510
      IF (IOTYPE.LE.3)    WRITE(6,110)  TOBS,OBSTYP(IOTYPE),OO,OC,R,WF  00000520
      IF (IOTYPE.GE.4)    WRITE(6,120)  TOBS,OBSTYP(IOTYPE),OO,OC,R,WF  00000530
   10 CONTINUE                                                          00000540
      RETURN                                                            00000550
C                                                                       00000560
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * * 00000570
```

```
OBSOUT                JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000580
  100 FORMAT(/10X,112('_')/15X,'***   OBSERVATION PROCESSING INVOLV',    00000590
     1       'ED IN THIS EVALUATION OF THE COST FUNCTIONAL AND ITS ',   00000600
     2       'GRADIENT   ***'/10X,112('_')//1X,'OBSERVATION',1X,        00000610
     3       'OBSERVATION',11X,'ACTUAL',18X,'COMPUTED'/4X,'TIMES',7X,   00000620
     4       'TYPES',11X,'OBSERVATION',15X,'OBSERVATION',16X,'RESID',   00000630
     5       'UALS',15X,'WEIGHTING FACTORS'/2(1X,11('_')),3(1X,         00000640
     6       25('_')),1X,29('_'))                                       00000650
C                                                                       00000660
  110 FORMAT(1X,F8.2,'MIN  ',A1,'-POSITION ',1PD20.13,'  KM  ',         00000670
     1       1PD20.13,'  KM  ',1PD20.13,'  KM  ',1PD20.13,             00000680
     2       '(1/KM)**2')                                               00000690
C                                                                       00000700
  120 FORMAT(1X,F8.2,'MIN  ',A1,'-VELOCITY ',1PD20.13,' KM/S ',         00000710
     1       1PD20.13,' KM/S ',1PD20.13,' KM/S ',1PD20.13,              00000720
     2       '(S/KM)**2')                                               00000730
C                                                                       00000740
      END                                                               00000750

******** END OF MEMBER OBSOUT      75 RECORDS ********
```

OUTPRD JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE OUTPRD(X,Y,Z,N)                                        00000010
C                                                                       00000020
C     VERSION OF 09/15/79      *** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO COMPUTE THE OUTER (DYADIC) PRODUCT OF TWO VECTORS.          00000060
C                                                                       00000070
C     CALLING SEQUENCE                                                  00000080
C        CALL OUTPRD(X,Y,Z,N)                                           00000090
C                                                                       00000100
C     CALLING SEQUENCE PARAMETERS                                       00000110
C        X,Y         I          VECTORS (UP TO 6X1) FOR WHICH THE      00000120
C                               OUTER PRODUCT (UP TO 6X6) IS TO BE     00000130
C                               COMPUTED.                              00000140
C        N           I          DIMENSION OF THE VECTORS X AND Y.      00000150
C        Z           O          MATRIX (6X6) IN WHICH THE ELEMENTS     00000160
C                               ARE COMPUTED AS FOLLOWS:               00000170
C                                                                       00000180
C                                      Z    = X  * Y                   00000190
C                                       I,J    I    J                  00000200
C                                                                       00000210
C     EXTERNAL REFERENCES                                               00000220
C        NONE                                                           00000230
C                                                                       00000240
C     ANALYSIS                                                          00000250
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000260
C                                                                       00000270
C     PROGRAMMING                                                       00000280
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000290
C                                                                       00000300
      IMPLICIT REAL*8(A-H,O-Z)                                          00000310
      DIMENSION       X(6)       ,  Y(6)        ,  Z(6,6)              00000320
      DO  20   J = 1,N                                                  00000330
      DO  10   I = 1,N                                                  00000340
   10 Z(I,J) = X(I) * Y(J)                                              00000350
   20 CONTINUE                                                          00000360
      RETURN                                                            00000370
      END                                                               00000380
```

******** END OF MEMBER OUTPRD 38 RECORDS ********

PAETAK JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE PAETAK(AK,PAEDAK)                                          00000010
C                                                                           00000020
C     VERSION OF 08/20/79     **** CHECKED OUT ****                         00000030
C                                                                           00000040
C     PURPOSE                                                               00000050
C         TO COMPUTE THE PARTIAL DERIVATIVES OF A SET OF EQUINOCTIAL        00000060
C         ORBIT ELEMENTS WITH RESPECT TO THEIR CORRESPONDING                00000070
C         KEPLERIAN ELEMENTS                                                00000080
C                                                                           00000090
C     CALLING SEQUENCE                                                      00000100
C         CALL PAETAK(AK,PAEDAK)                                            00000110
C                                                                           00000120
C     CALLING SEQUENCE PARAMETERS                                           00000130
C         AK       I   ARRAY (6X1) CONTAINING THE KEPLERIAN ELEMENTS        00000140
C         PAEDAK   O   ARRAY (6X6) CONTAINING THE PARTIAL DERIVATIVES       00000150
C                      OF THE EQUINOCTIAL ORBIT ELEMENTS WITH RESPECT       00000160
C                      TO THE KEPLERIAN ORBIT ELEMENTS                      00000170
C                                                                           00000180
C     EXTERNAL REFERENCES                                                   00000190
C         NONE                                                              00000200
C                                                                           00000210
C     ANALYSIS                                                              00000220
C         CAPTAIN JEFFREY S. SHAVER, USAF                                   00000230
C                                                                           00000240
C     PROGRAMMING                                                           00000250
C         CAPTAIN JEFFREY S. SHAVER, USAF                                   00000260
C                                                                           00000270
C * * * * * * * * *          START PROGRAM          * * * * * * * * * *    00000280
C                                                                           00000290
      IMPLICIT REAL*8(A-H,O-Z)                                              00000300
C                                                                           00000310
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * *    00000320
C                                                                           00000330
      DIMENSION       AK(6)    ,    PAEDAK(6,6)                             00000340
C                                                                           00000350
C     DETERMINE RETROGRADE FACTOR FROM GIVEN INCLINATION                    00000360
C                                                                           00000370
      PI = 3.141592653589793D0                                              00000380
      IF (AK(3).GE.0.D0.AND.AK(3).LE.PI/2.D0)     RETRO =  1.D0             00000390
      IF (AK(3).GT.PI/2.D0.AND.AK(3).LE.PI)       RETRO = -1.D0             00000400
C                                                                           00000410
C     DETERMINE AUXILIARY QUANTITIES                                        00000420
C                                                                           00000430
      SWPIOM = DSIN(AK(4) + RETRO * AK(5))                                  00000440
      CWPIOM = DCOS(AK(4) + RETRO * AK(5))                                  00000450
      SI2    = DSIN(AK(3)/2.D0)                                             00000460
      CI2    = DCOS(AK(3)/2.D0)                                             00000470
      SOM    = DSIN(AK(5))                                                  00000480
      COM    = DCOS(AK(5))                                                  00000490
C                                                                           00000500
C     FORM NONZERO PARTIALS OF SEMIMAJOR AXIS (A) WITH RESPECT TO           00000510
C     KEPLERIAN ELEMENTS                                                    00000520
C                                                                           00000530
      PAEDAK(1,1) = 1.D0                                                    00000540
C                                                                           00000550
C     FORM NONZERO PARTIALS OF H WITH RESPECT TO KEPLERIAN ELEMENTS         00000560
C                                                                           00000570
```

```
      PAEDAK(2,2) = SWPIOM                                                    00000580
      PAEDAK(2,4) = AK(2) * CWPIOM                                            00000590
      PAEDAK(2,5) = AK(2) * CWPIOM * RETRO                                    00000600
C                                                                             00000610
C     FORM NONZERO PARTIALS OF K WITH RESPECT TO KEPLERIAN ELEMENTS           00000620
C                                                                             00000630
      PAEDAK(3,2) =  CWPIOM                                                   00000640
      PAEDAK(3,4) = -AK(2) * SWPIOM                                           00000650
      PAEDAK(3,5) = -AK(2) * SWPIOM * RETRO                                   00000660
C                                                                             00000670
C     FORM NONZERO PARTIALS OF P AND Q WITH RESPECT TO KEPLERIAN              00000680
C     ELEMENTS FOR DIRECT ORBITS:  RETRO =  1                                 00000690
C                                                                             00000700
      IF (RETRO.LT.0.D0)                GO TO 30                              00000710
      PAEDAK(4,3) =  SOM/(2.D0 * CI2 * CI2)                                   00000720
      PAEDAK(5,3) =  COM/(2.D0 * CI2 * CI2)                                   00000730
      PAEDAK(4,5) =  (SI2 * COM)/CI2                                          00000740
      PAEDAK(5,5) = -(SI2 * SOM)/CI2                                          00000750
      GO TO 40                                                                00000760
C                                                                             00000770
C     FORM NONZERO PARTIALS OF P AND Q WITH RESPECT TO KEPLERIAN              00000780
C     ELEMENTS FOR RETROGRADE ORBITS:  RETRO = -1                             00000790
C                                                                             00000800
   30 PAEDAK(4,3) = -SOM/(2.D0 * SI2 * SI2)                                   00000810
      PAEDAK(5,3) = -COM/(2.D0 * SI2 * SI2)                                   00000820
      PAEDAK(4,5) =  (CI2 * COM)/SI2                                          00000830
      PAEDAK(5,5) = -(CI2 * SOM)/SI2                                          00000840
   40 CONTINUE                                                                00000850
C                                                                             00000860
C     FORM NONZERO PARTIALS OF MEAN LONGITUDE (LAMDA) WITH RESPECT TO         00000870
C     KEPLERIAN ELEMENTS                                                      00000880
C                                                                             00000890
      PAEDAK(6,4) = 1.D0                                                      00000900
      PAEDAK(6,5) = RETRO                                                     00000910
      PAEDAK(6,6) = 1.D0                                                      00000920
C                                                                             00000930
      RETURN                                                                  00000940
C                                                                             00000950
      END                                                                     00000960

******** END OF MEMBER PAETAK       96 RECORDS ********
```

PAKTAE JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE PAKTAE(AE,RETRO,PAKDAE)                                00000010
C                                                                       00000020
C     VERSION OF 08/20/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO COMPUTE THE PARTIAL DERIVATIVES OF A SET OF KEPLERIAN      00000060
C         ORBIT ELEMENTS WITH RESPECT TO THEIR CORRESPONDING            00000070
C         EQUINOCTIAL ELEMENTS                                          00000080
C                                                                       00000090
C     CALLING SEQUENCE                                                  00000100
C         CALL PAKTAE(AE,RETRO,PAKDAE)                                  00000110
C                                                                       00000120
C     CALLING SEQUENCE PARAMETERS                                       00000130
C         AE       I   ARRAY (6X1) CONTAINING THE EQUINOCTIAL ELEMENTS  00000140
C         RETRO    I   RETROGRADE FACTOR                                00000150
C         PAKDAE   O   ARRAY (6X6) CONTAINING THE PARTIAL DERIVATIVES   00000160
C                      OF THE KEPLERIAN ORBIT ELEMENTS WITH RESPECT     00000170
C                      TO THE EQUINOCTIAL ORBIT ELEMENTS                00000180
C                                                                       00000190
C     EXTERNAL REFERENCES                                               00000200
C         NONE                                                          00000210
C                                                                       00000220
C     ANALYSIS                                                          00000230
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000240
C                                                                       00000250
C     PROGRAMMING                                                       00000260
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000270
C                                                                       00000280
C * * * * * * * * *          START PROGRAM          * * * * * * * * * *00000290
C                                                                       00000300
      IMPLICIT REAL*8(A-H,O-Z)                                          00000310
C                                                                       00000320
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * *00000330
C                                                                       00000340
      DIMENSION       AE(6)    ,    PAKDAE(6,6)                       00000350
C                                                                       00000360
C     INITIALIZE PARTIAL DERIVATIVE ARRAY TO ZERO; COMPUTE AUXILIARY    00000370
C     QUANTITIES                                                        00000380
C                                                                       00000390
      DO  20   I = 1,6                                                  00000400
      DO  10   J = 1,6                                                  00000410
   10 PAKDAE(I,J) = 0.D0                                                00000420
   20 CONTINUE                                                          00000430
      H2K2 = AE(2)**2 + AE(3)**2                                        00000440
      P2Q2 = AE(4)**2 + AE(5)**2                                        00000450
      PQ   = 1.D0 + P2Q2                                                00000460
C                                                                       00000470
C     FORM NONZERO PARTIALS OF SEMIMAJOR AXIS (A) WITH RESPECT TO       00000480
C     EQUINOCTIAL ELEMENTS                                              00000490
C                                                                       00000500
      PAKDAE(1,1) = 1.D0                                                00000510
C                                                                       00000520
C     FORM NONZERO PARTIALS OF ECCENTRICITY (E) WITH RESPECT TO         00000530
C     EQUINOCTIAL ELEMENTS                                              00000540
C                                                                       00000550
      PAKDAE(2,2) = AE(2)/DSQRT(H2K2)                                   00000560
      PAKDAE(2,3) = AE(3)/DSQRT(H2K2)                                   00000570
```

```
PAKTAE              JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000580
C     FORM NONZERO PARTIALS OF INCLINATION (I) WITH RESPECT TO          00000590
C     EQUINOCTIAL ELEMENTS                                              00000600
C                                                                       00000610
      PAKDAE(3,4) = (2.D0 * RETRO * AE(4))/(PQ * DSQRT(P2Q2))           00000620
      PAKDAE(3,5) = (2.D0 * RETRO * AE(5))/(PQ * DSQRT(P2Q2))           00000630
C                                                                       00000640
C     FORM NONZERO PARTIALS OF THE ARGUMENT OF PERIAPSIS WITH RESPECT   00000650
C     TO EQUINOCTIAL ELEMENTS                                           00000660
C                                                                       00000670
      PAKDAE(4,2) =  AE(3)/H2K2                                         00000680
      PAKDAE(4,3) = -AE(2)/H2K2                                         00000690
      PAKDAE(4,4) = -RETRO * AE(5)/P2Q2                                 00000700
      PAKDAE(4,5) =  RETRO * AE(4)/P2Q2                                 00000710
C                                                                       00000720
C     FORM NONZERO PARTIALS OF THE LONGITUDE OF THE ASCENDING NODE      00000730
C     WITH RESPECT TO EQUINOCTIAL ELEMENTS                              00000740
C                                                                       00000750
      PAKDAE(5,4) = AE(5)/P2Q2                                          00000760
      PAKDAE(5,5) = AE(4)/P2Q2                                          00000770
C                                                                       00000780
C     FORM NONZERO PARTIALS OF MEAN ANOMALY (M) WITH RESPECT TO         00000790
C     EQUINOCTIAL ELEMENTS                                              00000800
C                                                                       00000810
      PAKDAE(6,2) = -AE(3)/H2K2                                         00000820
      PAKDAE(6,3) =  AE(2)/H2K2                                         00000830
      PAKDAE(6,6) =  1.D0                                               00000840
      RETURN                                                            00000850
      END                                                               00000860

******** END OF MEMBER PAKTAE      86 RECORDS ********
```

```
      SUBROUTINE PATA0(EL,TCUR,TEPOK,IJ2RTS,PAWRAO)                     00000010
C                                                                       00000020
C     VERSION OF 12/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO COMPUTE THE MATRIX OF ANALYTIC PARTIAL DERIVATIVES OF THE   00000060
C        CURRENT EQUINOCTIAL ORBITAL ELEMENT VECTOR A WITH RESPECT TO   00000070
C        THE EPOCH EQUINOCTIAL ORBIT ELEMENT VECTOR A .                 00000080
C                                                    0                  00000090
C                                                                       00000100
C     CALLING SEQUENCE                                                  00000110
C        CALL PATA0(EL,TCUR,TEPOK,IJ2RTS,PAWRAO)                        00000120
C                                                                       00000130
C     CALLING SEQUENCE PARAMETERS                                       00000140
C        EL       I   ARRAY (26X1) CONTAINING THE CURRENT EQUINOCTIAL   00000150
C                     ELEMENTS AND ASSOCIATED AUXILIARY QUANTITIES      00000160
C        TCUR     I   CURRENT TIME FOR EVALUATION OF THE PARTIALS       00000170
C        TEPOK    I   EPOCH TIME                                        00000180
C        IJ2RTS   I   FLAG TO INDICATE WHETHER THE SECULAR RATES IN THE 00000190
C                     ARGUMENT OF PERIAPSIS AND THE LONGITUDE OF THE    00000200
C                     ASCENDING NODE DUE TO J2 SHOULD BE INCLUDED IN    00000210
C                     THE COMPUTATION OF THE PARTIAL DERIVATIVES:       00000220
C                     = 0:  DO NOT INCLUDE THE SECULAR RATES            00000230
C                     = 1:  INCLUDE THE RATES IN THE COMPUTATIONS       00000240
C        PAWRAO  .O   ARRAY (6X6) CONTAINING THE PARTIAL DERIVATIVES    00000250
C                     OF THE CURRENT ELEMENT VECTOR WITH RESPECT TO     00000260
C                     THE EPOCH ELEMENT VECTOR                          00000270
C                                                                       00000280
C                                          III     III                  00000290
C                                          I         I                  00000300
C                                          I   ∂Ā    I                  00000310
C                              PAWRAO  =   I _____   I                  00000320
C                                          I         I                  00000330
C                                          I   ∂Ā    I                  00000340
C                                          I    0    I                  00000350
C                                          III     III                  00000360
C                                                                       00000370
C     EXTERNAL REFERENCES                                               00000380
C        NONE                                                           00000390
C                                                                       00000400
C     ANALYSIS                                                          00000410
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000420
C                                                                       00000430
C     PROGRAMMING                                                       00000440
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000450
C                                                                       00000460
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * *00000470
C                                                                       00000480
      IMPLICIT REAL*8(A-H,O-Z)                                          00000490
C                                                                       00000500
C * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * *00000510
C                                                                       00000520
      DIMENSION       PAWRAO(6,6) ,    EL(26)                         00000530
C                                                                       00000540
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * *00000550
C                                                                       00000560
      COMMON/ERTSHP/  REQ     ,   WE       ,   FINV      ,   SPHINF   , 00000570
```

```
     *                GM                                                     00000580
C                                                                           00000590
      COMMON/XEPOCH/  EPST(6) ,   EPAK(26) ,   EPAE(26)                     00000600
C                                                                           00000610
      COMMON/ZNLCOF/  SHJ(8)  ,   IZNLS                                     00000620
C                                                                           00000630
C     INITIALIZE THE PARTIAL DERIVATIVE MATRIX TO ZERO                      00000640
C                                                                           00000650
      DO  20   I = 1,6                                                      00000660
      DO  10   J = 1,6                                                      00000670
   10 PARRAO(I,J) = 0.D0                                                    00000680
   20 CONTINUE                                                              00000690
C                                                                           00000700
C     COMPUTE THE TIME IN SECONDS FROM EPOCH                                00000710
C                                                                           00000720
      DELT   =  TCUR - TEPOK                                                00000730
C                                                                           00000740
C                  2   2                                                    00000750
C     COMPUTE C = P + Q                                                     00000760
C                                                                           00000770
      C = EL(4)**2 + EL(5)**2                                               00000780
C                                                                           00000790
C                              2   2                                        00000800
C     COMPUTE GAMMA = 1/(1- H - K )                                         00000810
C                                                                           00000820
      GAMMA = 1.D0/(1.D0 - EL(2)**2 - EL(3)**2)                             00000830
C                                                                           00000840
C     ESTABLISH VALUE OF THE J  SPHERICAL HARMONIC COEFFICIENT              00000850
C                             2                                             00000860
C                                                                           00000870
      SHJ2 = SHJ(2)                                                         00000880
      IF (IZNLS.LT.2)                  SHJ2 = 0.D0                          00000890
      IF (SHJ2.EQ.0.D0)                GO TO 30                             00000900
C                                                                           00000910
C     BEGIN COMPUTATION OF THE SHORT-PERIODIC COMPONENT OF THE SEMIMAJOR     00000920
C     AXIS DUE TO THE J  ZONAL HARMONIC:                                    00000930
C                     2                                                     00000940
C                                                                           00000950
C                       2                                                   00000960
C     COMPUTE AR2 = (A/R)                                                   00000970
C                                                                           00000980
      AR2 = (EL(1)/EL(14))**2                                               00000990
C                                                                           00001000
C                   2   2                                                   00001010
C     COMPUTE QP = Q  - P                                                   00001020
C                                                                           00001030
      QP = EL(5)**2 - EL(4)**2                                              00001040
C                                                                           00001050
C                                    2   2                                  00001060
C     COMPUTE RTGAM = SQRT(1/(1 - H  - K ))                                 00001070
C                                                                           00001080
      RTGAM = DSQRT(GAMMA)                                                  00001090
C                                                                           00001100
C     COMPUTE PQI = I*P*Q, WHERE I IS THE RETROGRADE FACTOR                 00001110
C                                                                           00001120
      PQI = EL(13) * EL(4) * EL(5)                                          00001130
C                                                                           00001140
```

```
C     COMPUTE PQIH = H*I*P*Q                                                    00001150
C                                                                               00001160
      PQIH = EL(2) * PQI                                                        00001170
C                                                                               00001180
C     COMPUTE PQIK = I*K*P*Q                                                    00001190
C                                                                               00001200
      PQIK = EL(3) * PQI                                                        00001210
C                                                                               00001220
C                                                 2                             00001230
C     COMPUTE THE POLYNOMIAL IN C:  CPOL = C  - 7C+ 1                           00001240
C                                                                               00001250
      CPOL = C**2 - 7.D0*C + 1.D0                                               00001260
C                                                                               00001270
C     COMPUTE THE LEADING TERM WHICH MULTIPLIES ALL HARMONIC TERMS IN           00001280
C     THE SHORT PERIODIC EXPRESSION:                                            00001290
C                                                                               00001300
C                          2   2                        2                      00001310
C     COFMN = (J * (A/R) * R  * GAMMA)/(A * (1+C)  )                            00001320
C                2           EQ                                                00001330
C                                                                               00001340
      COFMN = (SHJ2*AR2*(REQ**2)*GAMMA)/(EL(1)*(1.D0+C)**2)                     00001350
C                                                                               00001360
C     COMPUTE COEFFICIENTS OF THE SIN(3L) AND COS(3L) HARMONICS                 00001370
C                                                                               00001380
      CSN3L = 3.D0*EL(2)*QP + 6.D0*PQIK                                         00001390
      CCS3L = 3.D0*EL(3)*QP - 6.D0*PQIH                                         00001400
C                                                                               00001410
C     COMPUTE COEFFICIENTS OF THE SIN(2L) AND COS(2L) HARMONICS                 00001420
C                                                                               00001430
      CSN2L = 12.D0 * PQI                                                       00001440
      CCS2L = 6.D0 * QP                                                         00001450
C                                                                               00001460
C     COMPUTE COEFFICIENTS OF THE SIN(1L) AND COS(1L) HARMONICS                 00001470
C                                                                               00001480
      CSNL  = 6.D0*PQIK + 6.D0*EL(2)*EL(4)**2 + EL(2)*CPOL                      00001490
      CCSL  = 6.D0*PQIH - 6.D0*EL(3)*EL(4)**2 + EL(3)*(CPOL+6.D0*C)             00001500
C                                                                               00001510
C     COMPUTE CONSTANT TERM IN THE EXPRESSION                                   00001520
C                                                                               00001530
      CZERO = (CPOL + 3.D0*C) * (1.D0 - RTGAM/AR2)                              00001540
C                                                                               00001550
C     COMPUTE THE SHORT PERIODIC PART OF THE SEMIMAJOR AXIS                     00001560
C                                                                               00001570
      DELA = COFMN * (CSN3L*DSIN(3.D0*EL(24)) + CCS3L*DCOS(3.D0*EL(24))         00001580
     1              + CSN2L*DSIN(2.D0*EL(24)) + CCS2L*DCOS(2.D0*EL(24))        00001590
     2              + CSNL*DSIN(EL(24)) + CCSL*DCOS(EL(24)) + CZERO  )         00001600
C                                                                               00001610
C     COMPUTE THE MEAN SEMIMAJOR AXIS, THE MEAN MEAN MOTION AND THE             00001620
C     MEAN SEMI-LATUS RECTUM                                                    00001630
C                                                                               00001640
   30 IF (SHJ2.EQ.0.D0)                  DELA = 0.D0                          00001650
      AMEAN  = EL(1) - DELA                                                     00001660
      XNMEAN = DSQRT(GM/AMEAN**3)                                               00001670
      SLRMN  = AMEAN/GAMMA                                                      00001680
C                                                                               00001690
C     COMPUTE THE SECULAR RATES IN THE ARGUMENT OF PERIAPSIS AND THE            00001700
C     LONGITUDE OF THE ASCENDING NODE DUE TO THE J  ZONAL HARMONIC:             00001710
```

```
C                                                    2                    00001720
C                                                                         00001730
      IF (IJ2RTS.EQ.0)                    SHJ2 = 0.D0                     00001740
      XK     =  1.5D0 * SHJ2 * REQ**2 * DSQRT(GM)                         00001750
      COF12  =  XK * DSQRT(1.D0/AMEAN**7) * GAMMA**2                      00001760
      COF3   =  COF12/RTGAM                                               00001770
C                                                                         00001780
C     COMPUTE POLYNOMIALS IN C = P**2 + Q**2                              00001790
C                                                                         00001800
      CPOL1  =  (1.D0 - 6.D0*C + 3.D0*C**2)/(1.D0 + C)**2                 00001810
      CPOL2  =  (EL(13) * (1.D0 - C))/(1.D0 + C)                          00001820
      CPOL3  =  (1.D0 - 4.D0*C + C**2)/(1.D0 + C)**2                      00001830
      CPOL4  =  (3.D0*C - 2.D0)/(1.D0 + C)**3                             00001840
      CPOL5  =  EL(13)/(1.D0 + C)**2                                      00001850
      CPOL6  =  (C - 1.D0)/(1.D0 + C)**3                                  00001860
C                                                                         00001870
C     COMPUTE THE DELTA FACTORS                                           00001880
C                                                                         00001890
      DELTA1 =  COF12 * CPOL1                                             00001900
      DELTA2 = -COF12 * CPOL2                                             00001910
      DELTA3 =  COF3  * CPOL3                                             00001920
C                                                                         00001930
C                 ∂A      ∂A      ∂A      ∂A      ∂A      ∂A              00001940
C     COMPUTE     ---  ;  ---  =  ---  =  ---  =  ---  =  -----  = 0      00001950
C                 ∂A      ∂H      ∂K      ∂P      ∂Q      ∂LAMDA          00001960
C                   0       0       0       0       0       0             00001970
C                                                                         00001980
      PAWRAO(1,1) =  1.D0                                                 00001990
C                                                                         00002000
C                 ∂H      ∂H      ∂H      ∂H      ∂H      ∂H              00002010
C     COMPUTE     ---  ;  ---  ;  ---  ;  ---  ;  ---  ;  -----          00002020
C                 ∂A      ∂H      ∂K      ∂P      ∂Q      ∂LAMDA          00002030
C                   0       0       0       0       0       0             00002040
C                                                                         00002050
      PAWRAO(2,1) = -3.5D0*(EL(3)/AMEAN)*DELTA1*DELT                      00002060
      PAWRAO(2,2) =  DCOS(DELTA1*DELT) +                                  00002070
     *                        4.D0*EL(3)*EPAE(2)*GAMMA*DELTA1*DELT       00002080
      PAWRAO(2,3) =  DSIN(DELTA1*DELT) +                                  00002090
     *                        4.D0*EL(3)*EPAE(3)*GAMMA*DELTA1*DELT       00002100
      PAWRAO(2,4) =  8.D0*EL(3)*EPAE(4)*COF12*CPOL4*DELT                  00002110
      PAWRAO(2,5) =  8.D0*EL(3)*EPAE(5)*COF12*CPOL4*DELT                  00002120
C                                                                         00002130
C                 ∂K      ∂K      ∂K      ∂K      ∂K      ∂K              00002140
C     COMPUTE     ---  ;  ---  ;  ---  ;  ---  ;  ---  ;  -----          00002150
C                 ∂A      ∂H      ∂K      ∂P      ∂Q      ∂LAMDA          00002160
C                   0       0       0       0       0       0             00002170
C                                                                         00002180
      PAWRAO(3,1) =  3.5D0*(EL(2)/AMEAN)*DELTA1*DELT                      00002190
      PAWRAO(3,2) = -DSIN(DELTA1*DELT) -                                  00002200
     *                        4.D0*EL(2)*EPAE(2)*GAMMA*DELTA1*DELT       00002210
      PAWRAO(3,3) =  DCOS(DELTA1*DELT) -                                  00002220
     *                        4.D0*EL(2)*EPAE(3)*GAMMA*DELTA1*DELT       00002230
      PAWRAO(3,4) = -8.D0*EL(2)*EPAE(4)*COF12*CPOL4*DELT                  00002240
      PAWRAO(3,5) = -8.D0*EL(2)*EPAE(5)*COF12*CPOL4*DELT                  00002250
C                                                                         00002260
C                 ∂P      ∂P      ∂P      ∂P      ∂P      ∂P              00002270
C     COMPUTE     ---  ;  ---  ;  ---  ;  ---  ;  ---  ;  -----          00002280
```

```
C                ∂A       ∂H       ∂K       ∂P       ∂Q       ∂LAMDA          00002290
C                  0        0        0        0        0            0          00002300
C                                                                              00002310
      PAWRAO(4,1) =  3.5D0*(EL(5)/AMEAN)*DELTA2*DELT                           00002320
      PAWRAO(4,2) =  4.D0*EL(5)*EPAE(2)*GAMMA*DELTA2*DELT                      00002330
      PAWRAO(4,3) =  4.D0*EL(5)*EPAE(3)*GAMMA*DELTA2*DELT                      00002340
      PAWRAO(4,4) =  DCOS(DELTA2*DELT) +                                       00002350
     *                          4.D0*EL(5)*EPAE(4)*COF12*CPOL5*DELT           00002360
      PAWRAO(4,5) =  DSIN(DELTA2*DELT) +                                       00002370
     *                          4.D0*EL(5)*EPAE(5)*COF12*CPOL5*DELT           00002380
C                                                                              00002390
C                ∂Q       ∂Q       ∂Q       ∂Q       ∂Q       ∂Q             00002400
C     COMPUTE   ____ ;  ____ ;  ____ ;  ____ ;  ____ ;  _____                 00002410
C                ∂A       ∂H       ∂K       ∂P       ∂Q       ∂LAMDA         00002420
C                  0        0        0        0        0            0          00002430
C                                                                              00002440
      PAWRAO(5,1) = -3.5D0*(EL(4)/AMEAN)*DELTA2*DELT                           00002450
      PAWRAO(5,2) = -4.D0*EL(4)*EPAE(2)*GAMMA*DELTA2*DELT                      00002460
      PAWRAO(5,3) = -4.D0*EL(4)*EPAE(3)*GAMMA*DELTA2*DELT                      00002470
      PAWRAO(5,4) = -DSIN(DELTA2*DELT) -                                       00002480
     *                          4.D0*EL(4)*EPAE(4)*COF12*CPOL5*DELT           00002490
      PAWRAO(5,5) =  DCOS(DELTA2*DELT) -                                       00002500
     *                          4.D0*EL(4)*EPAE(5)*COF12*CPOL5*DELT           00002510
C                                                                              00002520
C                ∂ML      ∂ML      ∂ML      ∂ML      ∂ML      ∂ML             00002530
C     COMPUTE   ____ ;  ____ ;  ____ ;  ____ ;  ____ ;  _____                 00002540
C                ∂A       ∂H       ∂K       ∂P       ∂Q       ∂LAMDA         00002550
C                  0        0        0        0        0            0          00002560
C                                                                              00002570
      PAWRAO(6,1) = -(1.5D0*XNMEAN + 3.5D0*(DELTA1+DELTA3))*DELT/AMEAN         00002580
      PAWRAO(6,2) =  2.D0*EPAE(2)*GAMMA*(2.D0*DELTA1+1.5D0*DELTA3)*DELT        00002590
      PAWRAO(6,3) =  2.D0*EPAE(3)*GAMMA*(2.D0*DELTA1+1.5D0*DELTA3)*DELT        00002600
      PAWRAO(6,4) =  (8.D0*EPAE(4)*COF12*CPOL4 +                               00002610
     *                           12.D0*EPAE(4)*COF3*CPOL6)*DELT                00002620
      PAWRAO(6,5) =  (8.D0*EPAE(5)*COF12*CPOL4 +                               00002630
     *                           12.D0*EPAE(5)*COF3*CPOL6)*DELT                00002640
      PAWRAO(6,6) =  1.D0                                                      00002650
C                                                                              00002660
      RETURN                                                                   00002670
C                                                                              00002680
      END                                                                      00002690
```

******** END OF MEMBER PATA0 269 RECORDS ********

```
      SUBROUTINE PEQTX(ST,EL,IPARRV)                                    00000010
C                                                                       00000020
C     VERSION OF 08/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO COMPUTE THE PARTIAL DERIVATIVE OF THE VECTOR OF EQUINOC-    00000060
C        TIAL ORBITAL ELEMENTS A WITH RESPECT TO THE STATE VECTOR X.     00000070
C                                                                       00000080
C     METHOD                                                            00000090
C        THE GENERAL METHOD FOR THE EQUINOCTIAL PARTIALS WITH RESPECT   00000100
C        TO THE STATE IS DESCRIBED IN THE FOLLOWING PAPER:              00000110
C                                                                       00000120
C     (1) BROUCKE, R. A., AND P. J. CEFOLA, "ON THE EQUINOCTIAL ORBIT   00000130
C        ELEMENTS," CELESTIAL MECHANICS 5(1972), PP. 303-310.           00000140
C                                                                       00000150
C        MUCH SIMPLIFIED FORMULAE FOR SOME OF THE DERIVATIVES ARE FOUND 00000160
C        IN THE FOLLOWING PAPER:                                        00000170
C                                                                       00000180
C     (2) "A RECURSIVELY FORMULATED FIRST-ORDER SEMIANALYTIC ARTIFICIAL 00000190
C        SATELLITE THEORY BASED ON THE GENERALIZED METHOD OF AVERAGING. 00000200
C        VOLUME 1:  THE GENERALIZED METHOD OF AVERAGING APPLIED TO THE  00000210
C        ARTIFICIAL SATELLITE PROBLEM," CSC/TR-77/6010, NOVEMBER 1977.  00000220
C                                                                       00000230
C        OTHER FORMS OF THESE FORMULAE CAN BE FOUND IN THE FOLLOWING    00000240
C        REFERENCE:                                                     00000250
C                                                                       00000260
C     (3) "ON/OFF-SITE SYSTEM ANALYSIS AND COMPUTER PROGRAMMING SER-    00000270
C        VICES:  THE LONG-TERM PREDICTION OF ARTIFICIAL SATELLITE       00000280
C        ORBITS (REVISED)," CSC/9101-14300-01TR, SEPTEMBER 1973.        00000290
C                                                                       00000300
C        SIMILAR SIMPLIFIED FORMULAE FOR THE REMAINING PARTIALS NOT     00000310
C        DERIVED IN THE ABOVE REFERENCES HAVE BEEN DERIVED BY THE       00000320
C        ANALYST IN THE FORMAT AND NOTATION OF THE ABOVE REFERENCE (2). 00000330
C                                                                       00000340
C     CALLING SEQUENCE                                                  00000350
C        CALL PEQTX(ST,EL,IPARRV)                                       00000360
C                                                                       00000370
C     CALLING SEQUENCE PARAMENTERS                                      00000380
C        ST         I     ARRAY (6X1) CONTAINING THE THREE COMPONENTS OF00000390
C                         POSITION AND THE THREE COMPONENTS OF VELOCITY 00000400
C                         FOR WHICH THE PARTIAL DERIVATIVES ARE TO BE   00000410
C                         COMPUTED.                                     00000420
C        EL         I     ARRAY (26X1) CONTAINING THE SIX EQUINOCTIAL   00000430
C                         ORBIT ELEMENTS AND CERTAIN AUXILIARY QUANTI-  00000440
C                         TIES (SEE SUBROUTINE 'ELMRPT' FOR DESCRIPTION 00000450
C                         OF THESE QUANTITIES).                         00000460
C        IPARRV     I     FLAG TO INDICATE WHICH PARTIALS ARE TO BE     00000470
C                         COMPUTED:                                     00000480
C                         = 0:  COMPUTE PARTIALS WITH RESPECT TO BOTH   00000490
C                               POSITION AND VELOCITY                   00000500
C                         = 1:  COMPUTE PARTIALS WITH RESPECT TO        00000510
C                               POSITION ONLY                           00000520
C                         = 2:  COMPUTE PARTIALS WITH RESPECT TO        00000530
C                               VELOCITY ONLY                           00000540
C                                                                       00000550
C     EXTERNAL REFERENCES                                               00000560
C        NONE                                                           00000570
```

```
C                                                                       00000580
C     ANALYSIS                                                          00000590
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000600
C                                                                       00000610
C     PROGRAMMING                                                       00000620
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000630
C                                                                       00000640
C * * * * * * * * *          START PROGRAM         * * * * * * * * * * * 00000650
C                                                                       00000660
      IMPLICIT REAL*8(A-H,O-Z)                                          00000670
C                                                                       00000680
C * * * * * * * * *      DIMENSION STATEMENTS      * * * * * * * * * * * 00000690
C                                                                       00000700
      DIMENSION       ST(6)      ,  EL(26)     ,  F(3)      ,  G(3)    ,00000710
     *                W(3)                                             00000720
C                                                                       00000730
C * * * * * * * * *          COMMON BLOCKS         * * * * * * * * * * * 00000740
C                                                                       00000750
      COMMON/EQPARM/  FGW(3,3)   ,  YAUX(3)    ,  PQ        ,  BETA    ,00000760
     *                XX         ,  YY         ,  XXD       ,  YYD     ,00000770
     *                AHK        ,  GG                                 00000780
C                                                                       00000790
      COMMON/EQPTLX/  PEQDR(6,3) ,  PEQDV(6,3) ,  PEQDX(6,6)           00000800
C                                                                       00000810
C * * * * * * * * *    EQUIVALENCE STATEMENTS      * * * * * * * * * * * 00000820
C                                                                       00000830
C     FROM COMMON/EQPARM/:                                              00000840
      EQUIVALENCE     (FGW(1,1)  ,  F(1)      ),                       00000850
     *                (FGW(1,2)  ,  G(1)      ),                       00000860
     *                (FGW(1,3)  ,  W(1)      )                        00000870
C                                                                       00000880
C     IF IT IS DESIRED TO COMPUTE THE PARTIAL DERIVATIVES OF THE        00000890
C     EQUINOCTIAL ORBITAL ELEMENTS WITH RESPECT TO THE POSITION VECTOR, 00000900
C     FIRST COMPUTE THE FOLLOWING AUXILIARY QUANTITIES:                 00000910
C                                                                       00000920
C       BETA*SQRT(1-H**2-K**2)*(A/R**3), A/SQRT(1-H**2-K**2)*R**3, A/G, 00000930
C       SQRT(1-H**2-K**2)/(NA**2), AND                                  00000940
C              .     .     .     .     .                                00000950
C            (QIY - PX), ∂X/∂H, ∂X/∂K, ∂Y/∂H, ∂Y/∂K                     00000960
C                                                                       00000970
C                                                                       00000980
      IF (IPARRV.EQ.2)                  GO TO 20                        00000990
C                                                                       00001000
      RFAC  = BETA * AHK/EL(14)**3                                      00001010
C                                                                       00001020
      AXR3  = EL(1)**2/(AHK * EL(14)**3)                                00001030
C                                                                       00001040
      AOG   = EL(1)/GG                                                  00001050
C                                                                       00001060
      HKNA2 = AHK/(EL(7) * EL(1)**3)                                    00001070
C                                                                       00001080
      WFAC1 = EL(5) * EL(13) * YYD - EL(4) * XXD                        00001090
C                                                                       00001100
      PXDWRH =  AOG*YYD**2 + AXR3*(GG*EL(3)*BETA*XX - EL(7)*EL(1)*YY**2)00001110
C                                                                       00001120
      PXDWRK =  AOG*XXD*YYD - AXR3*(GG*EL(2)*BETA*XX +EL(7)*EL(1)*XX*YY)00001130
C                                                                       00001140
```

```
      PYDWRH = -AOG*XXD*YYD + AXR3*(GG*EL(3)*BETA*YY +EL(7)*EL(1)*XX*YY)00001150
C                                                                       00001160
      PYDWRK = -AOG*XXD**2 - AXR3*(GG*EL(2)*BETA*YY - EL(7)*EL(1)*XX**2)00001170
C                                                                       00001180
C     COMPUTE THE PARTIAL DERIVATIVES OF THE EQUINOCTIAL ORBITAL        00001190
C     ELEMENTS WITH RESPECT TO THE POSITION VECTOR                      00001200
C                                                                       00001210
      DO  10   JCOL = 1,3                                               00001220
C                                                                       00001230
      PEQDR(1,JCOL) =  2.D0 * (EL(1)**2/EL(14)**3) * ST(JCOL)           00001240
C                                                                       00001250
      PEQDR(2,JCOL) = -EL(2)*RFAC*ST(JCOL) - (EL(3)*WFAC1/GG) * W(JCOL) 00001260
     *               - HKNA2*PXDWRK*F(JCOL) - HKNA2*PYDWRK*G(JCOL)      00001270
C                                                                       00001280
      PEQDR(3,JCOL) = -EL(3)*RFAC*ST(JCOL) + (EL(2)*WFAC1/GG) * W(JCOL) 00001290
     *               + HKNA2*PXDWRH*F(JCOL) + HKNA2*PYDWRH*G(JCOL)      00001300
C                                                                       00001310
      PEQDR(4,JCOL) = -(PQ * YYD * W(JCOL))/(2.D0 * GG)                 00001320
C                                                                       00001330
      PEQDR(5,JCOL) = -(PQ * XXD * W(JCOL))/(2.D0 * GG)                 00001340
C                                                                       00001350
      PEQDR(6,JCOL) = -ST(JCOL+3)/(EL(7)*EL(1)**2) - (WFAC1*W(JCOL))/GG 00001360
     *  -BETA*HKNA2*((EL(2)*PXDWRH+EL(3)*PXDWRK)*F(JCOL)+(EL(2)*PYDWRH  00001370
     *  +EL(3)*PYDWRK)*G(JCOL))                                         00001380
C                                                                       00001390
   10 CONTINUE                                                          00001400
C                                                                       00001410
C     IF IT IS DESIRED TO COMPUTE THE PARTIAL DERIVATIVES OF THE        00001420
C     EQUINOCTIAL ORBITAL ELEMENTS WITH RESPECT TO VELOCITY, FIRST      00001430
C     COMPUTE THE FOLLOWING AUXILIARY QUANTITIES:                       00001440
C                                                                       00001450
C          SQRT(1-H**2-K**2)/(NA**2), MU, (QIY - PX), AND               00001460
C                   ∂X/∂H, ∂X/∂K, ∂Y/∂H, ∂Y/∂K                          00001470
C                                                                       00001480
   20 IF (IPARRV.EQ.1)                    RETURN                        00001490
C                                                                       00001500
      HKNA2 = AHK/(EL(7) * EL(1)**3)                                    00001510
C                                                                       00001520
      GMC   = EL(7)**2 * EL(1)**3                                       00001530
C                                                                       00001540
      WFAC2 = EL(5) * EL(13) * YY - EL(4) * XX                          00001550
C                                                                       00001560
      PXWRTH = -(EL(3)*BETA*XXD)/EL(7) + (EL(1)/GG) * YY * YYD          00001570
C                                                                       00001580
      PXWRTK =  (EL(2)*BETA*XXD)/EL(7) + (EL(1)/GG)* (XXD*YY-GG)        00001590
C                                                                       00001600
      PYWRTH = -(EL(3)*BETA*YYD)/EL(7) - (EL(1)/GG) * (XX*YYD+GG)       00001610
C                                                                       00001620
      PYWRTK =  (EL(2)*BETA*YYD)/EL(7) - (EL(1)/GG) * XX * XXD          00001630
C                                                                       00001640
C     COMPUTE THE PARTIAL DERIVATIVES OF THE EQUINOCTIAL ORBITAL        00001650
C     ELEMENTS WITH RESPECT TO THE VELOCITY VECTOR                      00001660
C                                                                       00001670
      DO  30   JCOL = 1,3                                               00001680
C                                                                       00001690
      PEQDV(1,JCOL) =  (2.D0 * ST(JCOL+3))/(EL(1) * EL(7)**2)           00001700
C                                                                       00001710
```

```
      PEQDV(2,JCOL) = -(GG*F(JCOL)+EL(14)*XXD*YAUX(JCOL))/GMC           00001720
     *                 + (EL(3)*WFAC2*W(JCOL))/GG                       00001730
C                                                                       00001740
      PEQDV(3,JCOL) =  (GG*G(JCOL)+EL(14)*YYD*YAUX(JCOL))/GMC           00001750
     *                 - (EL(2)*WFAC2*W(JCOL))/GG                       00001760
C                                                                       00001770
      PEQDV(4,JCOL) =  (PQ * YY * W(JCOL))/(2.D0 * GG)                  00001780
C                                                                       00001790
      PEQDV(5,JCOL) =  (PQ * XX * W(JCOL))/(2.D0 * GG)                  00001800
C                                                                       00001810
      PEQDV(6,JCOL) = -(2.D0*ST(JCOL))/(EL(7)*EL(1)**2) +               00001820
     *  (WFAC2*W(JCOL))/GG + BETA*HKNA2*((EL(2)*PXWRTH+EL(3)*PXWRTK)*   00001830
     *  F(JCOL)+(EL(2)*PYWRTH+EL(3)*PYWRTK)*G(JCOL))                    00001840
C                                                                       00001850
   30 CONTINUE                                                          00001860
C                                                                       00001870
C                                                                       00001880
C     CONSTRUCT THE FULL MATRIX OF PARTIAL DERIVATIVES OF THE           00001890
C     EQUINOCTIAL ORBITAL ELEMENTS WITH RESPECT TO THE STATE VECTOR     00001900
C                                                                       00001910
C                                                                       00001920
C                                                                       00001930
C                          III                 III                    00001940
C                          I                I  I                    I  00001950
C                  ∂Ā      I   ∂Ā       I   I  ∂Ā      I             I  00001960
C                 ----  =  I  ----      I   I  ----     I             I  00001970
C                          I            I   I  .        I             I  00001980
C                  ∂X̄      I            I   I           I             I  00001990
C                          I   ∂R̄       I   I  ∂R̄      I             I  00002000
C                          III                 III                    00002010
C                                                                       00002020
C                                                                       00002030
      DO  50   IROW = 1,6                                               00002040
      DO  40   JCOL = 1,3                                               00002050
      PEQDX(IROW,JCOL)   = PEQDR(IROW,JCOL)                             00002060
   40 PEQDX(IROW,JCOL+3) = PEQDV(IROW,JCOL)                             00002070
   50 CONTINUE                                                          00002080
      RETURN                                                            00002090
      END                                                               00002100
```

******** END OF MEMBER PEQTX 210 RECORDS ********

PTLOUT JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE PTLOUT                                                 00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO OUTPUT THE OBSERVATION PARTIAL DERIVATIVE REPORT            00000060
C                                                                       00000070
C     CALLING SEQUENCE                                                  00000080
C        CALL PTLOUT                                                    00000090
C                                                                       00000100
C     CALLING SEQUENCE PARAMETERS                                       00000110
C        NONE                                                           00000120
C                                                                       00000130
C     EXTERNAL REFERENCES                                               00000140
C        SUBROUTINES                                                    00000150
C             NONE                                                      00000160
C                                                                       00000170
C        COMMON BLOCKS                                                  00000180
C             /ICNTRL/                                                  00000190
C             /OBSOUT/                                                  00000200
C                                                                       00000210
C     ANALYSIS                                                          00000220
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000230
C                                                                       00000240
C     PROGRAMMING                                                       00000250
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000260
C                                                                       00000270
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000280
C                                                                       00000290
      IMPLICIT REAL*8(A-H,O-Z)                                          00000300
C                                                                       00000310
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00000320
C                                                                       00000330
      COMMON/ICNTRL/  IRNTYP      ,  IVINIT          ,  IVINTG        , 00000340
     *                ISLVFR                                           00000350
C                                                                       00000360
      COMMON/OBSDAT/  OBSFIL(100,6)          ,      OBSPAR(100,6)     , 00000370
     *                OBSCMP(100,6)          ,      RES(100)          , 00000380
     *                CMTXMS(100)            ,      NOBS              00000390
C                                                                       00000400
C     PRINT OBSERVATION PARTIAL DERIVATIVE REPORT                       00000410
C                                                                       00000420
      WRITE(6,100)                                                      00000430
      IF (ISLVFR.EQ.1)                 WRITE(6,110)                     00000440
      IF (ISLVFR.GT.1)                 WRITE(6,120)                     00000450
      DO  10   I = 1,NOBS                                               00000460
      ITYP = OBSFIL(I,2)                                                00000470
      IDUM = ITYP - 3                                                   00000480
      IF (ITYP.LE.3)          WRITE(6,130) ITYP,(OBSPAR(I,J),J=1,6)     00000490
      IF (ITYP.GE.4)          WRITE(6,140) IDUM,(OBSPAR(I,J),J=1,6)     00000500
   10 CONTINUE                                                          00000510
      RETURN                                                            00000520
C                                                                       00000530
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * * 00000540
C                                                                       00000550
  100 FORMAT(/10X,112('_')/15X,'***   OBSERVATION PARTIAL DERIVATIVES', 00000560
     *       ' INVOLVED IN THIS EVALUATION OF THE COST FUNCTIONAL ',    00000570
```

```
PTLOUT              JSS1493.PRLLEL.ORBDET.FORT

     *       'GRADIENT  ***'/10X,112('_'))                                  00000580
C                                                                           00000590
  110 FORMAT(/19X,'/@ R0(1)',12X,'/@ R0(2)',12X,'/@ R0(3)',11X,             00000600
     *       '/@ RDOT0(1)',9X,'/@ RDOT0(2)',9X,'/@ RDOT0(3)'/12X,          00000610
     *       6(3X,17('_')))                                                 00000620
C                                                                           00000630
  120 FORMAT(/19X,'/@ EL0(1)',11X,'/@ EL0(2)',11X,'/@ EL0(3)',11X,          00000640
     *       '/@ EL0(4)',11X,'/@ EL0(5)',11X,'/@ EL0(6)'/12X,              00000650
     *       6(3X,17('_')))                                                 00000660
C                                                                           00000670
  130 FORMAT(5X,'@ R(',I1,')/',6(1PD20.10))                                 00000680
C                                                                           00000690
  140 FORMAT(2X,'@ RDOT(',I1,')/',6(1PD20.10))                              00000700
C                                                                           00000710
      END                                                                   00000720

******** END OF MEMBER PTLOUT      72 RECCRDS ********
```

```
      SUBROUTINE PXTEQ(ST,EL,IRVPAR)                                    00000010
C                                                                       00000020
C     VERSION OF 03/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO COMPUTE THE PARTIAL DERIVATIVE OF THE STATE VECTOR X WITH   00000060
C        RESPECT TO THE VECTOR OF EQUINOCTIAL ORBITAL ELEMENTS A.       00000070
C                                                                       00000080
C     METHOD                                                            00000090
C        THE GENERAL METHOD FOR THE EQUINOCTIAL PARTIALS IS DESCRIBED   00000100
C        IN THE FOLLOWING PAPER:                                        00000110
C                                                                       00000120
C     (1) BROUCKE, R. A., AND P. J. CEFOLA, "ON THE EQUINOCTIAL ORBIT   00000130
C        ELEMENTS," CELESTIAL MECHANICS 5(1972), PP. 303-310.           00000140
C                                                                       00000150
C        MUCH SIMPLIFIED FORMULAE FOR SOME OF THE DERIVATIVES ARE FOUND 00000160
C        IN THE FOLLOWING PAPER:                                        00000170
C                                                                       00000180
C     (2) "A RECURSIVELY FORMULATED FIRST-ORDER SEMIANALYTIC ARTIFICIAL  00000190
C        SATELLITE THEORY BASED ON THE GENERALIZED METHOD OF AVERAGING. 00000200
C        VOLUME 1:  THE GENERALIZED METHOD OF AVERAGING APPLIED TO THE  00000210
C        ARTIFICIAL SATELLITE PROBLEM," CSC/TR-77/6010, NOVEMBER 1977.  00000220
C                                                                       00000230
C        SIMILAR SIMPLIFIED FORMULAE FOR THE REMAINING PARTIALS NOT     00000240
C        DERIVED IN THE ABOVE REFERENCE HAVE BEEN DERIVED BY THE        00000250
C        ANALYST IN THE FORMAT AND NOTATION OF THE ABOVE REFERENCE (2). 00000260
C                                                                       00000270
C     CALLING SEQUENCE                                                  00000280
C        CALL PXTEQ(ST,EL,IRVPAR)                                       00000290
C                                                                       00000300
C     CALLING SEQUENCE PARAMENTERS                                      00000310
C        ST         I     ARRAY (6X1) CONTAINING THE THREE COMPONENTS OF00000320
C                         POSITION AND THE THREE COMPONENTS OF VELOCITY 00000330
C                         FOR WHICH THE PARTIAL DERIVATIVES ARE TO BE   00000340
C                         COMPUTED.                                     00000350
C        EL         I     ARRAY (26X1) CONTAINING THE SIX EQUINOCTIAL   00000360
C                         ORBIT ELEMENTS AND CERTAIN AUXILIARY QUANTI-  00000370
C                         TIES (SEE SUBROUTINE 'ELMRPT' FOR DESCRIPTION 00000380
C                         OF THESE QUANTITIES).                         00000390
C        IRVPAR     I     FLAG TO INDICATE WHICH PARTIALS ARE TO BE     00000400
C                         COMPUTED:                                     00000410
C                         = 0:  COMPUTE BOTH POSITION AND VELOCITY      00000420
C                               PARTIALS                                00000430
C                         = 1:  COMPUTE ONLY POSITION PARTIALS          00000440
C                         = 2:  COMPUTE ONLY VELOCITY PARTIALS          00000450
C                                                                       00000460
C     EXTERNAL REFERENCES                                               00000470
C        NONE                                                           00000480
C                                                                       00000490
C     ANALYSIS                                                          00000500
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000510
C                                                                       00000520
C     PROGRAMMING                                                       00000530
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000540
C                                                                       00000550
C * * * * * * * * *          START PROGRAM        * * * * * * * * * * * 00000560
C                                                                       00000570
```

PXTEQ JSS1493.PRLLEL.ORBDET.FORT

```
      IMPLICIT REAL*8(A-H,O-Z)                                          00000580
C                                                                       00000590
C * * * * * * * * * *      DIMENSION STATEMENTS      * * * * * * * * * * 00000600
C                                                                       00000610
      DIMENSION       ST(6)      ,  EL(26)     ,  F(3)       ,  G(3)    ,00000620
     *                W(3)                                              00000630
C                                                                       00000640
C * * * * * * * * * *           COMMON BLOCKS          * * * * * * * * * * 00000650
C                                                                       00000660
      COMMON/EQPARM/  FGW(3,3)   ,  YAUX(3)    ,  PQ         ,  BETA    ,00000670
     *                XX         ,  YY         ,  XXD        ,  YYD     ,00000680
     *                AHK        ,  GG                                  00000690
C                                                                       00000700
      COMMON/XPTLEQ/  FRDEQ(3,6) ,  PVDEQ(3,6) ,  PXDEQ(6,6)            00000710
C                                                                       00000720
C * * * * * * * * * *    EQUIVALENCE STATEMENTS      * * * * * * * * * * 00000730
C                                                                       00000740
C     FROM COMMON/EQPARM/:                                              00000750
      EQUIVALENCE     (FGW(1,1)  ,  F(1)     ),                         00000760
     *                (FGW(1,2)  ,  G(1)     ),                         00000770
     *                (FGW(1,3)  ,  W(1)     )                          00000780
C                                                                       00000785
C     IF IT IS DESIRED TO COMPUTE THE PARTIAL DERIVATIVES OF THE        00000790
C     POSITION VECTOR WITH RESPECT TO THE EQUINOCTIAL ELEMENTS, FIRST   00000795
C     COMPUTE THE AUXILIARY QUANTITIES:  ∂X/∂H, ∂X/∂K, ∂Y/∂H, ∂Y/∂K     00000800
C                                                                       00000810
      IF (IRVPAR.EQ.2)                  GO TO 20                        00000815
C                                                                       00000816
      PXWRTH = -(EL(3)*BETA*XXD)/EL(7) + (EL(1)/GG) * YY * YYD          00000820
C                                                                       00000830
      PXWRTK =  (EL(2)*BETA*XXD)/EL(7) + (EL(1)/GG)* (XXD*YY-GG)        00000840
C                                                                       00000850
      PYWRTH = -(EL(3)*BETA*YYD)/EL(7) - (EL(1)/GG) * (XX*YYD+GG)       00000860
C                                                                       00000870
      PYWRTK =  (EL(2)*BETA*YYD)/EL(7) - (EL(1)/GG) * XX * XXD          00000880
C                                                                       00000890
C     COMPUTE THE PARTIAL DERIVATIVES OF THE POSITION VECTOR WITH       00000900
C     RESPECT TO THE EQUINOCTIAL ORBITAL ELEMENTS                       00000910
C                                                                       00000920
      DO  10   IROW = 1,3                                               00000940
C                                                                       00000945
      PRDEQ(IROW,1) = ST(IROW)/EL(1)                                    00000950
C                                                                       00000960
      PRDEQ(IROW,2) = PXWRTH * F(IROW) + PYWRTH * G(IROW)               00000970
C                                                                       00000980
      PRDEQ(IROW,3) = PXWRTK * F(IROW) + PYWRTK * G(IROW)               00000990
C                                                                       00001000
      PRDEQ(IROW,4) = (2.D0/PQ) * (EL(5)*YY*EL(13)*F(IROW)              00001010
     *                -EL(5)*XX*EL(13)*G(IROW) - XX*W(IROW))            00001020
C                                                                       00001030
      PRDEQ(IROW,5) = (2.D0*EL(13)/PQ) * (EL(4)*XX*G(IROW)              00001040
     *                -EL(4)*YY*F(IROW) + YY*W(IROW))                   00001050
C                                                                       00001060
      PRDEQ(IROW,6) = ST(IROW+3)/EL(7)                                  00001070
C                                                                       00001075
   10 CONTINUE                                                          00001080
C                                                                       00001090
```

```
C     IF IT IS DESIRED TO COMPUTE THE PARTIAL DERIVATIVES OF THE VELO-  00001100
C     CITY VECTOR WITH RESPECT TO THE EQUINOCTIAL ELEMENTS, FIRST       00001110
C     COMPUTE THE FOLLOWING AUXILIARY QUANTITIES:                       00001120
C                                                                       00001130
C       A/(R**3)*SQRT(1-H**2-K**2), A/G, ∂X/∂H, ∂X/∂K, ∂Y/∂H, ∂Y/∂K     00001140
C                                                                       00001150
C                                                                       00001160
   20 IF (IRVPAR.EQ.1)                    RETURN                        00001170
C                                                                       00001175
      AXR3 = EL(1)**2/(AHK*EL(14)**3)                                   00001180
C                                                                       00001185
      AOG  = EL(1)/GG                                                   00001190
C                                                                       00001195
      PXDWRH =  AOG*YYD**2 + AXR3*(GG*EL(3)*BETA*XX - EL(7)*EL(1)*YY**2)00001200
C                                                                       00001210
      PXDWRK =  AOG*XXD*YYD - AXR3*(GG*EL(2)*BETA*XX +EL(7)*EL(1)*XX*YY)00001220
C                                                                       00001230
      PYDWRH = -AOG*XXD*YYD + AXR3*(GG*EL(3)*BETA*YY +EL(7)*EL(1)*XX*YY)00001240
C                                                                       00001250
      PYDWRK = -AOG*XXD**2 - AXR3*(GG*EL(2)*BETA*YY - EL(7)*EL(1)*XX**2)00001260
C                                                                       00001270
C     COMPUTE THE PARTIAL DERIVATIVES OF THE VELOCITY VECTOR WITH       00001280
C     RESPECT TO THE EQUINOCTIAL ORBITAL ELEMENTS                       00001285
C                                                                       00001290
      DO  30   IROW = 1,3                                               00001300
C                                                                       00001305
      PVDEQ(IROW,1) = -ST(IROW+3)/(2.D0*EL(1))                          00001310
C                                                                       00001320
      PVDEQ(IROW,2) =  PXDWRH * F(IROW) + PYDWRH * G(IROW)              00001330
C                                                                       00001340
      PVDEQ(IROW,3) =  PXDWRK * F(IROW) + PYDWRK * G(IROW)              00001350
C                                                                       00001360
      PVDEQ(IROW,4) =  (2.D0/PQ) * (EL(5)*EL(13)*YYD*F(IROW)            00001370
     *                -EL(5)*EL(13)*XXD*G(IROW) - XXD*W(IROW))         00001380
C                                                                       00001390
      PVDEQ(IROW,5) =  (2.D0*EL(13)/PQ) * (EL(4)*XXD*G(IROW)            00001400
     *                -EL(4)*YYD*F(IROW) + YYD*W(IROW))                00001410
C                                                                       00001420
      PVDEQ(IROW,6) = -EL(7) * ((EL(1)/EL(14))**3) * ST(IROW)           00001430
C                                                                       00001435
   30 CONTINUE                                                          00001440
C                                                                       00001445
C                                                                       00001450
C     CONSTRUCT THE FULL MATRIX OF STATE PARTIALS:                      00001460
C                                                                       00001470
C                                    III    III                         00001480
C                                    I        I                         00001490
C                                    I   ∂R   I                         00001500
C                                    I  ----  I                         00001510
C                                    I  ∂Ā    I                         00001520
C                        ∂X          I   ∂Ā   I                         00001530
C                       ----    =    I--------I                         00001540
C                        ∂Ā          I    .   I                         00001550
C                                    I  ∂R    I                         00001560
C                                    I   ∂R   I                         00001570
C                                    I  ----  I                         00001580
C                                    I   ∂    I                         00001590
```

```
PXTEQ                JSS1493.PRLLEL.ORBDET.FORT

C                                    I   ∂A   I                                00001600
C                                    III     III                              00001610
C                                                                             00001620
      DO  50   JCOL = 1,6                                                     00001630
      DO  40   IROW = 1,3                                                     00001640
      PXDEQ(IROW,JCOL)    = PRDEQ(IROW,JCOL)                                  00001650
   40 PXDEQ(IROW+3,JCOL) = PVDEQ(IROW,JCOL)                                  00001660
   50 CONTINUE                                                                00001670
      RETURN                                                                  00001680
      END                                                                     00001690

******** END OF MEMBER PXTEQ      181 RECORDS ********
```

```
      SUBROUTINE PXTSLV(ST,EL,TCUR,TEPOK)                                00000010
C                                                                        00000020
C     VERSION OF 09/15/79      **** CHECKED OUT ****                     00000030
C                                                                        00000040
C     PURPOSE                                                            00000050
C         TO COMPUTE THE PARTIAL DERIVATIVES OF THE CURRENT POSITION     00000060
C         AND VELOCITY WITH RESPECT TO THE SOLVE-FOR PARAMETERS FOR      00000070
C         USE IN THE ESTIMATION SUBROUTINES                              00000080
C                                                                        00000090
C     METHOD                                                             00000100
C         THIS SUBROUTINE COMPUTES THE PARTIAL DERIVATIVES OF CURRENT    00000110
C         POSITION AND VELOCITY WITH RESPECT TO CURRENT EQUINOCTIAL      00000120
C         ELEMENTS BY THE EXACT FORMULAE IN SUBROUTINE 'PXTEQ'.          00000130
C         THE PARTIAL DERIVATIVES OF CURRENT EQUINOCTIAL ELEMENTS        00000140
C         WITH RESPECT TO EPOCH EQUINOCTIAL ELEMENTS ARE COMPUTED BY     00000150
C         THE ANALYTICAL FORMULAE OF SUBROUTINE 'PATAO'.  THESE          00000160
C         MATRICES ARE MULTIPLIED TO PROVIDE THE PARTIAL DERIVATIVES     00000170
C         OF CURRENT POSITION AND VELOCITY WITH RESPECT TO EPOCH         00000180
C         EQUINOCTIAL ELEMENTS.  IF IT IS EPOCH EQUINOCTIAL ELEMENTS     00000190
C         THAT ARE THE SOLVE-FOR PARAMETERS, THEN NO FURTHER             00000200
C         COMPUTATIONS ARE PERFORMED.                                    00000210
C                                                                        00000220
C         IF IT IS EPOCH KEPLERIAN ELEMENTS THAT ARE THE SOLVE-FOR       00000230
C         PARAMETERS, THEN THE MATRIX "PEODKO" IS RETRIEVED FROM         00000240
C         THE LABELLED COMMON BLOCK /XEPOCH/ TO PROVIDE THE              00000250
C         EXACT PARTIAL DERIVATIVES OF THE EPOCH EQUINOCTIAL             00000260
C         ELEMENTS WITH RESPECT TO THE EPOCH KEPLERIAN ELEMENTS.         00000270
C         THESE RESULTS ARE PREMULTIPLIED BY THE ABOVE RESULTS TO        00000280
C         GIVE THE PARTIAL DERIVATIVES OF CURRENT POSITION AND           00000290
C         VELOCITY WITH RESPECT TO EPOCH KEPLERIAN ELEMENTS.             00000300
C                                                                        00000310
C         IF IT IS EPOCH POSITION AND VELOCITY THAT ARE THE SOLVE-FOR    00000320
C         PARAMETERS, THEN THE MATRIX "PEODXO" IS RETRIEVED FROM         00000330
C         THE LABELLED COMMON BLOCK /XEPOCH/ TO PROVIDE THE              00000340
C         EXACT PARTIAL DERIVATIVES OF THE EPOCH EQUINOCTIAL             00000350
C         ELEMENTS WITH RESPECT TO EPOCH POSITION AND VELOCITY.          00000360
C         THESE RESULTS ARE PREMULTIPLIED BY THE RESULTS FROM THE        00000370
C         FIRST PARAGRAPH TO GIVE THE PARTIAL DERIVATIVES OF CURRENT     00000380
C         POSITION AND VELOCITY WITH RESPECT TO EPOCH POSITION AND       00000390
C         VELOCITY.                                                      00000400
C                                                                        00000410
C     CALLING SEQUENCE                                                   00000420
C         CALL PXTSLV(ST,EL,TCUR,TEPOK)                                  00000430
C                                                                        00000440
C     CALLING SEQUENCE PARAMETERS                                        00000450
C         ST        I   ARRAY (6X1) CONTAINING THE CURRENT POSITION AND  00000460
C                       VELOCITY COMPONENTS                              00000470
C         EL        I   ARRAY (26X1) CONTAINING CURRENT EQUINOCTIAL      00000480
C                       ELEMENTS AND AUXILIARY PARAMETERS                00000490
C         TCUR      I   CURRENT TIME                                     00000500
C         TEPOK     I   EPOCH TIME                                       00000510
C                                                                        00000520
C         INPUT VIA COMMON/ICNTRL/:                                      00000530
C         ISLVFR    I   FLAG INDICATING THE SOLVE-FOR VECTOR             00000540
C                         = 1:  SOLVE FOR EPOCH POSITION AND VELOCITY    00000550
C                         = 2:  SOLVE FOR EPOCH KEPLERIAN ELEMENTS       00000560
C                         = 3:  SOLVE FOR EPOCH EQUINOCTIAL ELEMENTS     00000570
```

```
C         IJ2RTS   I   FLAG TO INCLUDE THE SECULAR RATES IN THE ARGUMENT00000580
C                      OF PERIAPSIS AND THE LONGITUDE OF THE ASCENDING  00000590
C                      NODE DUE TO THE J  ZONAL HARMONIC:               00000600
C                                       2                               00000610
C                      = 0:  DO NOT INCLUDE THE SECULAR RATES           00000620
C                      = 1:  INCLUDE THE SECULAR RATES                  00000630
C                                                                       00000640
C         OUTPUT IS VIA COMMON/STAPAR/:                                 00000650
C         PXDAEO   O   ARRAY (6X6) CONTAINING THE PARTIAL DERIVATIVES   00000660
C                      OF CURRENT POSITION AND VELOCITY WITH RESPECT TO 00000670
C                      EPOCH EQUINOCTIAL ELEMENTS                       00000680
C         PXDAKO   O   ARRAY (6X6) CONTAINING THE PARTIAL DERIVATIVES   00000690
C                      OF CURRENT POSITION AND VELOCITY WITH RESPECT TO 00000700
C                      EPOCH KEPLERIAN ELEMENTS                         00000710
C         PXDXO    O   ARRAY (6X6) CONTAINING THE PARTIAL DERIVATIVES   00000720
C                      OF CURRENT POSITION AND VELOCITY WITH RESPECT TO 00000730
C                      EPOCH POSITION AND VELOCITY                      00000740
C                                                                       00000750
C     EXTERNAL REFERENCES                                               00000760
C         M6666, PATA0, PXTEQ                                           00000770
C                                                                       00000780
C     ANALYSIS                                                          00000790
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000800
C                                                                       00000810
C     PROGRAMMING                                                       00000820
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000830
C                                                                       00000840
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000850
C                                                                       00000860
      IMPLICIT REAL*8(A-H,O-Z)                                          00000870
C                                                                       00000880
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000890
C                                                                       00000900
      DIMENSION       ST(6)       , EL(26)       ,  PAWRAO(6,6)       , 00000910
     *                PAEDAK(6,6)                                       00000920
C                                                                       00000930
C * * * * * * * * *         COMMON BLOCKS           * * * * * * * * * * 00000940
C                                                                       00000950
      COMMON/EQPTLX/  PEQDR(6,3)  , PEQDV(6,3)   ,  PEQDX(6,6)          00000960
C                                                                       00000970
      COMMON/ICNTRL/  IRNTYP      , IVINIT       ,  IVINTG            , 00000980
     1                ISLVFR      , NSLVFR       ,  IPTLS             , 00000990
     2                IJ2RTS      , IRVPAR       ,  IPARRV            , 00001000
     3                IPRNT1      , IPRNT2       ,  IPRNT3            , 00001010
     4                IPRNT4      , IPRNT5       ,  ICUT              , 00001020
     5                ICGCMP      , ISCID        ,  NPAR              , 00001030
C                                                                       00001040
      COMMON/STAPAR/  PXDAEO(6,6) , PXDAKO(6,6)  ,  PXDXO(6,6)         00001050
C                                                                       00001060
      COMMON/XEPOCH/  EPST(6)     , EPAK(26)     ,  EPAE(26)          , 00001070
     1                PEODKO(6,6) , PEODXO(6,6)                         00001080
C                                                                       00001090
      COMMON/XPTLEQ/  PRDEQ(3,6)  , PVDEQ(3,6)   ,  PXDEQ(6,6)          00001100
C                                                                       00001110
C     COMPUTE THE EXACT PARTIAL DERIVATIVES OF THE CURRENT POSITION AND 00001120
C     VELOCITY WITH RESPECT TO THE CURRENT EQUINOCTIAL ELEMENTS         00001130
C                                                                       00001140
```

```
      CALL PXTEQ(ST,EL,0)                                                00001150
C     WRITE(6,170)                                                       00001160
C     DO 1000  I = 1,6                                                   00001170
C     WRITE(6,200)  (PXDEQ(I,J),J=1,6)                                   00001180
C1000 CONTINUE                                                           00001190
C                                                                        00001200
C     COMPUTE THE ANALYTICAL PARTIAL DERIVATIVES OF THE CURRENT          00001210
C     EQUINOCTIAL ELEMENTS WITH RESPECT TO EPOCH EQUINOCTIAL ELEMENTS    00001220
C                                                                        00001230
      CALL PATA0(EL,TCUR,TEPOK,IJ2RTS,PAWRA0)                            00001240
C     WRITE(6,170)                                                       00001250
C     DO 2000  I = 1,6                                                   00001260
C     WRITE(6,200)  (PAWRA0(I,J),J=1,6)                                  00001270
C2000 CONTINUE                                                           00001280
C                                                                        00001290
C     MULTIPLY (IN THE ORDER COMPUTED) TO GET PARTIAL DERIVATIVES OF     00001300
C     CURRENT POSITION AND VELOCITY WITH RESPECT TO EPOCH EQUINOCTIAL    00001310
C     ELEMENTS                                                           00001320
C                                                                        00001330
      CALL M6666(PXDEQ,PAWRA0,PXDAE0)                                    00001340
C     WRITE(6,170)                                                       00001350
C     DO 3000  I = 1,6                                                   00001360
C     WRITE(6,200)  (PXDAE0(I,J),J=1,6)                                  00001370
C3000 CONTINUE                                                           00001380
      IF (ISLVFR.EQ.3)                  GO TO 40                         00001390
      IF (ISLVFR.EQ.1)                  GO TO 10                         00001400
C                                                                        00001410
C     RETRIEVE PARTIAL DERIVATIVES OF EPOCH EQUINOCTIAL ELEMENTS WITH    00001420
C     RESPECT TO EPOCH KEPLERIAN ELEMENTS FROM COMMON/XEPOCH/.           00001430
C     PREMULTIPLY THIS BY 'PXDAE0' TO GET THE PARTIAL DERIVATIVES OF     00001440
C     CURRENT POSITION AND VELOCITY WITH RESPECT TO EPOCH KEPLERIAN      00001450
C     ELEMENTS                                                           00001460
C                                                                        00001470
      CALL M6666(PXDAE0,PE0DK0,PXDAK0)                                   00001480
      GO TO 60                                                           00001490
C                                                                        00001500
C     RETRIEVE PARTIAL DERIVATIVES OF EPOCH EQUINOCTIAL ELEMENTS WITH    00001510
C     RESPECT TO EPOCH POSITION AND VELOCITY FROM COMMON/XEPOCH/.        00001520
C     PREMULTIPLY THIS BY 'PXDAE0' TO GET THE PARTIAL DERIVATIVES OF     00001530
C     THE CURRENT POSITION AND VELOCITY WITH RESPECT TO EPOCH POSITION   00001540
C     AND VELOCITY                                                       00001550
C                                                                        00001560
   10 CALL M6666(PXDAE0,PE0DX0,PXDX0)                                    00001570
C     WRITE(6,170)                                                       00001580
C     DO 4000  I = 1,6                                                   00001590
C     WRITE(6,200)  (PE0DX0(I,J),J=1,6)                                  00001600
C4000 CONTINUE                                                           00001610
C                                                                        00001620
C     OUTPUT 'PXDX0' IF ISLVFR = 1:                                      00001630
C                                                                        00001640
      IF (IFRNT2.EQ.0)                  RETURN                           00001650
      WRITE(6,100)                                                       00001660
      WRITE(6,130)                                                       00001670
      DO  20   J = 1,3                                                   00001680
      WRITE(6,140)  J,(PXDX0(I,J),I=1,6)                                 00001690
   20 CONTINUE                                                           00001700
      DO  30   J = 4,6                                                   00001710
```

```
PXTSLV              JSS1493.PRLLEL.ORBDET.FORT

      JMN3 = J - 3                                                          00001720
      WRITE(6,150)  JMN3,(PXDXO(I,J),I=1,6)                                 00001730
   30 CONTINUE                                                              00001740
      WRITE(6,170)                                                          00001750
      RETURN                                                                00001760
C                                                                           00001770
C     OUTPUT 'PXDAEO' IF ISLVFR = 3:                                        00001780
C                                                                           00001790
   40 IF (IPRNT2.EQ.0)                    RETURN                            00001800
      WRITE(6,110)                                                          00001810
      WRITE(6,130)                                                          00001820
      DO  50   J = 1,6                                                      00001830
      WRITE(6,160)  J,(PXDAEO(I,J),I=1,6)                                   00001840
   50 CONTINUE                                                              00001850
      WRITE(6,170)                                                          00001860
      RETURN                                                                00001870
C                                                                           00001880
C     OUTPUT 'PXDAKO' IF ISLVFR = 2:                                        00001890
C                                                                           00001900
   60 IF (IPRNT2.EQ.0)                    RETURN                            00001910
      WRITE(6,120)                                                          00001920
      WRITE(6,130)                                                          00001930
      DO  70   J = 1,6                                                      00001940
      WRITE(6,160)  J,(PXDAKO(I,J),I=1,6)                                   00001950
   70 CONTINUE                                                              00001960
      WRITE(6,170)                                                          00001970
      RETURN                                                                00001980
C                                                                           00001990
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * *    00002000
C                                                                           00002010
  100 FORMAT(/17X,'ANALYTIC PARTIAL DERIVATIVES OF CURRENT POSITION/',      00002020
     *       'VELOCITY WITH RESPECT TO EPOCH POSITION/VELOCITY')            00002030
C                                                                           00002040
  110 FORMAT(/10X,'ANALYTIC PARTIAL DERIVATIVES OF CURRENT POSITION/',      00002050
     *       'VELOCITY WITH RESPECT TO EPOCH EQUINOCTIAL ELEMENTS')         00002060
C                                                                           00002070
  120 FORMAT(/17X,'ANALYTIC PARTIAL DERIVATIVES OF CURRENT POSITION/',      00002080
     *       'VELOCITY WITH RESPECT TO EPOCH KEPLERIAN ELEMENTS')           00002090
C                                                                           00002100
  130 FORMAT(//20X,'R(1)',16X,'R(2)',16X,'R(3)',14X,'RDOT(1)',13X,          00002110
     *       'RDOT(2)',13X,'RDOT(3)')                                       00002120
C                                                                           00002130
  140 FORMAT(5X,'R0(',I1,')',6(1PD20.10))                                   00002140
C                                                                           00002150
  150 FORMAT(2X,'RODOT(',I1,')',6(1PD20.10))                                00002160
C                                                                           00002170
  160 FORMAT(4X,'EL0(',I1,')',6(1PD20.10))                                  00002180
C                                                                           00002190
  170 FORMAT(/)                                                             00002200
C                                                                           00002210
  200 FORMAT(10X,6(1PD20.10))                                               00002220
C                                                                           00002230
      END                                                                   00002240

******** END OF MEMBER PXTSLV     224 RECORDS ********
```

RUNSET JSS1493.PRLLEL.ORBDET.FORT

```
      SUBROUTINE RUNSET                                                 00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO READ THE DATA CARDS AND SET UP THE APPROPRIATE COMMON      00000060
C         BLOCKS FOR RUNNING THE PARALLEL ORBIT DETERMINATION SYSTEM    00000070
C         NOTE:  THIS IS ESSENTIALLY A SCALED-DOWN VERSION OF THE       00000080
C                GTDS SUBROUTINE 'SETRUN', WHICH PERFORMS THE SAME      00000090
C                FUNCTION IN THAT SYSTEM.                               00000100
C                                                                       00000110
C     CALLING SEQUENCE                                                  00000120
C         CALL RUNSET                                                   00000130
C                                                                       00000140
C     CALLING SEQUENCE PARAMETERS                                       00000150
C         NONE:  INPUT IS FROM THE APPROPRIATE FILE OF CARD IMAGES, AS  00000160
C                SPECIFIED IN THE JOB CONTROL LANGUAGE FOR THE RUN.     00000170
C                OUTPUT IS TO THE COMMON BLOCKS NECESSARY FOR EXECUTION 00000180
C                OF THE CURRENT RUN.                                    00000190
C                                                                       00000200
C     EXTERNAL REFERENCES                                               00000210
C         SUBROUTINES                                                   00000220
C                AKTOAE                                                 00000230
C                ELMRPT                                                 00000240
C                EQELPV                                                 00000250
C                PAETAK                                                 00000260
C                PEQTX                                                  00000270
C                                                                       00000280
C         COMMON BLOCKS                                                 00000290
C                /CNVRG /                                               00000300
C                /DRGPAR/                                               00000310
C                /EQPTLX/                                               00000320
C                /ERTSHP/                                               00000330
C                /ICNTRL/                                               00000340
C                /NPCNTL/                                               00000350
C                /STATE /                                               00000360
C                /TIMER /                                               00000370
C                /XEPOCH/                                               00000380
C                /XESTMT/                                               00000390
C                /ZNLCOF/                                               00000400
C                                                                       00000410
C     ANALYSIS                                                          00000420
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000430
C                                                                       00000440
C     PROGRAMMING                                                       00000450
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000460
C                                                                       00000470
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000480
C                                                                       00000490
      IMPLICIT REAL*8(A-H,O-Z)                                          00000500
C                                                                       00000510
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000520
C                                                                       00000530
      DIMENSION       EPEDUM(6)   ,  FIELD1(16)  ,  ICRD(16)           00000540
C                                                                       00000550
C * * * * * * * * *         COMMON BLOCKS           * * * * * * * * * * 00000560
C                                                                       00000570
```

```
RUNSET              JSS1493.PRLLEL.ORBDET.FORT

      COMMON/CNVRG /  CON(6)       ,  CONKEP                              00000580
C                                                                         00000590
      COMMON/DRGPAR/  CSUBD        ,  SCAREA       ,  SCMASS              , 00000600
     1                RHOZRO       ,  RZERO        ,  SCLHGT              , 00000610
     2                IDRAG                                               00000620
C                                                                         00000630
      COMMON/EQPTLX/  PEQDR(6,3)   ,  PEQDV(6,3)   ,  PEQDX(6,6)         00000640
C                                                                         00000650
      COMMON/ERTSHP/  REQ          ,  WE           ,  FINV                , 00000660
     1                SPHINF       ,  GM                                  00000670
C                                                                         00000680
      COMMON/ICNTRL/  IRNTYP       ,  IVINIT       ,  IVINTG              , 00000690
     1                ISLVFR       ,  NSLVFR       ,  IPTLS               , 00000700
     2                IJ2RTS       ,  IRVPAR       ,  IPARRV              , 00000710
     3                IPRNT1       ,  IPRNT2       ,  IPRNT3              , 00000720
     4                IPRNT4       ,  IPRNT5       ,  IOUT                , 00000730
     5                ICGCMP       ,  ISCID        ,  NPAR                , 00000740
     6                ICOV                                                00000750
C                                                                         00000760
      COMMON/NPCNTL/  MORD         ,  NSUB         ,  IEQSET              , 00000770
     1                ITERM                                               00000780
C                                                                         00000790
      COMMON/STATE /  S0(6)        ,  STK(70,6)    ,  STK1(70,6)          00000800
C                                                                         00000810
      COMMON/TIMER /  T0           ,  TSPAN        ,  TOUT                , 00000820
     1                H            ,  TINC                                00000830
C                                                                         00000840
      COMMON/XEPOCH/  EPST(6)      ,  EPAK(26)     ,  EPAE(26)            , 00000850
     1                PEODKO(6,6)  ,  PEODXO(6,6)                         00000860
C                                                                         00000870
      COMMON/XESTMT/  XAP(6)       ,  SCLFAC(6)    ,  EPSILN              , 00000880
     1                ALFA1        ,  STPSZ        ,  C1DIM               , 00000890
     2                CMTXAP(6,6)  ,  XMIN(6)                             00000900
C                                                                         00000910
      COMMON/ZNLCOF/  SHJ(8)       ,  IZNLS                               00000920
C                                                                         00000930
C * * * * * * * * *          DATA BLOCKS          * * * * * * * * * *     00000940
C                                                                         00000950
      DATA FIELD1  /  'EPHEMGEN'   ,  'ESTIMATE'   ,  'ELEMENT1'         , 00000960
     1                'ELEMENT2'   ,  'P-C INTG'   ,  'FORC MDL'         , 00000970
     2                'SCPARAM '   ,  'INT SPAN'   ,  'PARTIALS'         , 00000980
     3                'EST PARS'   ,  '1-D SRCH'   ,  'APSLVFOR'         , 00000990
     4                'APCOVMTX'   ,  'PRINTOUT'   ,  'SCLFCTRS'         , 00001000
     5                'LASTCARD'   /                                      00001010
C                                                                         00001020
C     BEGIN THE CARD-READ LOOP                                            00001030
C                                                                         00001040
      NBRCRD = 0                                                          00001050
      WRITE(6,1030)                                                       00001060
   10 READ(5,1000)  AA,I1,I2,I3,R1,R2,R3                                  00001070
      NBRCRD = NBRCRD + 1                                                 00001080
      WRITE(6,1010) AA,I1,I2,I3,R1,R2,R3                                  00001090
C                                                                         00001100
C     BRANCH TO THE APPROPRIATE KEYWORD PROCESSING SECTION                00001110
C                                                                         00001120
      PI  = 3.1415926535897932D0                                          00001130
      DTR = PI/180.D0                                                     00001140
```

```
      DO  20   I = 1,16                                                 00001150
      IF (AA.NE.FIELD1(I))              GO TO 20                        00001160
      ICARD = I                                                         00001170
      ICRD(ICARD) = 1                                                   00001180
      GO TO 30                                                          00001190
   20 CONTINUE                                                          00001200
C                                                                       00001210
C     ERROR MESSAGE FOR IMPROPER KEYWORD CARD                           00001220
C                                                                       00001230
      WRITE(6,1020)  NBRCRD                                             00001240
      GO TO 10                                                          00001250
C                                                                       00001260
C     JUMP TO APPROPRIATE KEYWORD PROCESSING LOGIC                      00001270
C                                                                       00001280
   30 GO TO (40,50,60,100,170,200,210,220,230,240,                      00001290
     1                    250,260,270,280,290,300) , ICARD              00001300
C                                                                       00001310
C     KEYWORD 'EPHEMGEN' PROCESSING                                     00001320
C                                                                       00001330
   40 IRNTYP = 1                                                        00001340
      IPTLS  = 0                                                        00001350
      IRVPAR = 0                                                        00001360
      IPARRV = 0                                                        00001370
      GO TO 10                                                          00001380
C                                                                       00001390
C     KEYWORD 'ESTIMATE' PROCESSING                                     00001400
C                                                                       00001410
   50 IRNTYP = 2                                                        00001420
      ISLVFR = 1                                                        00001430
      NSLVFR = 6                                                        00001440
      IPTLS  = 0                                                        00001450
      IRVPAR = 0                                                        00001460
      IPARRV = 0                                                        00001470
      GO TO 10                                                          00001480
C                                                                       00001490
C     KEYWORD 'ELEMENT1' PROCESSING                                     00001500
C                                                                       00001510
   60 IVINIT = I1                                                       00001520
      GO TO (70,80,90) , I1                                             00001530
   70 EPST(1) = R1                                                      00001540
      EPST(2) = R2                                                      00001550
      EPST(3) = R3                                                      00001560
      GO TO 10                                                          00001570
   80 EPAK(1) = R1                                                      00001580
      EPAK(2) = R2                                                      00001590
      EPAK(3) = R3 * DTR                                                00001600
      GO TO 10                                                          00001610
   90 RETRO = DFLOAT(I3)                                                00001620
      EPAE(1) = R1                                                      00001630
      EPAE(2) = R2                                                      00001640
      EPAE(3) = R3                                                      00001650
      GO TO 10                                                          00001660
C                                                                       00001670
C     KEYWORD 'ELEMENT2' PROCESSING                                     00001680
C                                                                       00001690
  100 GO TO (110,120,130) , IVINIT                                      00001700
  110 EPST(4) = R1                                                      00001710
```

```
      EPST(5) = R2                                                    00001720
      EPST(6) = R3                                                    00001730
      GO TO 140                                                       00001740
  120 EPAK(4) = R1 * DTR                                              00001750
      EPAK(5) = R2 * DTR                                              00001760
      EPAK(6) = R3 * DTR                                              00001770
      CALL AKTOAE(EPAK,RETRO,EPEDUM)                                  00001780
      CALL EQELPV(EPEDUM,RETRO,GM,EPST)                               00001790
      GO TO 140                                                       00001800
  130 EPAE(4) = R1                                                    00001810
      EPAE(5) = R2                                                    00001820
      EPAE(6) = R3 * DTR                                              00001830
      CALL EQELPV(EPAE,RETRO,GM,EPST)                                 00001840
  140 CALL ELMRPT(EPST,0.D0,0.D0,GM,1,0,EPAK)                         00001850
      CALL ELMRPT(EPST,0.D0,0.D0,GM,2,0,EPAE)                         00001860
      CALL PAETAK(EPAK,PEQDKO)                                        00001870
      CALL PEQTX(EPST,EPAE,0)                                         00001880
      DO  160  I = 1,6                                                00001890
      DO  150  J = 1,6                                                00001900
  150 PEQDXO(I,J) = PEQDX(I,J)                                        00001910
  160 CONTINUE                                                        00001920
      GO TO 10                                                        00001930
C                                                                     00001940
C     KEYWORD 'P-C INTG' PROCESSING                                   00001950
C                                                                     00001960
  170 MCRD   = I1                                                     00001970
      NSUB   = I2                                                     00001980
      IEQSET = I3                                                     00001990
      IVINTG = I3                                                     00002000
      ITERM  = R1                                                     00002010
      DO  180  I = 1,6                                                00002020
      CON(I) = R2                                                     00002030
  180 CONTINUE                                                        00002040
      CONKEP = R2                                                     00002050
      H      = R3                                                     00002060
      DO  190  I = 1,6                                                00002070
      IF (IEQSET.EQ.1)                S0(I) = EPST(I)                 00002080
      IF (IEQSET.EQ.2)                S0(I) = EPAK(I)                 00002090
      IF (IEQSET.EQ.3)                S0(I) = EPAE(I)                 00002100
  190 CONTINUE                                                        00002110
      GO TO 10                                                        00002120
C                                                                     00002130
C     KEYWORD 'FORC MDL' PROCESSING                                   00002140
C                                                                     00002150
  200 IZNLS  = I1                                                     00002160
      IDRAG  = I3                                                     00002170
      IF (I2.EQ.0)                    GO TO 10                        00002180
      RHOZRO = R1                                                     00002190
      RZERO  = R2                                                     00002200
      SCLHGT = R3                                                     00002210
      GO TO 10                                                        00002220
C                                                                     00002230
C     KEYWORD 'SCPARAM ' PROCESSING                                   00002240
C                                                                     00002250
  210 ISCID  = I1                                                     00002260
      CSUBD  = R1                                                     00002270
      SCAREA = R2                                                     00002280
```

```
      SCMASS = R3                                                       00002290
      GO TO 10                                                          00002300
C                                                                       00002310
C     KEYWORD 'INT SPAN' PROCESSING                                     00002320
C                                                                       00002330
  220 T0 = 0.D0                                                         00002340
      TDAYS  = DFLOAT(I2) * 24.D0 * 3600.D0                             00002350
      THOURS = DFLOAT(I3) * 3600.D0                                     00002360
      TMINS  = R1 * 60.D0                                               00002370
      TSPAN  = TDAYS + THOURS + TMINS + R2                              00002380
      TINC   = R3                                                       00002390
      GO TO 10                                                          00002400
C                                                                       00002410
C     KEYWORD 'PARTIALS' PROCESSING                                     00002420
C                                                                       00002430
  230 IPTLS  = I1                                                       00002440
      ISLVFR = I2                                                       00002450
      NSLVFR = I3                                                       00002460
      IJ2RTS = R2                                                       00002470
      GO TO 10                                                          00002480
C                                                                       00002490
C     KEYWORD 'EST PARS' PROCESSING                                     00002500
C                                                                       00002510
  240 ICGCMP = I1                                                       00002520
      NPAR   = I3                                                       00002530
      EPSILN = R2                                                       00002540
      ICOV   = R3                                                       00002550
      GO TO 10                                                          00002560
C                                                                       00002570
C     KEYWORD '1-D SRCH' PROCESSING                                     00002580
C                                                                       00002590
  250 ALFA1  = R1                                                       00002600
      STPSZ  = R2                                                       00002610
      C1DIM  = R3                                                       00002620
      GO TO 10                                                          00002630
C                                                                       00002640
C     KEYWORD 'APSLVFOR' PROCESSING                                     00002650
C                                                                       00002660
  260 XAP(I1)   = R1                                                    00002670
      XAP(I1+1) = R2                                                    00002680
      XAP(I1+2) = R3                                                    00002690
      IF (ISLVFR.EQ.2.AND.I1.EQ.1)      XAP(I1+2) = XAP(I1+2) * DTR     00002700
      IF (ISLVFR.EQ.3.AND.I1.EQ.4)      XAP(I1+2) = XAP(I1+2) * DTR     00002710
      IF (ISLVFR.EQ.2.AND.I1.EQ.4)      XAP(I1)   = XAP(I1) * DTR       00002720
      IF (ISLVFR.EQ.2.AND.I1.EQ.4)      XAP(I1+1) = XAP(I1+1) * DTR     00002730
      IF (ISLVFR.EQ.2.AND.I1.EQ.4)      XAP(I1+2) = XAP(I1+2) * DTR     00002740
      GO TO 10                                                          00002750
C                                                                       00002760
C     KEYWORD 'APCOVMTX' PROCESSING                                     00002770
C                                                                       00002780
  270 CMTXAP(I1,I3)   = R1                                              00002790
      CMTXAP(I1,I3+1) = R2                                              00002800
      CMTXAP(I1,I3+2) = R3                                              00002810
      GO TO 10                                                          00002820
C                                                                       00002830
C     KEYWORD 'PRINTOUT' PROCESSING                                     00002840
C                                                                       00002850
```

```
RUNSET              JSS1493.PRLLEL.ORBDET.FORT

  280 IPRNT1 = I1                                                        00002860
      IPRNT2 = I2                                                        00002870
      IPRNT3 = I3                                                        00002880
      IPRNT4 = R1                                                        00002890
      IPRNT5 = R2                                                        00002900
      IOUT   = R3                                                        00002910
      GO TO 10                                                           00002920
C                                                                        00002930
C     KEYWORD 'SCLFCTRS' PROCESSING                                      00002940
C                                                                        00002950
  290 SCLFAC(I1)   = R1                                                  00002960
      SCLFAC(I1+1) = R2                                                  00002970
      SCLFAC(I1+2) = R3                                                  00002980
      GO TO 10                                                           00002990
C                                                                        00003000
C     KEYWORD 'LASTCARD' PROCESSING                                      00003010
C                                                                        00003020
  300 IF (IRNTYP.EQ.2)                  GO TO 320                        00003030
      ITOT = ICRD(1) + ICRD(3) + ICRD(4) + ICRD(5) + ICRD(6) + ICRD(8)   00003040
     1     + ICRD(14) + ICRD(16)                                         00003050
      IF (ITOT.NE.8)                    WRITE(6,1040)                    00003060
      RETURN                                                             00003070
  320 ITOT = ICRD(2) + ICRD(3) + ICRD(4) + ICRD(5) + ICRD(6)             00003080
      DO  330   I = 8,16                                                 00003090
      ITOT = ITOT + ICRD(I)                                              00003100
  330 CONTINUE                                                           00003110
      IF (ITOT.NE.14)                   WRITE(6,1050)                    00003120
      RETURN                                                             00003130
C                                                                        00003140
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * * 00003150
C                                                                        00003160
 1000 FORMAT(A8,3I3,3G21.14)                                             00003170
C                                                                        00003180
 1010 FORMAT(23X,A8,3I3,3(2X,1PD21.14)/)                                 00003190
C                                                                        00003200
 1020 FORMAT(1X,22('*'),9X,'WARNING --- CARD ',I2,' DOES NOT CONTAIN ',  00003210
     1       'A RECOGNIZED KEYWORD IN FIELD 1',9X,23('*')/)              00003220
C                                                                        00003230
 1030 FORMAT(23X,18('*'),'PARALLEL PROCESSING ORBIT DETERMINATION ',     00003240
     1       'INPUT DECK',18('*')/23X,86('_')//)                         00003250
C                                                                        00003260
 1040 FORMAT(15X,'WARNING --- AT LEAST ONE MANDATORY CARD FOR AN ',      00003270
     1       'EPHEMERIS GENERATION RUN IS MISSING FROM THE INPUT ',      00003280
     2       'DECK')                                                     00003290
C                                                                        00003300
 1050 FORMAT(16X,'WARNING --- AT LEAST ONE MANDATORY CARD FOR A PARA',   00003310
     1       'LLEL ESTIMATION RUN IS MISSING FROM THE INPUT DECK')       00003320
C                                                                        00003330
      END                                                                00003340

******** END OF MEMBER RUNSET     334 RECORDS ********
```

```
      SUBROUTINE SERCH1(XN,SN,N,ALFA1,STPSZ,CNVG,I1,I2,I3,                00000010
     *                                  ALFAN,XN1,CXN1)                  00000020
C                                                                         00000030
C     VERSION OF 09/15/79      **** CHECKED OUT ****                      00000040
C                                                                         00000050
C     PURPOSE                                                             00000060
C        TO FIND THE UNCONSTRAINED MINIMUM OF A NONLINEAR FUNCTION        00000070
C        F(X) OF A SINGLE VARIABLE X.                                     00000080
C                                                                         00000090
C     METHOD                                                              00000100
C        THE PROCEDURE IS GENERALLY ATTRIBUTED TO COGGINS.  IT IS A       00000110
C        COMBINATION OF TWO SINGLE VARIABLE METHODS, ONE DUE TO           00000120
C        DAVIES, SWANN AND CAMPEY AND ONE DUE TO POWELL.  UNIMODALITY     00000130
C        IS ASSUMED AND THUS THE USE OF MULTIPLE STARTING POINTS IS       00000140
C        RECOMMENDED IF A MULTIMODAL FUNCTION IS SUSPECTED.  THE          00000150
C        ALGORITHM PROCEEDS AS FOLLOWS:                                   00000160
C                                                                         00000170
C        1)  A STARTING POINT IS CHOSEN AND THE OBJECTIVE FUNCTION IS     00000180
C        EVALUATED.                                                       00000190
C                                                                         00000200
C        2)  THE INDEPENDENT VARIABLE IS INCREMENTED A DISTANCE DELX      00000210
C        AND THE OBJECTIVE FUNCTION EVALUATED AGAIN.  IF A FUNCTION       00000220
C        IMPROVEMENT (REDUCTION) IS OBTAINED, THE STEPSIZE IS DOUBLED     00000230
C        FOR THE NEXT FUNCTION EVALUATION.  IF A FUNCTION IMPROVEMENT     00000240
C        IS NOT OBTAINED ON THE FIRST STEP, THE DIRECTION IS REVERSED     00000250
C        AND THE NEXT POINT IS LOCATED A DISTANCE -DELX FROM THE          00000260
C        STARTING POINT.                                                  00000270
C                                                                         00000280
C        3)  AFTER THE FIRST STEP, THE STEPSIZE IS DOUBLED IF A FUNC-     00000290
C        TION IMPROVEMENT IS OBTAINED AND HALVED IF A POORER FUNCTION     00000300
C        EVALUATION IS OBTAINED.                                          00000310
C                                                                         00000320
C        4)  WHEN A LOCAL OPTIMUM IS ENCOUNTERED (I.E., THE OBJECTIVE     00000330
C        FUNCTION DECREASES AND THEN STARTS TO INCREASE AGAIN), THE       00000340
C        PROCEDURE WILL YIELD THREE POINTS (XK,XK-1,XK-2) STRADDLING      00000350
C        THE OPTIMUM.  AN ADDITIONAL POINT, XK+1, IS THEN LOCATED:        00000360
C                                                                         00000370
C                              X    = X    + DELX/2                      00000380
C                               K+1    K-1                               00000390
C                                                                         00000400
C        WHERE DELX IS THE CURRENT STEPSIZE.  THE BEST THREE POINTS       00000410
C        ARE THEN RETAINED.                                               00000420
C                                                                         00000430
C        5)  A QUADRATIC FUNCTION, FQ(X), OF THE FORM:                    00000440
C                                                                         00000450
C                                    2                                    00000460
C                     FQ(X) = A * X  + B * X + C  = Y                     00000470
C                                                                         00000480
C        IS THEN CURVE FITTED TO THE THREE RETAINED POINTS.               00000490
C                                                                         00000500
C                                   *                                     00000510
C        THE OPTIMUM LOCATION, X , IS LOCATED BY SETTING THE PARTIAL      00000520
C        DERIVATIVE OF FQ(X) WITH RESPECT TO X EQUAL TO ZERO, SUCH        00000530
C        THAT:                                                            00000540
C                                                                         00000550
C                                     *                                   00000560
C            2*A*X + B = 0     =>    X  = -B/2*A                          00000570
```

```
C                                                                       00000580
C       WHERE A AND B ARE DETERMINED FROM THE THREE GIVEN POINTS BY     00000590
C       SOLVING THE THREE SIMULTANEOUS QUADRATIC EQUATIONS.             00000600
C                                                                       00000610
C                                        *                              00000620
C       6)  THE OBJECTIVE FUNCTION AT X  IS THEN COMPARED WITH THE      00000630
C       BEST PREVIOUS POINT SUBJECT TO A CONVERGENCE LIMIT,             00000640
C                                                                       00000650
C                          *                                            00000660
C                 DABS(X   - X (BEST) ).LE.CNVG                         00000670
C                        I                                              00000680
C                                                                       00000690
C       IF THE ABOVE CRITERION IS SATISFIED, THE PROCEDURE STOPS.  IF   00000700
C       NOT, THE WORST POINT IS REPLACED BY THE PREVIOUS OPTIMUM POINT  00000710
C       AND A NEW QUADRATIC SURFACE FITTED AND LOCAL OPTIMUM OBTAINED.  00000720
C       THIS PROCESS IS REPEATED UNTIL THE CONVERGENCE CRITERION IS     00000730
C       SATISFIED.                                                      00000740
C                                                                       00000750
C    REFERENCES                                                         00000760
C                                                                       00000770
C        1)  FOR COGGINS' METHOD:                                       00000780
C                                                                       00000790
C            KUESTER, J.L. AND J.H. MIZE, "OPTIMIZATION TECHNIQUES WITH 00000800
C            FORTRAN," MCGRAW-HILL BOOK COMPANY, NEW YORK, 1973.        00000810
C                                                                       00000820
C        2)  FOR DAVIES, SWANN, AND CAMPEY METHOD:                      00000830
C                                                                       00000840
C            BOX, M.J., D. DAVIES AND W.H. SWANN, "NON-LINEAR OPTIMI-   00000850
C            ZATION TECHNIQUES," IMPERIAL CHEMICAL INDUSTRIES           00000860
C            MONOGRAPH NO. 5, OLIVER AND BOYD, EDINBURGH, 1969.         00000870
C                                                                       00000880
C        3)  FOR POWELL METHOD:                                         00000890
C                                                                       00000900
C            POWELL, M.J.D., "AN EFFICIENT METHOD FOR FINDING THE       00000910
C            MINIMUM OF A FUNCTION OF SEVERAL VARIABLES WITHOUT         00000920
C            CALCULATING DERIVATIVES," COMPUTER JOURNAL, VOL. 7,        00000930
C            1964, PP. 155-162.                                         00000940
C                                                                       00000950
C    CALLING SEQUENCE                                                   00000960
C        CALL SERCH1(XN,SN,N,ALFA1,STPSZ,CNVG,I1,I2,I3,ALFAN,XN1,CXN1)  00000970
C                                                                       00000980
C    CALLING SEQUENCE PARAMETERS                                        00000990
C        XN       I     VECTOR (6X1) CONTAINING THE CURRENT ITERATE OF  00001000
C                       THE SOLVE-FOR VECTOR X FROM SUBROUTINE          00001010
C                       'ESTMAT'.                                       00001020
C        SN       I     VECTOR (6X1) CONTAINING THE CURRENT ITERATE OF  00001030
C                       THE SEARCH DIRECTION VECTOR S FROM SUBROUTINE   00001040
C                       'ESTMAT'.                                       00001050
C        N        I     DIMENSION OF THE SOLVE-FOR VECTOR X.            00001060
C        ALFA1    I     STARTING VALUE OF THE INDEPENDENT VARIABLE      00001070
C                       ALPHA.                                          00001080
C        STPSZ    I     INITIAL ESTIMATE OF THE SEARCH STEPSIZE.        00001090
C        CNVG     I     CONVERGENCE CRITERION FOR THE POWELL METHOD.    00001100
C        I1       I     FLAG TO INDICATE THAT THE OBSERVATION PRO-      00001110
C                       CESSING REPORT SHOULD BE PRINTED:               00001120
C                       0 = DO NOT PRINT                                00001130
C                       1 = PRINT OBSERVATION PROCESSING REPORT         00001140
```

```
C         I2        I     FLAG TO INDICATE THE METHOD OF EVALUATING THE  00001150
C                         COST FUNCTIONAL AND ITS GRADIENT:              00001160
C                         1 = DIRECT NUMERICAL INTEGRATION               00001170
C                         2 = LINEAR APPROXIMATION                       00001180
C         I3        I     FLAG TO INDICATE THAT THE OBSERVATION PARTIAL  00001190
C                         DERIVATIVE REPORT SHOULD BE PRINTED:           00001200
C                         0 = DO NOT PRINT                               00001210
C                         1 = PRINT OBSERVATION PARTIAL DERIVATIVES      00001220
C         ALFAN     O     THE CONVERGED VALUE OF ALPHA WHICH MINIMIZES   00001230
C                         THE COST FUNCTIONAL:                           00001240
C                                                                        00001250
C                                    N                                   00001260
C                              F(X   + ALFA * S ) = MIN                  00001270
C                                                 N                      00001280
C                                                                        00001290
C         XN1       O     VECTOR (6X1) CONTAINING THE NEW ITERATE OF THE 00001300
C                         SOLVE-FOR VECTOR TO BE RETURNED TO SUBROUTINE  00001310
C                         'ESTMAT':                                      00001320
C                                                                        00001330
C                                   N+1    N                            00001340
C                                  X    = X  + ALFA  * S                00001350
C                                                N    N                  00001360
C                                                                        00001370
C         CXN1      O     COST FUNCTIONAL EVALUATED AT THE NEW ITERATE OF00001380
C                         THE SOLVE-FOR VECTOR.                          00001390
C                                                                        00001400
C     EXTERNAL REFERENCES                                                00001410
C         SUBROUTINES                                                    00001420
C              CGINTG                                                    00001430
C              CGLINE                                                    00001440
C              OBSOUT                                                    00001450
C              PTLOUT                                                    00001460
C                                                                        00001470
C     ANALYSIS                                                           00001480
C         GENERAL ALGORITHM DOCUMENTED IN REFERENCE 1); ALGORITHM MODI-  00001490
C         FIED BY CAPTAIN JEFFREY S. SHAVER, USAF, TO DO THE FOLLOWING:  00001500
C         1)  SOLVE FOR A MINIMUM INSTEAD OF A MAXIMUM                   00001510
C         2)  ALLOW A VECTOR INPUT TO THE COST FUNCTIONAL, WHERE THE     00001520
C             VECTOR IS A FUNCTION OF THE SCALAR ALPHA.                  00001530
C                                                                        00001540
C     PROGRAMMING                                                        00001550
C         CAPTAIN JEFFREY S. SHAVER, USAF                                00001560
C                                                                        00001570
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00001580
C                                                                        00001590
      IMPLICIT REAL*8(A-H,O-Z)                                           00001600
C                                                                        00001610
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00001620
C                                                                        00001630
      DIMENSION       XN(6)       ,  SN(6)        ,  XN1(6)          , 00001640
     *                GDUM(6)                                           00001650
C                                                                        00001660
      DIMENSION       WW(3)       ,  YF(3)        ,  ZZ(3)           , 00001670
C                                                                        00001680
C * * * * * * * * *          COMMON BLOCKS          * * * * * * * * * * 00001690
C                                                                        00001700
      COMMON/ISTSTC/  ITCNT  ,  IF1CNT ,  IG1CNT ,  IF2CNT ,           00001710
```

```
     *               IG2CNT ,  ICCNT  ,  IFG1WO ,  IFG2WO ,          00001720
     *               IFG1W  ,  IFG2W  ,  IT1WO  ,  IT2WO  ,          00001730
     *               IT1W   ,  IT2W                                  00001740
C                                                                    00001750
      COMMON/XESTMT/  XAP(6)                                         00001760
C                                                                    00001770
      WRITE(6,500)                                                   00001780
      WRITE(6,510)  ALFA1,STPSZ,CNVG,N                               00001790
      STPSV = STPSZ                                                  00001800
C                                                                    00001810
C     EVALUATE THE FUNCTION FOR THE INITIAL VALUE OF THE INDEPENDENT 00001820
C     VARIABLE, ALPHA.                                               00001830
C                                                                    00001840
C     DAVIES, SWANN AND CAMPEY ALGORITHM:                            00001850
C                                                                    00001860
      WRITE(6,520)                                                   00001870
      DO  10   IROW = 1,N                                            00001880
      XN1(IROW) = XN(IROW) + ALFA1 * SN(IROW)                        00001890
   10 XAP(IROW) = XN1(IROW)                                          00001900
      CALL CGINTG(XN1,Y1,GDUM)                                       00001910
      ICCNT = 1                                                      00001920
      WRITE(6,530)  ALFA1,Y1,STPSZ                                   00001930
      DO  1000 IROW = 1,N                                            00001940
      WRITE(6,540)  XN(IROW),ALFA1,SN(IROW),XN1(IROW)                00001950
 1000 CONTINUE                                                       00001960
      IF (I1.EQ.1)            CALL OBSOUT                          00001970
      IF (I3.EQ.1)            CALL PTLOUT                          00001980
C                                                                    00001990
C     INCREMENT THE INDEPENDENT VARIABLE AND EVALUATE FUNCTION.      00002000
C                                                                    00002010
      ALFA2 = ALFA1 + STPSZ                                          00002020
      II = 0                                                         00002030
   20 DO  30   IROW = 1,N                                            00002040
      XN1(IROW) = XN(IROW) + ALFA2 * SN(IROW)                        00002050
   30 CONTINUE                                                       00002060
      IF (I2.EQ.1)            CALL CGINTG(XN1,Y2,GDUM)             00002070
      IF (I2.EQ.2)            CALL CGLINE(XN1,Y2,GDUM)             00002080
      ICCNT = ICCNT + 1                                              00002090
      WRITE(6,530)  ALFA2,Y2,STPSZ                                   00002100
      DO  2000 IROW = 1,N                                            00002110
      WRITE(6,540)  XN(IROW),ALFA2,SN(IROW),XN1(IROW)                00002120
 2000 CONTINUE                                                       00002130
      IF (I1.EQ.1)            CALL OBSOUT                          00002140
      IF (I3.EQ.1)            CALL PTLOUT                          00002150
C                                                                    00002160
C     DETERMINE WHICH OF THE FUNCTION EVALUATIONS IS LEAST.  IF Y2 IS 00002170
C     LESS THAN Y1, THE CURRENT DIRECTION OF SEARCH IS IN THE DIRECTION 00002180
C     OF THE MINIMUM AND THE SEARCH SHOULD CONTINUE IN THIS DIRECTION; 00002190
C     IF NOT, THE SEARCH SHOULD BEGIN IN THE OPPOSITE DIRECTION.     00002200
C                                                                    00002210
   40 IF (Y2 - Y1)    80,80,50                                       00002220
   50 IF (II -  1)    60,70,70                                       00002230
   60 II = II + 1                                                    00002240
      STPSZ = -STPSZ                                                 00002250
      ALFA2 = ALFA1 + STPSZ                                          00002260
      GO TO 20                                                       00002270
C                                                                    00002280
```

```
C     IF THE MINIMUM IS BRACKETED, TRANSFER CONTROL TO THE POWELL       00002290
C     ALGORITHM.                                                        00002300
C                                                                       00002310
   70 GO TO 160                                                         00002320
   80 CONTINUE                                                          00002330
      ALFA3 = ALFA2 + STPSZ                                             00002340
      GO TO 100                                                         00002350
   90 ALFA3 = ALFA4                                                     00002360
      Y3 = Y4                                                           00002370
      GO TO 120                                                         00002380
  100 DO  110  IROW = 1,N                                               00002390
      XN1(IROW) = XN(IROW) + ALFA3 * SN(IROW)                           00002400
  110 CONTINUE                                                          00002410
      IF (I2.EQ.1)            CALL CGINTG(XN1,Y3,GDUM)                  00002420
      IF (I2.EQ.2)            CALL CGLINE(XN1,Y3,GDUM)                  00002430
      ICCNT = ICCNT + 1                                                 00002440
      WRITE(6,530)  ALFA3,Y3,STPSZ                                      00002450
      DO  3000 IROW = 1,N                                               00002460
      WRITE(6,540)  XN(IROW),ALFA3,SN(IROW),XN1(IROW)                   00002470
 3000 CONTINUE                                                          00002480
      IF (I1.EQ.1)            CALL OBSOUT                               00002490
      IF (I3.EQ.1)            CALL PTLOUT                               00002500
C                                                                       00002510
C     IF THE FUNCTION EVALUATION IS LESS THAN THE PREVIOUS VALUE, DOUBLE00002520
C     THE STEPSIZE.                                                     00002530
C                                                                       00002540
  120 STPSZ = 2.D00 * STPSZ                                             00002550
      ALFA4 = ALFA3 + STPSZ                                             00002560
      DO  130  IROW = 1,N                                               00002570
      XN1(IROW) = XN(IROW) + ALFA4 * SN(IROW)                           00002580
  130 CONTINUE                                                          00002590
      IF (I2.EQ.1)            CALL CGINTG(XN1,Y4,GDUM)                  00002600
      IF (I2.EQ.2)            CALL CGLINE(XN1,Y4,GDUM)                  00002610
      ICCNT = ICCNT + 1                                                 00002620
      WRITE(6,530)  ALFA4,Y4,STPSZ                                      00002630
      DO  4000 IROW = 1,N                                               00002640
      WRITE(6,540)  XN(IROW),ALFA4,SN(IROW),XN1(IROW)                   00002650
 4000 CONTINUE                                                          00002660
      IF (I1.EQ.1)            CALL OBSOUT                               00002670
      IF (I3.EQ.1)            CALL PTLOUT                               00002680
      IF (Y4 - Y3)    90,90,170                                         00002690
C                                                                       00002700
C     WHEN THE MINIMUM IS STRADDLED, EVALUATE THREE POINTS ABOUT THE    00002710
C     MINIMUM.                                                          00002720
C                                                                       00002730
C     POWELL ALGORITHM:                                                 00002740
C                                                                       00002750
  160 ZZ(1) = ALFA1                                                     00002760
      ZZ(2) = ALFA1 + STPSZ/2.D0                                        00002770
      ZZ(3) = ALFA2                                                     00002780
      GO TO 180                                                         00002790
  170 ZZ(1) = ALFA3                                                     00002800
      ZZ(2) = ALFA3 + STPSZ/2.D0                                        00002810
      ZZ(3) = ALFA4                                                     00002820
C                                                                       00002830
C     EVALUATE THE COST FUNCTIONAL AT THESE THREE POINTS.               00002840
C                                                                       00002850
```

```
  180 DO  200  INDX = 1,3                                                    00002860
      DO  190  IROW = 1,N                                                    00002870
  190 XN1(IROW) = XN(IROW) + ZZ(INDX) * SN(IROW)                             00002880
      IF (I2.EQ.1)            CALL CGINTG(XN1,YF(INDX),GDUM)                00002890
      IF (I2.EQ.2)            CALL CGLINE(XN1,YF(INDX),GDUM)                00002900
      IF (I1.EQ.1)            CALL OBSOUT                                   00002910
      IF (I3.EQ.1)            CALL PTLOUT                                   00002920
      ICCNT = ICCNT + 1                                                     00002930
  200 CONTINUE                                                              00002940
      WRITE(6,550)                                                          00002950
      WRITE(6,560)                                                          00002960
C                                                                           00002970
C     FIT A QUADRATIC TO THESE THREE POINTS.                                00002980
C                                                                           00002990
  210 CONTINUE                                                              00003000
      A = ZZ(2) - ZZ(3)                                                     00003010
      B = ZZ(3) - ZZ(1)                                                     00003020
      C = ZZ(1) - ZZ(2)                                                     00003030
      D = ZZ(2)**2 - ZZ(3)**2                                               00003040
      E = ZZ(3)**2 - ZZ(1)**2                                               00003050
      F = ZZ(1)**2 - ZZ(2)**2                                               00003060
C                                                                           00003070
C     ANALYTICALLY EVALUATE THE MINIMUM OF THE FITTED CURVE.                00003080
C                                                                           00003090
      ZZT = (D * YF(1) + E * YF(2) + F * YF(3))/2.D0                        00003100
      ZZB =  A * YF(1) + B * YF(2) + C * YF(3)                              00003110
      ZZMIN = ZZT/ZZB                                                       00003120
C                                                                           00003130
C     EVALUATE THE FUNCTION VALUE AT THIS POINT.                            00003140
C                                                                           00003150
      DO  220  IROW = 1,N                                                   00003160
      XN1(IROW) = XN(IROW) + ZZMIN * SN(IROW)                               00003170
  220 CONTINUE                                                              00003180
      IF (I2.EQ.1)            CALL CGINTG(XN1,YFMIN,GDUM)                   00003190
      IF (I2.EQ.2)            CALL CGLINE(XN1,YFMIN,GDUM)                   00003200
      ICCNT = ICCNT + 1                                                     00003210
      WRITE(6,570)  ZZMIN,YFMIN                                             00003220
      IF (I1.EQ.1)            CALL OBSOUT                                   00003230
      IF (I3.EQ.1)            CALL PTLOUT                                   00003240
C                                                                           00003250
C     CHECK TO SEE IF ANY OF THE DESIRED POINTS ARE WITHIN THE              00003260
C     CONVERGENCE CRITERION.                                                00003270
C                                                                           00003280
      DO  250  INDX = 1,3                                                   00003290
      WW(INDX) = DABS(ZZ(INDX) - ZZMIN)                                     00003300
      IF (WW(INDX) - CNVG)    230,230,240                                   00003310
  230 ALFAN = ZZMIN                                                         00003320
      GO TO 400                                                             00003330
  240 CONTINUE                                                              00003340
  250 CONTINUE                                                              00003350
C                                                                           00003360
C     DETERMINE WHICH FUNCTION VALUE IS THE LARGEST AND REPLACE IT WITH     00003370
C     THE INTERPOLATED MINIMUM POINT.                                       00003380
C                                                                           00003390
      IF (YF(2) - YF(1))    260,260,270                                     00003400
  260 IF (YF(3) - YF(1))    280,280,270                                     00003410
  270 IF (YF(3) - YF(2))    290,290,300                                     00003420
```

```
  280 JK = 1                                                              00003430
      GO TO 310                                                           00003440
  290 JK = 2                                                              00003450
      GO TO 310                                                           00003460
  300 JK = 3                                                              00003470
  310 ZZ(JK) = ZZMIN                                                      00003480
      YF(JK) = YFMIN                                                      00003490
C                                                                         00003500
C     FIT A QUADRATIC CURVE TO THE NEW POINTS.                            00003510
C                                                                         00003520
      GO TO 210                                                           00003530
C                                                                         00003540
C     OUTPUT THE CONVERGENCE REPORT.                                      00003550
C                                                                         00003560
  400 CXN1 = YFMIN                                                        00003570
      WRITE(6,580)                                                        00003580
      WRITE(6,590)  ALFAN,CXN1,ICCNT                                      00003590
      DO  410  IROW = 1,N                                                 00003600
      WRITE(6,600)  IROW,XN1(IROW)                                        00003610
  410 CONTINUE                                                            00003620
      WRITE(6,610)                                                        00003630
      STPSZ = STPSV                                                       00003640
      RETURN                                                              00003650
C                                                                         00003660
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * * 00003670
C                                                                         00003680
  500 FORMAT(//32X,69('*')/32X,'*',67X,'*'/32X,'*',10X,'SINGLE VARIAB',  00003690
     1       'LE SEARCH REPORT -- COGGINS METHOD',10X,'*',/32X,'*',67X,  00003700
     2       '*'/32X,69('*'))                                            00003710
C                                                                         00003720
  510 FORMAT(//55X,'INITIALIZATION VALUES'/52X,27('_')//19X,'THE INITI',  00003730
     1       'AL VALUE OF THE INDEPENDENT VARIABLE ALPHA:',10X,'ALPH',   00003740
     2       'A(1) = ',1PD20.10//19X,'THE INITIAL VALUE OF THE STEPSI',  00003750
     3       'ZE',18X,':',7X,'DELTA ALPHA = ',1PD20.10//19X,'THE CONV',  00003760
     4       'ERGENCE LIMIT FOR THE INDEPENDENT VARIABLE :',11X,'EPSI',  00003770
     5       'LON = ',1PD20.10//19X,'THE NUMBER OF PARAMETERS IN THE ',  00003780
     6       'SOLVE-FOR VECTOR X : NO. OF PARAMETERS =     ',I2)         00003790
C                                                                         00003800
  520 FORMAT(//1X,21('*'),10X,'BRACKETING THE FUNCTION MINIMUM -- ',      00003810
     1       'DAVIES, SWANN AND CAMPEY ALGORITHM',10X,21('*'))           00003820
C                                                                         00003830
  530 FORMAT(//11X,'ALPHA',14X,'F(XN + ALPHA * SN)',10X,'DELTA ',         00003840
     1       'ALPHA'/4X,3(20('_'),5X)/2X,3(1PD20.10,5X)//24X,'WHERE:',   00003850
     2       10X,'X(N)',10X,'+',9X,'ALPHA',10X,'*',10X,'S(N)',10X,       00003860
     3       '=',9X,'X(N+1)'/32X,4(20('_'),5X))                          00003870
C                                                                         00003880
  540 FORMAT(30X,4(1PD20.10,5X))                                          00003890
C                                                                         00003900
  550 FORMAT(//1X,19('*'),10X,'FITTING A QUADRATIC FUNCTION TO THE ',     00003910
     1       'BRACKETING POINTS -- POWELL ALGORITHM',10X,19('*')//)      00003920
C                                                                         00003930
  560 FORMAT(31X,'MINIMIZING VALUE OF ALPHA',21X,'MINIMUM FUNCTION ',     00003940
     1       'VALUE'/31X,25('_'),20X,25('_')/)                           00003950
C                                                                         00003960
  570 FORMAT(32X,1PD20.10,25X,1PD20.10)                                   00003970
C                                                                         00003980
  580 FORMAT(///1X,39('*'),10X,'COGGINS METHOD CONVERGENCE REPORT',10X,   00003990
```

```
SERCH1              JSS1493.PRLLEL.ORBDET.FORT

     1       39('*')///)                                                    00004000
C                                                                           00004010
  590 FORMAT(38X,'MINIMIZING VALUE OF ALPHA',11X,':',1PD20.10//38X,         00004020
     1       'MINIMUM COST FUNCTION VALUE',9X,':',1PD20.10//36X,'TOTAL',00004030
     2       ' NUMBER OF FUNCTION EVALUATIONS:   ',I2//52X,'MINIMIZING',00004040
     3       ' SOLVE-FOR VECTOR X'/)                                        00004050
C                                                                           00004060
  600 FORMAT(53X,'X(',I2,'):',1PD20.10)                                     00004070
C                                                                           00004080
  610 FORMAT(///1X,37('*'),10X,'END OF SINGLE VARIABLE SEARCH REPORT',      00004090
     1       10X,38('*'))                                                   00004100
C                                                                           00004110
      END                                                                   00004120

******** END OF MEMBER SERCH1     412 RECORDS ********
```

STAPAR JSS1493.PRLLEL.ORBDET.FORT

```
      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK STAPAR   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/STAPAR/   PXDAEO(6,6)   ,   PXDAKO(6,6)   ,   PXDXO(6,6)  00000120
C                                                                       00000130
C                                                                       00000140
C                     #LOCA-                                            00000150
C             DIMEN-    TION                                            00000160
C  VARIABLE    SION     (REAL)                DESCRIPTION               00000170
C  ________   _______   _______   ____________________________________  00000180
C                                                                       00000190
C    PXDAEO      6,6         1   ARRAY (6X6) CONTAINING THE PARTIAL DE- 00000200
C                                RIVATIVES OF CURRENT POSITION AND VELO-00000210
C                                CITY WITH RESPECT TO EPOCH EQUINOCTIAL 00000220
C                                ELEMENTS                               00000230
C    PXDAKO      6,6        37   ARRAY (6X6) CONTAINING THE PARTIAL DE- 00000240
C                                RIVATIVES OF CURRENT POSITION AND VELO-00000250
C                                CITY WITH RESPECT TO EPOCH KEPLERIAN   00000260
C                                ELEMENTS                               00000270
C     PXDXO      6,6        73   ARRAY (6X6) CONTAINING THE PARTIAL DE- 00000280
C                                RIVATIVES OF CURRENT POSITION AND VELO-00000290
C                                CITY WITH RESPECT TO EPOCH POSITION AND00000300
C                                VELOCITY                               00000310
C                                                                       00000320
      DATA PXDAEO / 36*0.D0      /                                      00000330
C                                                                       00000340
      DATA PXDAKO / 36*0.D0      /                                      00000350
C                                                                       00000360
      DATA PXDXO  / 36*0.D0      /                                      00000370
C                                                                       00000380
      END
```

******** END OF MEMBER STAPAR 38 RECORDS ********

```
STATE                JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK STATE    00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/STATE /   S0(6)    ,   STK(70,6)    ,   STK1(70,6)       00000120
C                                                                       00000130
C                      #LOCA-                                           00000140
C              DIMEN-   TION                                            00000150
C  VARIABLE    SION     (REAL)                 DESCRIPTION              00000160
C  ________    _____    _____   ____________________________________    00000170
C                                                                       00000180
C        S0        6          1   ARRAY (6X1) CONTAINING THE INITIAL CON-00000190
C                                  DITIONS FOR EACH OF THE COMPONENTS OF 00000200
C                                  THE STATE VECTOR TO BE INTEGRATED     00000210
C       STK      70,6          7   ARRAY (70X6) IN WHICH THE ITH COLUMN  00000220
C                                  CONTAINS THE ITH STATE VARIABLE,      00000230
C                                  EVALUATED AT THE N+1 CHEBYSHEV POINTS 00000240
C                                  OF ORTHOGONALITY, FOR THE KTH ITERA-  00000250
C                                  TION OF THE PICARD-CHEBYSHEV METHOD   00000260
C      STK1      70,6        427   ARRAY (70X6) IN WHICH THE ITH COLUMN  00000270
C                                  CONTAINS THE ITH STATE VARIABLE,      00000280
C                                  EVALUATED AT THE N+1 CHEBYSHEV POINTS 00000290
C                                  OF ORTHOGONALITY, FOR ITERATION K+1   00000300
C                                  OF THE PICARD-CHEBYSHEV METHOD        00000310
C                                                                       00000320
      DATA S0     / 6*0.D0      /                                       00000330
C                                                                       00000340
      DATA STK    / 420*0.D0    /                                       00000350
C                                                                       00000360
      DATA STK1   / 420*0.D0    /                                       00000370
C                                                                       00000380
      END                                                               00000390

******** END OF MEMBER STATE       39 RECORDS ********
```

```
TIMER                JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK TIMER   00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/TIMER /   T0      ,     TSPAN     ,     TOUT      ,       00000120
     *                 H       ,     TINC                              00000130
C                                                                       00000140
C                      #LOCA-                                           00000150
C             DIMEN-     TION                                           00000160
C  VARIABLE    SION     (REAL)              DESCRIPTION                 00000170
C  ________   _______   _______   ____________________________________  00000180
C                                                                       00000190
C        T0         1         1   EPOCH TIME OF THE INTEGRATION SPAN    00000200
C     TSPAN         1         2   LENGTH (SEC) OF THE INTEGRATION SPAN  00000210
C      TOUT         1         3   CURRENT OUTPUT TIME                   00000220
C         H         1         4   INTEGRATION STEPSIZE                  00000230
C      TINC         1         5   TIME INCREMENT BETWEEN OUTPUT POINTS  00000240
C                                                                       00000250
      DATA T0     / 0.D0        /                                       00000260
C                                                                       00000270
      DATA TSPAN  / 0.D0        /                                       00000280
C                                                                       00000290
      DATA TOUT   / 0.D0        /                                       00000300
C                                                                       00000310
      DATA H      / 5400.D0     /                                       00000320
C                                                                       00000330
      DATA TINC   / 60.D0       /                                       00000340
C                                                                       00000350
      END                                                               00000360

******** END OF MEMBER TIMER       36 RECORDS ********
```

```
      SUBROUTINE VCOINT(MORD,IEQSET,TL,TU)                                  00000010
C                                                                           00000020
C     VERSION OF 06/15/79     **** CHECKED OUT ****                         00000030
C                                                                           00000040
C     PURPOSE                                                               00000050
C         TO COMPUTE THE INTEGRAL OF A VECTOR OF CHEBYSHEV SERIES           00000060
C         AS FOLLOWS:                                                       00000070
C                                                                           00000080
C             T                T            M                             00000090
C         INT/  F(T) DT = INT/ (A /2 + SUM/  A T (X) ) DX                   00000100
C            T             T   0       K=1  K K                           00000110
C             0            0                                              00000120
C                                        X            M                   00000130
C                       = ((TU-TL)/2) * INT/ (A /2 + SUM/  A T (X) ) DX   00000140
C                                       X   0       K=1  K K              00000150
C                                        0                                  00000160
C                                                                           00000170
C                                              M+1                          00000180
C                       = ((TU-TL)/2) * (B /2 + SUM/   B T (X) )            00000190
C                                       0       K=1   K K                   00000200
C                                                                           00000210
C         THE B'S ARE FOUND FROM THE A'S BY THE FORMULA:                    00000220
C                                                                           00000230
C                   B  = (1/2*L) * (A   - A   );  1.LE.L.LE.M+1             00000240
C                    L               L-1   L+1                              00000250
C                                                                           00000260
C         B  IS ADJUSTED TO SATISFY THE INITIAL CONDITION BCOND = F(T )     00000270
C          0                                                          0     00000280
C                                                  M+1                      00000290
C         ((TU-TL)/2)*B /2 = F(T ) - ((TU-TL)/2)*SUM/  B T (X )             00000300
C                       0       0                  K=1  K K  0              00000310
C                                                                           00000320
C     CALLING SEQUENCE                                                      00000330
C         CALL VCOINT(MORD,IEQSET,TL,TU)                                    00000340
C                                                                           00000350
C     PARAMETERS                                                            00000360
C         MORD       I    MAXIMUM ORDER CHEBYSHEV POLYNOMIAL IN THE        00000370
C                         CHEBYSHEV SERIES TO BE INTEGRATED.                00000380
C         IEQSET     I    FLAG INDICATING THE EQUATION SET BEING           00000390
C                         INTEGRATED BY THE PICARD-CHEBYSHEV                00000400
C                         INTEGRATOR.                                       00000410
C                         IEQSET = 1:  COWELL EQUATIONS                     00000420
C                         IEQSET = 2:  KEPLERIAN VOP EQUATIONS              00000430
C                                      (MEAN ANOMALY)                       00000440
C                         IEQSET = 3:  EQUINOCTIAL VOP EQUATIONS            00000450
C                                      (MEAN LONGITUDE)                     00000460
C                         IEQSET = 4:  EQUINOCTIAL VOP EQUATIONS            00000470
C                                      (ECCENTRIC LONGITUDE)                00000480
C                         IEQSET = 5:  AVERAGED EQUINOCTIAL VOP             00000490
C                                      EQUATIONS (MEAN MEAN LONGITUDE)      00000500
C                         IEQSET = 6:  AVERAGED KEPLERIAN VOP EQUATIONS     00000510
C                                      (MEAN MEAN ANOMALY)                  00000520
C         TL         I    LOWER LIMIT OF INTEGRATION IN THE TRUE            00000530
C                         INDEPENDENT VARIABLE T.                           00000540
C         TU         I    UPPER LIMIT OF INTEGRATION IN THE TRUE            00000550
C                         INDEPENDENT VARIABLE T.                           00000560
C                                                                           00000570
```

```
C     COMMON/COVAL/A(70,6), BC(70,6), FORC(70,6)                        00000580
C         A(70,6)          ARRAY (70X6) IN WHICH THE KTH COLUMN         00000590
C                          CONTAINS THE M+1 COEFFICIENTS OF THE         00000600
C                          CHEBYSHEV SERIES APPROXIMATING THE INTEGRAND  00000610
C                          OF THE KTH EQUATION TO BE INTEGRATED, AS     00000620
C                          SPECIFIED BY THE VALUE OF FLAG IEQSET.       00000630
C         BC(70,6)         ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS 00000640
C                          THE COEFFICIENTS OF THE CHEBYSHEV SERIES     00000650
C                          APPROXIMATING THE INTEGRAL OF THE KTH        00000660
C                          INTEGRAND STORED IN THE KTH COLUMN OF A.     00000670
C         FORC(70,6)       ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS 00000680
C                          THE INTEGRAND OF THE KTH EQUATION TO BE      00000690
C                          INTEGRATED, EVALUATED AT THE N+1 POINTS      00000700
C                          OF ORTHOGONALITY.                            00000710
C                                                                       00000720
C     COMMON/STATE/S0(6), ST(70,6)                                      00000730
C         S0(6)            ARRAY (6X1) CONTAINING THE INITIAL CONDITIONS 00000740
C                          FOR EACH OF THE COMPONENTS OF THE STATE VECTOR00000750
C                          BEING INTEGRATED, AS SPECIFIED BY THE VALUE  00000760
C                          OF THE FLAG IEQSET.                          00000770
C         STK(70,6)        ARRAY (70X6) IN WHICH THE ITH COLUMN CONTAINS 00000780
C                          THE ITH COMPONENT OF THE STATE VECTOR BEING  00000790
C                          INTEGRATED, EVALUATED AT THE N+1 POINTS OF   00000800
C                          ORTHOGONALITY, FOR THE KTH ITERATION OF THE  00000810
C                          PICARD INTEGRATION METHOD.                   00000820
C         STK1(70,6)       ARRAY (70X6) IN WHICH THE ITH COLUMN CONTAINS 00000830
C                          THE ITH COMPONENT OF THE STATE VECTOR BEING  00000840
C                          INTEGRATED, EVALUATED AT THE N+1 CHEBYSHEV   00000850
C                          POINTS, FOR ITERATION K+1 OF THE PICARD      00000860
C                          INTEGRATION METHOD.                          00000870
C                                                                       00000880
C     EXTERNAL REFERENCES                                               00000890
C         CSER1                                                         00000900
C                                                                       00000910
C     ANALYSIS                                                          00000920
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000930
C                                                                       00000940
C     PROGRAMMER                                                        00000950
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000960
C                                                                       00000970
      IMPLICIT REAL*8(A-H,O-Z)                                          00000980
      DIMENSION     B(70)                                               00000990
      COMMON/COVAL /  A(70,6)    ,   BC(70,6)    ,   FORC(70,6)         00001000
      COMMON/STATE /  S0(6)      ,   STK(70,6)   ,   STK1(70,6)         00001010
      M1 = MORD + 1                                                     00001020
      M2 = MORD + 2                                                     00001030
      M3 = MORD + 3                                                     00001040
      M4 = MORD + 4                                                     00001050
      B(1) =  0.D0                                                      00001060
      X    = -1.D0                                                      00001070
C                                                                       00001080
C     BEGIN COMPUTATION FOR THE CASE OF 6 COUPLED FIRST ORDER           00001090
C     DIFFERENTIAL EQUATIONS (VOP EQUATIONS).                           00001100
C                                                                       00001110
      IF (IEQSET.EQ.1) GO TO 50                                         00001120
      DO  20   K = 1,6                                                  00001130
      A(M2,K) = 0.D0                                                    00001140
```

```
      A(M3,K) = 0.D0                                                    00001150
      DO  10   L = 2,M2                                                 00001160
      BC(L,K) = (1.D0/(2.D0*DFLOAT(L-1))) * (A(L-1,K)-A(L+1,K))         00001170
      BC(L,K) = ((TU - TL)/2.D0) * BC(L,K)                              00001180
      B(L)    = BC(L,K)                                                 00001190
   10 CONTINUE                                                          00001200
C                                                                       00001210
C     SET B    SUCH THAT THE KTH INITIAL CONDITION IS SATISFIED.        00001220
C          0,K                                                          00001230
C                                                                       00001240
      CALL CSER1(M1,X,B,SER)                                            00001250
      BC(1,K) = 2.D0 * (S0(K) - SER)                                    00001260
   20 CONTINUE                                                          00001270
      GO TO 90                                                          00001280
C                                                                       00001290
C     BEGIN COMPUTATION FOR THE CASE OF A SECOND ORDER VECTOR           00001300
C     DIFFERENTIAL EQUATION WITH THREE COMPONENTS (COWELL EQUATIONS).    00001310
C     1.  OBTAIN FIRST INTEGRAL                                         00001320
C                                                                       00001330
   50 DO  70   K = 1,3                                                  00001340
      A(M2,K) = 0.D0                                                    00001350
      A(M3,K) = 0.D0                                                    00001360
      DO  60   L = 2,M2                                                 00001370
      BC(L,K) = (1.D0/(2.D0*DFLOAT(L-1))) * (A(L-1,K) - A(L+1,K))       00001380
      BC(L,K) = ((TU - TL)/2.D0) * BC(L,K)                              00001390
      B(L)    = BC(L,K)                                                 00001400
   60 CONTINUE                                                          00001410
C                                                                       00001420
C     SET B    SUCH THAT INITIAL CONDITIONS ON VELOCITY ARE SATISFIED.  00001430
C          0,K                                                          00001440
C                                                                       00001450
      CALL CSER1(M1,X,B,SER)                                            00001460
      BC(1,K) = 2.D0 * (S0(K+3) - SER)                                  00001470
   70 CONTINUE                                                          00001480
C                                                                       00001490
C     2.  OBTAIN SECOND INTEGRAL                                        00001500
C                                                                       00001510
      DO  90   K = 4,6                                                  00001520
      BC(M3,K-3) = 0.D0                                                 00001530
      BC(M4,K-3) = 0.D0                                                 00001540
      DO  80   L = 2,M3                                                 00001550
      BC(L,K) = (1.D0/(2.D0*DFLOAT(L-1)))*(BC(L-1,K-3)-BC(L+1,K-3))     00001560
      BC(L,K) = ((TU - TL)/2.D0) * BC(L,K)                              00001570
      B(L)    = BC(L,K)                                                 00001580
   80 CONTINUE                                                          00001590
C                                                                       00001600
C     SET C    SUCH THAT INITIAL CONDITIONS ON POSITION ARE SATISFIED.  00001610
C          0,K                                                          00001620
C                                                                       00001630
      CALL CSER1(M2,X,B,SER)                                            00001640
      BC(1,K) = 2.D0 * (S0(K-3) - SER)                                  00001650
   90 CONTINUE                                                          00001660
C                                                                       00001670
C     OUTPUT RESULTS                                                    00001680
C                                                                       00001690
C     IF (IEQSET.EQ.1)   GO TO 120                                      00001700
C     DO  110  K = 1,6                                                  00001710
```

```
C     PRINT 200,K                                                        00001720
C     DO  100  L = 1,M2                                                  00001730
C     L1 = L - 1                                                         00001740
C 100 PRINT 205,L1,K,A(L,K),L1,K,BC(L,K)                                 00001750
C 110 CONTINUE                                                           00001760
C     GO TO 150                                                          00001770
C 120 DO  140  K = 1,3                                                   00001780
C     PRINT 210,K                                                        00001790
C     DO  130  L = 1,M3                                                  00001800
C     L1 = L - 1                                                         00001810
C 130 PRINT 215,L1,K,A(L,K),L1,K,BC(L,K),L1,K,BC(L,K+3)                  00001820
C 140 CONTINUE                                                           00001830
  150 RETURN                                                             00001840
C                                                                        00001850
C     FORMAT STATEMENTS                                                  00001860
C                                                                        00001870
  200 FORMAT('1','THE INTEGRAND AND FIRST INTEGRAL SERIES COEFFICIENTS',00001880
     1           ' FOR VOP ELEMENT ',I1,' ARE:'//20X,'INTEGRAND SERIES',00001890
     2           ' COEFFICIENTS',30X,'INTEGRATED SERIES COEFFICIENTS'/) 00001900
C                                                                        00001910
  205 FORMAT(17X,'A(',I2,',',I1,') =',1PD25.15,24X,'B(',I2,',',I1,       00001920
     1                          ') =',1PD25.15)                         00001930
C                                                                        00001940
  210 FORMAT('1','THE SERIES COEFFICIENTS FOR COMPONENT ',I1,' OF THE ',00001950
     1           'ACCELERATION, VELOCITY AND POSITION VECTORS ARE:'//   00001960
     2            5X,'ACCELERATION SERIES COEFFICIENTS',8X,'VELOCITY ', 00001970
     3           'SERIES COEFFICIENTS',11X,'POSITION SERIES COEFFICI',  00001980
     4           'ENTS'/)                                               00001990
C                                                                        00002000
  215 FORMAT(4X,'A(',I2,',',I1,') =',1PD25.15,5X,'B(',I2,',',I1,') =',  00002010
     1           1PD25.15,5X,'C(',I2,',',I1,') =',1PD25.15)             00002020
C                                                                        00002030
      END                                                                00002040
```

******** END OF MEMBER VCOINT 204 RECORDS ********

```
      SUBROUTINE VCOVAL(MORD,NSUB,IEQSET)                                   00000010
C                                                                           00000020
C     VERSION OF 06/15/79    **** CHECKED OUT ****                          00000030
C                                                                           00000040
C     PURPOSE                                                               00000050
C         TO COMPUTE THE COEFFICIENTS IN THE CHEBYSHEV SERIES               00000060
C         APPROXIMATION OF THE INTEGRAND FOR THE ITERATIVE                  00000070
C         PICARD INTEGRATION SCHEME.                                        00000080
C                                                                           00000090
C                                    M                                      00000100
C         NOTE:    F(X) = A /2 + SUM/  A T (X)                            00000110
C                         0       K=1  K K                                  00000120
C                                                                           00000130
C     CALLING SEQUENCE                                                      00000140
C         CALL VCOVAL(MORD,NSUB,IEQSET)                                     00000150
C                                                                           00000160
C     PARAMETERS                                                            00000170
C         MORD       I    MAXIMUM ORDER OF CHEBYSHEV POLYNOMIALS IN THE    00000180
C                         SERIES REPRESENTATION OF THE INTEGRAND.           00000190
C         NSUB       I    SUBSCRIPT OF THE LAST POINT OF ORTHOGONALITY      00000200
C                         IN THE SET X(0) THROUGH X(NSUB).                  00000210
C         IEQSET     I    FLAG INDICATING THE EQUATION SET BEING            00000220
C                         INTEGRATED BY THE CHEBYSHEV-PICARD INTEGRATOR.    00000230
C                         IEQSET = 1:  COWELL EQUATIONS                     00000240
C                         IEQSET = 2:  KEPLERIAN VOP EQUATIONS              00000250
C                                      (MEAN ANOMALY)                       00000260
C                         IEQSET = 3:  EQUINOCTIAL VOP EQUATIONS            00000270
C                                      (MEAN LONGITUDE)                     00000280
C                         IEQSET = 4:  EQUINOCTIAL VOP EQUATIONS            00000290
C                                      (ECCENTRIC LONGITUDE)                00000300
C                         IEQSET = 5:  AVERAGED EQUINOCTIAL VOP EQUATIONS   00000310
C                                      (MEAN MEAN LONGITUDE)                00000320
C                         IEQSET = 6:  AVERAGED KEPLERIAN VOP EQUATIONS     00000330
C                                      (MEAN MEAN ANOMALY)                  00000340
C                                                                           00000350
C     COMMON/CHBINT/T(70), X(70), TX(70,70)                                 00000360
C         T(70)          ARRAY (70X1) CONTAINING THE N+1 ORTHOGONALITY      00000370
C                        POINTS IN THE TRUE INDEPENDENT VARIABLE T.         00000380
C         X(70)          ARRAY (70X1) CONTAINING THE N+1 ORTHOGONALITY      00000390
C                        POINTS IN THE NORMALIZED INDEPENDENT VARIABLE X    00000400
C         TX(70,70)      ARRAY (70X70) CONTAINING THE M+1 CHEBYSHEV         00000410
C                        POLYNOMIALS EVALUATED AT THE N+1 ORTHOGONALITY     00000420
C                        POINTS.                                            00000430
C                                                                           00000440
C     COMMON/COVAL/A(70,6), BC(70,6), FORC(70,6)                            00000450
C         A(70,6)        ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS      00000460
C                        THE M+1 COEFFICIENTS OF THE CHEBYSHEV SERIES       00000470
C                        APPROXIMATING THE KTH COMPONENT OF THE             00000480
C                        STATE VECTOR BEING INTEGRATED (AS SPECIFIED)       00000490
C                        BY THE VALUE OF THE FLAG IEQSET).                  00000500
C         BC(70,6)       ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS      00000510
C                        THE COEFFICIENTS OF THE CHEBYSHEV SERIES           00000520
C                        APPROXIMATING THE INTEGRAL(S) OF THE KTH           00000530
C                        COMPONENT OF THE STATE VECTOR BEING INTEGRATED.    00000540
C         FORC(70,6)     ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS      00000550
C                        THE RIGHT HAND SIDE OF THE KTH DIFFERENTIAL        00000560
C                        EQUATION TO BE INTEGRATED, EVALUATED AT THE        00000570
```

VCOVAL JSS1493.PRLLEL.ORBDET.FORT

```
C                       N+1 ORTHOGONALITY POINTS.                       00000580
C                                                                       00000590
C     EXTERNAL REFERENCES                                               00000600
C         NONE                                                          00000610
C                                                                       00000620
C     ANALYSIS                                                          00000630
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000640
C                                                                       00000650
C     PROGRAMMER                                                        00000660
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000670
C                                                                       00000680
      IMPLICIT REAL*8(A-H,O-Z)                                          00000690
      COMMON/CHBINT/  T(70)    ,  X(70)     ,  TX(70,70)              00000700
      COMMON/COVAL /  A(70,6) ,  BC(70,6)  ,  FORC(70,6)              00000710
C                                                                       00000720
C     COMPUTE COEFFICIENTS                                              00000730
C                                                                       00000740
      M1 = MORD + 1                                                     00000750
      N1 = NSUB + 1                                                     00000760
      IEQ = 3                                                           00000770
      IF (IEQSET.GT.1) IEQ = 6                                          00000780
      DO 40 L = 1,IEQ                                                   00000790
C     PRINT 100,M1,L                                                    00000800
      DO 30 J = 1,M1                                                    00000810
      A(J,L) = (TX(J,1) * FORC(1,L) + TX(J,N1) * FORC(N1,L))/2.D0       00000820
      DO 20 K = 2,NSUB                                                  00000830
   20 A(J,L) = A(J,L) + TX(J,K) * FORC(K,L)                             00000840
      A(J,L) = A(J,L) * (2.D0/FLOAT(NSUB))                              00000850
   30 JMIN1 = J - 1                                                     00000860
C  30 PRINT 101,JMIN1,A(J,L)                                            00000870
   40 CONTINUE                                                          00000880
      RETURN                                                            00000890
C                                                                       00000900
C                         FORMAT STATEMENTS                             00000910
C                                                                       00000920
  100 FORMAT('1'/10X,'THE ',I2,' COEFFICIENTS IN THE CHEBYSHEV SERIES', 00000930
     1       ' APPROXIMATING THE INTEGRAND FOR EQUATION ',I1,' ARE:'//) 00000940
C                                                                       00000950
  101 FORMAT(50X,'A(',I2,') = ',1PD25.15)                               00000960
C                                                                       00000970
      END                                                               00000980
```

******** END OF MEMBER VCOVAL 98 RECORDS ********

```
      SUBROUTINE VMATTX(MORD,NSUB)                                          00000010
C                                                                           00000020
C     VERSION OF 4/4/79       **** CHECKED OUT ****                         00000030
C                                                                           00000040
C     PURPOSE                                                               00000050
C        TO PRODUCE THE MATRIX OF CHEBYSHEV POLYNOMIALS T(0) THROUGH        00000060
C        T(M) EVALUATED AT THE POINTS OF ORTHOGONALITY X(0) THROUGH         00000070
C        X(N) COMPUTED BY THE SUBROUTINE ORTHPT.                            00000080
C                                                                           00000090
C     CALLING SEQUENCE                                                      00000100
C        MATTX(MORD,NSUB)                                                   00000110
C                                                                           00000120
C     PARAMETERS                                                            00000130
C        MORD            I     MAXIMUM ORDER OF CHEBYSHEV POLYNOMIALS      00000140
C                              T(M) TO BE EVALUATED                         00000150
C        NSUB            I     SUBSCRIPT OF LAST ORTHOGONALITY POINT       00000160
C                                                                           00000170
C     COMMON/CHBINT/T(70), X(70), TX(70,70)                                 00000180
C        T(70)                 ARRAY (70X1) CONTAINING THE ORTHOGONALITY    00000190
C                              POINTS IN THE TRUE INDEPENDENT VARIABLE X    00000200
C        X(70)                 ARRAY (70X1) CONTAINING THE ORTHOGONALITY    00000210
C                              POINTS IN THE NORMALIZED INDEPENDENT         00000220
C                              VARIABLE X.                                  00000230
C        TX(70,70)             ARRAY (70X70) CONTAINING THE M+1             00000240
C                              CHEBYSHEV POLYNOMIALS EVALUATED AT THE       00000250
C                              N+1 ORTHOGONALITY POINTS.  THE ARRAY IS      00000260
C                              IN THE FOLLOWING FORMAT:                     00000270
C                                                                           00000280
C                                                                           00000290
C                                                                           00000300
C                 X      X      X     . . .   X     . . .    X             00000310
C                  0      1      2              K-1             N           00000320
C              IIII                                                 IIII    00000330
C        T     I  TX     TX     TX     . . .  TX     . . .    TX        I   00000340
C         0    I    1,1    1,2    1,3           1,K              1,N+1   I   00000350
C              I                                                        I   00000360
C        T     I  TX     TX     TX     . . .  TX     . . .    TX        I   00000370
C         1    I    2,1    2,2    2,3           2,K              2,N+1   I   00000380
C              I                                                        I   00000390
C        T     I  TX     TX     TX     . . .  TX     . . .    TX        I   00000400
C         2    I    3,1    3,2    3,3           3,K              3,N+1   I   00000410
C        .     I    .      .      .             .                .       I   00000420
C        .     I    .      .      .      . . .  .      . . .     .       I   00000430
C        .     I    .      .      .             .                .       I   00000440
C              I                                                        I   00000450
C        T     I  TX     TX     TX     . . .  TX     . . .    TX        I   00000460
C         J-1  I    J,1    J,2    J,3           J,K              J,N+1   I   00000470
C        .     I    .      .      .             .                .       I   00000480
C        .     I    .      .      .      . . .  .      . . .     .       I   00000490
C        .     I    .      .      .             .                .       I   00000500
C              I                                                        I   00000510
C        T     I  TX     TX     TX     . . .  TX     . . .    TX        I   00000520
C         M    I    M+1,1  M+1,2  M+1,3         M+1,K            M+1,N+1I   00000530
C              I                                                        I   00000540
C              IIII                                                 IIII    00000550
C                                                                           00000560
C                                                                           00000570
```

```
C     EXTERNAL REFERENCES                                                    00000580
C         NONE                                                               00000590
C                                                                            00000600
C     ANALYSIS                                                               00000610
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00000620
C                                                                            00000630
C     PROGRAMMER                                                             00000640
C         CAPTAIN JEFFREY S. SHAVER, USAF                                    00000650
C                                                                            00000660
      IMPLICIT REAL*8(A-H,O-Z)                                               00000670
      COMMON/CHBINT/T(70), X(70), TX(70,70)                                  00000680
C                                                                            00000690
C     BEGIN EVALUATION OF THE POLYNOMIALS FOR ONE VALUE OF X                 00000700
C                                                                            00000710
      NSUB1 = NSUB + 1                                                       00000720
      MORD1 = MORD + 1                                                       00000730
      DO 30 KIN = 1,NSUB1                                                    00000740
C                                                                            00000750
C     INITIALIZE THE THREE TERM RECURRENCE AT POINT X(KIN)                   00000760
C                                                                            00000770
      TX(1,KIN) = 1.D0                                                       00000780
      TX(2,KIN) = X(KIN)                                                     00000790
C                                                                            00000800
C     USE RECURRENCE TO SOLVE FOR REMAINING ELEMENTS IN COLUMN               00000810
C                                                                            00000820
      DO 30 JIN = 3,MORD1                                                    00000830
      TX(JIN,KIN) = 2.D0 * X(KIN) * TX(JIN-1,KIN) - TX(JIN-2,KIN)           00000840
   30 CONTINUE                                                               00000850
C                                                                            00000860
C     PREPARE OUTPUT OF RESULTS, ONE COLUMN PER PAGE                         00000870
C                                                                            00000880
C     DO 40 KIN = 1,NSUB1                                                    00000890
C     KIN1 = KIN - 1                                                         00000900
C     PRINT 300,KIN,KIN1,X(KIN)                                              00000910
C     DO 40 JIN = 1,MORD1                                                    00000920
C     JIN1 = JIN - 1                                                         00000930
C  40 PRINT 301,JIN1,KIN1,TX(JIN,KIN)                                        00000940
C                                                                            00000950
C                              FORMAT STATEMENTS                             00000960
C                                                                            00000970
  300 FORMAT('1',30X,'COLUMN ',I2,' OF MATRIX TX IS GIVEN BELOW FOR ',       00000980
     1               'X(',I2,') = ',1PD25.15//)                              00000990
C                                                                            00001000
  301 FORMAT(50X,'T(',I2,') AT POINT X(',I2,') = ',1PD25.15)                 00001010
C                                                                            00001020
      RETURN                                                                 00001030
      END                                                                    00001040
```

******** END OF MEMBER VMATTX 104 RECORDS ********

```
      SUBROUTINE VORTPT(NSUB,TL,TU)                                     00000010
C                                                                       00000020
C     VERSION OF 4/4/79       **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         COMPUTES THE N+1 POINTS X(0) TO X(N) AT WHICH THE             00000060
C         CHEBYSHEV POLYNOMIALS OF DEGREE M POSSESS THE PROPERTY        00000070
C         OF ORTHOGONALITY WITH RESPECT TO SUMMATION.                   00000080
C                                                                       00000090
C     CALLING SEQUENCE                                                  00000100
C         CALL VORTPT(NSUB,TL,TU)                                       00000110
C                                                                       00000120
C     PARAMETERS                                                        00000130
C         NSUB      I    SUBSCRIPT N OF FINAL POINT X(N), N<70        00000140
C         TL        I    LOWER LIMIT OF INTEGRATION IN TRUE           00000150
C                        INDEPENDENT VARIABLE.                        00000160
C         TU        I    UPPER LIMIT OF INTEGRATION IN TRUE           00000170
C                        INDEPENDENT VARIABLE.                        00000180
C                                                                       00000190
C     COMMON/CHBINT/T(70), X(70), TX(70,70)                             00000200
C         T(70)          ARRAY (70X1) CONTAINING THE ORTHOGONALITY     00000210
C                        POINTS IN THE TRUE INDEPENDENT VARIABLE T.    00000220
C         X(70)          ARRAY (70X1) CONTAINING THE ORTHOGONALITY     00000230
C                        POINTS IN THE NORMALIZED INDEPENDENT VARIABLE X00000240
C         TX(70,70)      ARRAY (70X70) CONTAINING THE M+1 CHEBYSHEV    00000250
C                        POLYNOMIALS EVALUATED AT THE N+1 ORTHOGONALITY00000260
C                        POINTS.                                      00000270
C                                                                       00000280
C     EXTERNAL REFERENCES                                               00000290
C         NONE                                                          00000300
C                                                                       00000310
C     ANALYSIS                                                          00000320
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000330
C                                                                       00000340
C     PROGRAMMER                                                        00000350
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000360
C                                                                       00000370
C                                                                       00000380
      IMPLICIT REAL*8(A-H,O-Z)                                          00000390
      COMMON/CHBINT/T(70), X(70), TX(70,70)                             00000400
C                                                                       00000410
C     COMPUTE THE ORTHOGONAL POINTS                                     00000420
C                                                                       00000430
      PI = 3.141592653589793D0                                          00000440
      NSUB1 = NSUB + 1                                                  00000450
      DO 10 K = 1,NSUB1                                                 00000460
      TTOP = FLOAT(K) - 1.D0                                            00000470
      TBOT = FLOAT(NSUB)                                                00000480
      X(K) = -DCOS((TTOP/TBOT)*PI)                                      00000490
      T(K) = ((X(K) + 1.D0) * (TU - TL))/2.D0 + TL                      00000500
   10 CONTINUE                                                          00000510
C                                                                       00000520
C     OUTPUT RESULTS                                                    00000530
C                                                                       00000540
C     PRINT 200,NSUB1                                                   00000550
C     DO 20 K = 1,NSUB1                                                 00000560
C     K1 = K - 1                                                        00000570
```

```
VORTPT              JSS1493.PRLLEL.ORBDET.FORT

C  20 PRINT 201,K1,X(K),K1,T(K)                                          00000580
C                                                                        00000590
C                              FORMAT STATEMENTS                         00000600
C                                                                        00000610
  200 FORMAT('1',' THE ',I2,' DISCRETE POINTS AT WHICH THE CHEBYSHEV',   00000620
     1           ' POLYNOMIALS ARE ORTHOGONAL ARE GIVEN BELOW:'//)       00000630
C                                                                        00000540
  201 FORMAT(29X,'X(',I2,') =',1PD25.15,10X,'T(',I2,') =',1PD25.15)      00000650
C                                                                        00000660
      RETURN                                                             00000670
      END                                                                00000680

******** END OF MEMBER VORTPT      68 RECORDS ********
```

```
      SUBROUTINE VTEST(TL,TU,GM)                                        00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INTEGRATE VIA THE METHOD OF PICARD ITERATION A VECTOR      00000060
C         OF FIRST ORDER DIFFERENTIAL EQUATIONS OF THE FORM:            00000070
C                                                                       00000080
C                 DY/DX = F(X,Y(X));  Y(X ) = Y                         00000090
C                                       0     0                         00000100
C                                                                       00000110
C                                  OR                                   00000120
C                                                                       00000130
C              DYDOT/DX = F(X,Y(X),YDOT(X));  YDOT(X ) = YDOT           00000140
C                                                    0        0         00000150
C                                                                       00000160
C                 DY/DX = YDOT(X);  Y(X ) = Y                           00000170
C                                     0     0                           00000180
C                                                                       00000190
C         WHERE THE RIGHT HAND SIDES OF THESE EQUATIONS ARE APPROXIMATED00000200
C         BY APPROPRIATE SERIES OF CHEBYSHEV POLYNOMIALS.               00000210
C                                                                       00000220
C     METHOD                                                            00000230
C         ONCE TRANSFORMED TO INTEGRAL FORM, THESE EQUATIONS ARE SOLVED 00000240
C         BY SUCCESSIVE APPROXIMATION FOR Y(X) IN THE FOLLOWING MANNER: 00000250
C                                                                       00000260
C                                    X                                  00000270
C             Y   (X) = Y(X ) + INT/  F(S,Y (S))DS                      00000280
C              K+1          0       X       K                          00000290
C                                    0                                  00000300
C                     OR                                                00000310
C                                                                       00000320
C                                    X       .                          00000330
C           YDOT   (X) = YDOT(X ) + INT/  F(S,Y (S),YDOT (S))DS         00000340
C               K+1             0       X       K        K             00000350
C                                       0                               00000360
C                                                                       00000370
C                                    X                                  00000380
C             Y   (X) = Y(X ) + INT/  YDOT   (S)DS                      00000390
C              K+1          0       X      K+1                         00000400
C                                    0                                  00000410
C                                                                       00000420
C                         WHERE S IS A DUMMY VARIABLE OF INTEGRATION    00000430
C                         AND THE SUBSCRIPT ON Y AND YDOT REPRESENTS    00000440
C                         THE ITERATION NUMBER.                         00000450
C                                                                       00000460
C         THE INTEGRANDS ABOVE ARE APPROXIMATED BY SERIES OF            00000470
C         CHEBYSHEV POLYNOMIALS SUCH THAT:                              00000480
C                                                                       00000490
C                                              M                        00000500
C                         F(X,Y(X)) = A /2 + SUM/  A T (X)              00000510
C                                     0       J=1  J J                  00000520
C                                                                       00000530
C                         WHERE THE A'S ARE COEFFICIENTS AND THE T'S    00000540
C                         ARE CHEBYSHEV POLYNOMIALS.                    00000550
C                                                                       00000560
C         THE COEFFICIENTS ARE FOUND BY APPLYING THE PROPERTY OF        00000570
```

```
C       ORTHOGONALITY OVER SUMMATION POSSESSED BY CHEBYSHEV                00000580
C       POLYNOMIALS AT THE N+1 POINTS X(0) TO X(N) GIVEN BY:               00000590
C                                                                          00000600
C                        X  = COS (I*PI/N);   I = 0,1,2,...,N              00000610
C                         I                                                00000620
C                                                                          00000630
C       THE COEFFICIENTS ARE GIVEN BY THE FOLLOWING:                       00000640
C                              N                                           00000650
C            A  = 2/N * SUM"/  F(X ,Y(X )) * T (X );  L = 0,1,...,M        00000660
C             L            I=0    I    I      L  I                        00000670
C                                                                          00000680
C                      WHERE THE DOUBLE PRIME (") NOTATION ON THE          00000690
C                      SUMMATION INDICATES THAT THE FIRST (I=0) AND        00000700
C                      THE LAST (I=N) TERMS ARE TO BE FACTORED BY          00000710
C                      ONE HALF.                                           00000720
C                                                                          00000730
C     CALLING SEQUENCE                                                     00000740
C         CALL VTEST(TL,TU,GM)                                             00000750
C                                                                          00000760
C     PARAMETERS                                                           00000770
C     FROM CALLING SEQUENCE:                                               00000780
C         TL          I    LOWER LIMIT OF INTEGRATION IN THE TRUE          00000790
C                          INDEPENDENT VARIABLE T.                         00000800
C         TU          I    UPPER LIMIT OF INTEGRATION IN THE TRUE          00000810
C                          INDEPENDENT VARIABLE T.                         00000820
C         GM          I    GRAVITATIONAL PARAMETER.                        00000830
C                                                                          00000840
C     COMMON/CHBINT/ T(70), X(70), TX(70,70)                               00000850
C         T(70)            ARRAY (70X1) CONTAINING THE ORTHOGONALITY       00000860
C                          POINTS IN THE TRUE INDEPENDENT VARIABLE T.      00000870
C         X(70)            ARRAY (70X1) CONTAINING THE ORTHOGONALITY       00000880
C                          POINTS IN  NORMALIZED INDEPENDENT VARIABLE X.   00000890
C         TX(70,70)        ARRAY (70X70) CONTAINING THE M+1 CHEBYSHEV      00000900
C                          POLYNOMIALS EVALUATED AT THE N+1 POINTS         00000910
C                          OF ORTHOGONALITY.                               00000920
C                                                                          00000930
C     COMMON/CNVRG / CON(6), CONKEP                                        00000940
C         CON(6)           ARRAY (6X1) CONTAINING THE CONVERGENCE          00000950
C                          CRITERION FOR EACH OF THE STATE EQUATIONS TO BE 00000960
C                          INTEGRATED.                                     00000970
C         CONKEP           CONVERGENCE CRITERION TO BE USED BY THE         00000980
C                          SUBROUTINE KEPLR IN SOLVING FOR ECCENTRIC       00000990
C                          (OR TRUE) ANOMALY BY NEWTON'S METHOD.           00001000
C                                                                          00001010
C     COMMON/COVAL / A(70,6), BC(70,6), FORC(70,6)                         00001020
C         A(70,6)          ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS   00001030
C                          THE M+1 COEFFICIENTS OF THE CHEBYSHEV SERIES    00001040
C                          APPROXIMATING THE INTEGRAND OF THE KTH          00001050
C                          EQUATION TO BE INTEGRATED (AS SPECIFIED BY THE  00001060
C                          VALUE OF THE FLAG IEQSET).                      00001070
C         BC(70,6)         ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS   00001080
C                          THE COEFFICIENTS OF THE CHEBYSHEV SERIES        00001090
C                          APPROXIMATING THE INTEGRAL OF THE KTH INTEGRAND 00001100
C                          WHOSE COEFFICIENTS WERE STORED IN THE KTH       00001110
C                          COLUMN OF MATRIX A.                             00001120
C         FORC(70,6)       ARRAY (70X6) IN WHICH THE KTH COLUMN CONTAINS   00001130
C                          THE INTEGRAND OF THE KTH EQUATION TO BE         00001140
```

```
C                       INTEGRATED, EVALUATED AT THE N+1 POINTS OF      00001150
C                       ORTHOGONALITY.                                  00001160
C                                                                       00001170
C     COMMON/NPCNTL/ MORD, NSUB, IEQSET, ITERM                          00001180
C         IEQSET        FLAG INDICATING THE SET OF FIRST ORDER          00001190
C                       DIFFERENTIAL EQUATIONS TO BE INTEGRATED         00001200
C                       BY THE PICARD-CHEBYSHEV INTEGRATOR.             00001210
C                       IEQSET = 1:  COWELL EQUATIONS                   00001220
C                       IEQSET = 2:  KEPLERIAN VOP EQUATIONS            00001230
C                       IEQSET = 3:  EQUINOCTIAL VOP EQUATIONS          00001240
C         ITERM         TERMINAL ITERATION, SPECIFYING THE MAXIMUM      00001250
C                       NUMBER OF ITERATIONS ALLOWED BEFORE THE         00001260
C                       INTEGRATION PROCESS WILL BE TERMINATED.         00001270
C         M             MAXIMUM ORDER OF CHEBYSHEV POLYNOMIAL IN THE    00001280
C                       SERIES REPRESENTATION OF THE INTEGRAND (M<69).  00001290
C         N             SUBSCRIPT OF FINAL ORTHOGONALITY POINT (N<70).  00001300
C                                                                       00001310
C     COMMON/STATE / S0(6), STK(70,6), STK1(70,6)                       00001320
C         S0(6)         ARRAY (6X1) CONTAINING THE INITIAL CONDITIONS   00001330
C                       FOR EACH OF THE COMPONENTS OF THE STATE VECTOR  00001340
C                       TO BE INTEGRATED.                               00001350
C         STK(70,6)     ARRAY (70X6) IN WHICH THE ITH COLUMN CONTAINS   00001360
C                       THE ITH STATE VARIABLE, EVALUATED AT THE N+1    00001370
C                       POINTS OF ORTHOGONALITY, FOR THE KTH ITERATION  00001380
C                       OF THE PICARD METHOD.                           00001390
C         STK1(70,6)    ARRAY (70X6) IN WHICH THE ITH COLUMN CONTAINS   00001400
C                       THE ITH STATE VARIABLE, EVALUATED AT THE N+1    00001410
C                       POINTS OF ORTHOGONALITY, FOR ITERATION K+1      00001420
C                       OF THE PICARD METHOD.                           00001430
C                                                                       00001440
C     COMMON/TIMER / T0, TSPAN, TOUT, H, TINC                           00001450
C         T0            EPOCH TIME OF THE INTEGRATION SPAN              00001460
C         TSPAN         LENGTH (SEC) OF THE INTEGRATION SPAN            00001470
C         TOUT          CURRENT OUTPUT TIME                             00001480
C         H             INTEGRATION STEPSIZE                            00001490
C         TINC          TIME INCREMENT BETWEEN OUTPUT POINTS            00001500
C                                                                       00001510
C     EXTERNAL REFERENCES                                               00001520
C         AKTOAE,CSER1,ELMRPT,ELRATE,EPELM,EQELPV,FORCE,F2BDY,KEPLR,    00001530
C         LGRNGE,VCOINT,VCOVAL,VMATTX,VORTPT                            00001540
C                                                                       00001550
C     ANALYSIS                                                          00001560
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00001570
C                                                                       00001580
C     PROGRAMMER                                                        00001590
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00001600
C                                                                       00001610
      IMPLICIT REAL*8(A-H,O-Z)                                          00001620
C                                                                       00001630
C * * * * * * * * *       DIMENSION STATEMENTS      * * * * * * * * * * 00001640
C                                                                       00001650
      DIMENSION      VPELMT(6) ,  POSVEL(6) ,  SCDUM(6)                 00001660
C                                                                       00001670
      DIMENSION      STA(6)    ,  RHS(6)    ,  EL(26)                   00001680
C                                                                       00001690
C * * * * * * * * *         LOGICAL VARIABLES       * * * * * * * * * * 00001700
C                                                                       00001710
```

```
      LOGICAL        CONK   , CONK1  , CNV(6)                              00001720
C                                                                           00001730
C * * * * * * * * *          COMMON BLOCKS        * * * * * * * * * * * * 00001740
C                                                                           00001750
      COMMON/CHBINT/ T(70)    , X(70)     , TX(70,70)                       00001760
C                                                                           00001770
      COMMON/CNVRG / CON(6)   , CONKEP                                      00001780
C                                                                           00001790
      COMMON/COVAL / A(70,6)  , BC(70,6)  , FORC(70,6)                      00001800
C                                                                           00001810
      COMMON/ICNTRL/ IRNTYP   , IVINIT    , IVINTG     , ISLVFR    ,        00001820
     1               NSLVFR   , IPTLS     , IJ2RTS     , IRVPAR    ,        00001830
     2               IPARRV   , IPRNT1    , IPRNT2     , IPRNT3    ,        00001840
     3               IPRNT4   , IPRNT5    , IOUT       , ICGCMP    ,        00001850
     4               ISCID    , NPAR                                       00001860
C                                                                           00001870
      COMMON/NPCNTL/ MORD     , NSUB      , IEQSET     , ITERM              00001880
C                                                                           00001890
      COMMON/STATE / S0(6)    , STK(70,6) , STK1(70,6)                      00001900
C                                                                           00001910
      COMMON/TIMER / T0       , TSPAN     , TOUT       , H         ,        00001920
     1               TINC                                                  00001930
                                                                            00001940
C                                                                           00001950
C * * * * * * * * *     EQUIVALENCE STATEMENTS    * * * * * * * * * * * * 00001960
C                                                                           00001970
C     FROM COMMON/NPCNTL/:                                                  00001980
      EQUIVALENCE    (MORD,M) , (NSUB,N)                                    00001990
C                                                                           00002000
C     ERROR CHECK INPUT PARAMETERS                                          00002010
C                                                                           00002020
      IF (IEQSET.LT.1.OR.IEQSET.GT.3)   GO TO 900                           00002030
      IF (ITERM.LT.2)                   GO TO 905                           00002040
      IF (M.GE.68)                      GO TO 910                           00002050
      IF (N.GE.70)                      GO TO 915                           00002060
      IF (TU.LT.TL)                     GO TO 920                           00002070
C                                                                           00002080
C     INITIALIZE VARIABLES, M1, M2, M3, N1, ISET, ICOUNT,                   00002090
C     CONK, AND CONK1                                                       00002100
C                                                                           00002110
      M1 = M + 1                                                            00002120
      M2 = M + 2                                                            00002130
      M3 = M + 3                                                            00002140
      N1 = N + 1                                                            00002150
      ISET   = 1                                                            00002160
      ICOUNT = 0                                                            00002170
      CONK   = .FALSE.                                                      00002180
      CONK1  = .FALSE.                                                      00002190
C                                                                           00002200
C     COMPUTE THE MATRIX OF CHEBYSHEV POLYNOMIALS T(0) THROUGH T(M) AT      00002210
C     THE N+1 POINTS OF ORTHOGONALITY X(0) THROUGH X(N).                    00002220
C                                                                           00002230
      CALL VORTPT(N,TL,TU)                                                  00002240
      CALL VMATTX(M,N)                                                      00002250
C                                                                           00002260
C     BEGIN COMPUTATION FOR COWELL EQUATIONS:  DETERMINE EPOCH              00002270
C     KEPLERIAN ELEMENTS                                                    00002280
```

```
C                                                                       00002290
      IF (IEQSET.GT.1)                  GO TO 200                       00002300
      IF (IPRNT3.EQ.1)                  WRITE(6,665)                    00002310
      CALL EPELM(SO,TL,GM,ISET,IPRNT3,EL)                               00002320
C                                                                       00002330
C     EVALUATE RIGHT-HAND MEMBERS FOR THE ZEROTH ITERATION              00002340
C                                                                       00002350
      DO  20   I = 1,N1                                                 00002360
      AM = EL(7) * (T(I) - EL(6))                                       00002370
      CALL KEPLR(AM,EL(2),CONKEP,AE,AF)                                 00002380
      CALL LGRNGE(SO,EL(10),GM,AF,EL(8),EL(13),STA)                     00002390
      CALL F2BDY(STA,GM,RHS)                                            00002400
      DO  10   KCOMP = 1,6                                              00002410
      STK(I,KCOMP)  = STA(KCOMP)                                        00002420
   10 FORC(I,KCOMP) = RHS(KCOMP)                                        00002430
   20 CONTINUE                                                          00002440
C                                                                       00002450
C     COMPUTE COEFFICIENTS FOR THE CHEBYSHEV SERIES FOR KTH             00002460
C     ITERATION OF THE RIGHT-HAND MEMBERS.                              00002470
C                                                                       00002480
   30 CALL VCOVAL(M,N,IEQSET)                                           00002490
C                                                                       00002500
C     COMPUTE COEFFICIENTS FOR THE INTEGRATED SERIES                    00002510
C                                                                       00002520
      ICOUNT = ICOUNT + 1                                               00002530
      DO  40   KCOMP = 1,6                                              00002540
   40 CNV(KCOMP) = .TRUE.                                               00002550
      CALL VCOINT(M,IEQSET,TL,TU)                                       00002560
      IF (IEQSET.GT.1)                  GO TO 270                       00002570
C                                                                       00002580
C     EVALUATE THE POSITION AND VELOCITY SERIES AT THE CHEBYSHEV POINTS 00002590
C     FOR ITERATION K+1; COMPUTE THE RELATIVE ERROR AT EACH POINT.      00002600
C                                                                       00002610
      DO  70   KCOMP = 1,3                                              00002620
      DO  60   I = 1,N1                                                 00002630
      CALL CSER1(M2,X(I),BC(1,KCOMP+3),STK1(I,KCOMP))                   00002640
      IF (STK(I,KCOMP).EQ.0.D0)         GO TO 50                        00002650
      RELERR = DABS((STK(I,KCOMP) - STK1(I,KCOMP))/STK(I,KCOMP))        00002660
      IF (RELERR.GT.CON(KCOMP)/2.D0)    CNV(KCOMP) = .FALSE.            00002670
   50 CALL CSER1(M2,X(I),BC(1,KCOMP),STK1(I,KCOMP+3))                   00002680
      IF (STK(I,KCOMP+3).EQ.0.D0)       GO TO 60                        00002690
      RELERR = DABS((STK(I,KCOMP+3) - STK1(I,KCOMP+3))/STK(I,KCOMP+3)) 00002700
      IF (RELERR.GT.CON(KCOMP+3)/2.D0)  CNV(KCOMP+3) = .FALSE.          00002710
   60 CONTINUE                                                          00002720
   70 CONTINUE                                                          00002730
C                                                                       00002740
C     USE NEW VALUES OF POSITION AND VELOCITY TO DETERMINE NEW          00002750
C     VALUES OF ACCELERATION FOR NEXT ITERATION.                        00002760
C                                                                       00002770
      DO 100   I = 1,N1                                                00002780
      DO  80   KCOMP = 1,6                                              00002790
   80 STA(KCOMP) = STK1(I,KCOMP)                                        00002800
      CALL FORCE(STA,GM,RHS)                                            00002810
      DO  90   KCOMP = 1,3                                              00002820
      FORC(I,KCOMP)   = RHS(KCOMP)                                      00002830
   90 FORC(I,KCOMP+3) = 0.D0                                            00002840
  100 CONTINUE                                                          00002850
```

```
C                                                                       00002860
C     CHECK FOR CONVERGENCE ON THIS ITERATION                           00002870
C                                                                       00002880
  110 CONK1 = CNV(1).AND.CNV(2).AND.CNV(3).AND.                         00002890
     1        CNV(4).AND.CNV(5).AND.CNV(6)                              00002900
C                                                                       00002910
C     IF CONVERGED, OUTPUT PICARD CONVERGENCE REPORT                    00002920
C                                                                       00002930
      IF (CONK.AND.CONK1.AND.IEQSET.EQ.1)    GO TO 150                  00002940
      IF (CONK.AND.CONK1.AND.IEQSET.GT.1)    GO TO 340                  00002950
C                                                                       00002960
C     OTHERWISE, ITERATION K+1 RESULTS BECOME ITERATION K RESULTS       00002970
C                                                                       00002980
      CONK = CONK1                                                      00002990
      DO 140  KCOMP = 1,6                                               00003000
      DO 130  I = 1,N1                                                  00003010
  130 STK(I,KCOMP) = STK1(I,KCOMP)                                      00003020
  140 CONTINUE                                                          00003030
C                                                                       00003040
C     IF THE TERMINAL ITERATION NUMBER HAS NOT BEEN EXCEEDED, RETURN TO 00003050
C     30 AND ITERATE AGAIN; OTHERWISE OUTPUT NON-CONVERGENCE REPORT.    00003060
C                                                                       00003070
      IF (ICOUNT.LT.ITERM)              GO TO 30                        00003080
      GO TO 500                                                         00003090
C                                                                       00003100
C     PREPARE AND OUTPUT COWELL CONVERGENCE REPORT                      00003110
C                                                                       00003120
  150 IF (IPRNT3.EQ.0)                  RETURN                          00003130
      WRITE(6,600)                                                      00003140
      WRITE(6,650)                                                      00003150
      WRITE(6,605)  M                                                   00003160
      WRITE(6,610)  N1                                                  00003170
      WRITE(6,670)  H                                                   00003180
      WRITE(6,640)  CON(4),CON(5),CON(6),CON(1),CON(2),CON(3)           00003190
      ILAST = ICOUNT + 1                                                00003200
      WRITE(6,645)  ILAST,ITERM                                         00003210
      DO 170  KCOMP = 1,3                                               00003220
      WRITE(6,625)  KCOMP                                               00003230
      DO 160   K = 1,M3                                                 00003240
      K1 = K - 1                                                        00003250
  160 WRITE(6,630)  K1,KCOMP,A(K,KCOMP),K1,KCOMP,BC(K,KCOMP),           00003260
     1              K1,KCOMP,BC(K,KCOMP+3)                             00003270
  170 CONTINUE                                                          00003280
      WRITE(6,615)                                                      00003290
      RETURN                                                            00003300
C                                                                       00003310
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00003320
C                                                                       00003330
C     BEGIN COMPUTATION FOR VOP EQUATIONS:  DETERMINE EPOCH STATE       00003340
C                                                                       00003350
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00003360
C                                                                       00003370
  200 IF (IEQSET.EQ.3)                  ISET = 2                        00003380
      IF (IPRNT3.EQ.1)                  WRITE(6,665)                    00003390
      IF (IEQSET.EQ.3)                  GO TO 210                       00003400
      CALL AKTOAE(S0,RETRO,S0DUM)                                       00003410
      CALL EQELPV(S0DUM,RETRO,GM,POSVEL)                                00003420
```

```
      GO TO 220                                                          00003430
  210 RETRO = 1.D0                                                       00003440
      CALL EQELPV(S0,RETRO,GM,POSVEL)                                    00003450
  220 CALL EPELM(POSVEL,TL,GM,ISET,IPRNT3,EL)                            00003460
C                                                                        00003470
C     EVALUATE RIGHT-HAND MEMBERS FOR THE ZEROTH ITERATION               00003480
C                                                                        00003490
      DO  260  I = 1,N1                                                  00003500
      AM = EL(23) + EL(7) * (T(I) - TL)                                  00003510
      IF (ISET.EQ.2)                    GO TO 230                        00003520
      FSTVAR = AM                                                        00003530
      GO TO 240                                                          00003540
  230 FSTVAR = AM + EL(11) + EL(13) * EL(12)                             00003550
  240 DO  250  KCOMP = 1,5                                               00003560
      STK(I,KCOMP)  = S0(KCOMP)                                          00003570
      FORC(I,KCOMP) = 0.D0                                               00003580
  250 CONTINUE                                                           00003590
      STK(I,6)  = FSTVAR                                                 00003600
      FORC(I,6) = EL(7)                                                  00003610
  260 CONTINUE                                                           00003620
C                                                                        00003630
C     COMPUTE COEFFICIENTS FOR THE CHEBYSHEV SERIES FOR THE KTH          00003640
C     ITERATION OF THE ELEMENT RATES; INTEGRATE SERIES.                  00003650
C                                                                        00003660
      GO TO 30                                                           00003670
C                                                                        00003680
C     EVALUATE THE VOP ELEMENT SERIES AT THE CHEBYSHEV POINTS FOR        00003690
C     ITERATION K+1; COMPUTE THE RELATIVE ERROR.                         00003700
C                                                                        00003710
  270 DO  290  KCOMP = 1,6                                               00003720
      DO  280  I = 1,N1                                                  00003730
      CALL CSER1(M1,X(I),BC(1,KCOMP),STK1(I,KCOMP))                      00003740
      RELERR = DABS((STK(I,KCOMP) - STK1(I,KCOMP))/STK(I,KCOMP))        00003750
      IF (RELERR.GT.CON(KCOMP)/2.D0)    CNV(KCOMP) = .FALSE.             00003760
  280 CONTINUE                                                           00003770
  290 CONTINUE                                                           00003780
C                                                                        00003790
C     USE NEW VALUES OF THE ELEMENTS TO EVALUATE THE ELEMENT RATES FOR   00003800
C     THE NEXT ITERATION.                                                00003810
C                                                                        00003820
      DO  320  I = 1,N1                                                  00003830
      DO  300  KCOMP = 1,6                                               00003840
  300 VPELMT(KCOMP) = STK1(I,KCOMP)                                      00003850
      CALL ELRATE(VPELMT,RETRO,GM,RHS)                                   00003860
      DO  310  KCOMP = 1,6                                               00003870
  310 FORC(I,KCOMP) = RHS(KCOMP)                                         00003880
  320 CONTINUE                                                           00003890
C                                                                        00003900
C     CHECK FOR CONVERGENCE ON THIS ITERATION.                           00003910
C                                                                        00003920
      GO TO 110                                                          00003930
C                                                                        00003940
C     PREPARE AND OUTPUT VOP CONVERGENCE REPORT.                         00003950
C                                                                        00003960
  340 IF (IPRNT3.EQ.0)                  RETURN                           00003970
      WRITE(6,600)                                                       00003980
      IF (IEQSET.EQ.2)                  WRITE(6,655)                     00003990
```

```
      IF (IEQSET.EQ.3)                  WRITE(6,660)                          00004000
      WRITE(6,605)  M                                                         00004010
      WRITE(6,610)  N1                                                        00004020
      WRITE(6,670)  H                                                         00004030
      WRITE(6,640)  (CON(KCOMP),KCOMP=1,6)                                    00004040
      ILAST = ICOUNT + 1                                                      00004050
      WRITE(6,645)  ILAST,ITERM                                               00004060
      DO  360  KCOMP = 1,6                                                    00004070
      WRITE(6,620)  KCOMP                                                     00004080
      DO  350  K = 1,M2                                                       00004090
      K1 = K - 1                                                              00004100
  350 WRITE(6,690)  K1,KCOMP,A(K,KCOMP),K1,KCOMP,BC(K,KCOMP)                  00004110
  360 CONTINUE                                                                00004120
      WRITE(6,615)                                                            00004130
      RETURN                                                                  00004140
C                                                                             00004150
C     PREPARE AND OUTPUT NON-CONVERGENCE REPORT                               00004160
C                                                                             00004170
  500 WRITE(6,635)  ICOUNT                                                    00004180
      IF (IEQSET.EQ.1)                  WRITE(6,650)                          00004190
      IF (IEQSET.EQ.2)                  WRITE(6,655)                          00004200
      IF (IEQSET.EQ.3)                  WRITE(6,660)                          00004210
      WRITE(6,605)  M                                                         00004220
      WRITE(6,610)  N1                                                        00004230
      RETURN                                                                  00004240
C                                                                             00004250
C     ERROR RETURN EXITS                                                      00004260
C                                                                             00004270
  900 WRITE(6,800)                                                            00004280
      RETURN                                                                  00004290
  905 WRITE(6,805)                                                            00004300
      RETURN                                                                  00004310
  910 WRITE(6,810)                                                            00004320
      RETURN                                                                  00004330
  915 WRITE(6,815)                                                            00004340
      RETURN                                                                  00004350
  920 WRITE(6,820)                                                            00004360
      RETURN                                                                  00004370
C                                                                             00004380
C                        FORMAT STATEMENTS                                    00004390
C                                                                             00004400
  600 FORMAT('1'/////44X,'* * * * * * * * * * * * * * * * * * * * * * *'/     00004410
     1       44X,'*',41X,'*'/44X,'*  PICARD INTEGRATION CONVERGENCE ',        00004420
     2       'REPORT  *'/44X,'*',41X,'*'/44X,'* * * * * * * * * * * *',      00004430
     3       ' * * * * * * * * * *'///)                                       00004440
C                                                                             00004450
  605 FORMAT(///29X,'THE ORDER OF THE CHEBYSHEV SERIES APPROXIMATING ',       00004460
     1           'THE INTEGRAND IS:  M = ',I2,'.')                            00004470
C                                                                             00004480
  610 FORMAT(///27X,'THE NUMBER OF CHEBYSHEV POINTS AT WHICH THE INTEG',      00004490
     1           'RAND IS EVALUATED IS: N+1 = ',I2,'.')                       00004500
C                                                                             00004510
  615 FORMAT(///1X,131('*')/1X,45('*'),38X,48('*')/1X,45('*'),'   END ',      00004520
     1       'OF PICARD COEFFICIENT REPORT   ',48('*')/1X,45('*'),38X,        00004530
     2       48('*')/1X,131('*')/'1')                                         00004540
C                                                                             00004550
  620 FORMAT('1'///38X,'THE ELEMENT RATE AND ELEMENT SERIES COEFFICIE',       00004560
```

```
     1              'NTS FOR VOP ELEMENT ',I1,' ARE:'//28X,'ELEMENT ',  00004570
     2              'RATE SERIES COEFFICIENTS',29X,'ELEMENT SERIES ',   00004580
     3              'COEFFICIENTS')                                     00004590
C                                                                        00004600
  625 FORMAT('1'///10X,'THE COEFFICIENTS FOR COMPONENT ',I1,' OF THE ', 00004610
     1              'STATE VECTOR ARE:'/8X,'ACCELERATION SERIES ',      00004620
     2              'COEFFICIENTS',9X,'VELOCITY SERIES COEFFICIENTS',   00004630
     3              13X,'POSITION SERIES COEFFICIENTS'/)                00004640
C                                                                        00004650
  630 FORMAT(7X,'A(',I2,',',I1,') =',1PD25.15,5X,'B(',I2,',',I1,') =',  00004660
     1          1PD25.15,7X,'C(',I2,',',I1,') =',1PD25.15)              00004670
C                                                                        00004680
  635 FORMAT('1'////28X,'**** THE PICARD INTEGRATION METHOD FAILED TO',  00004690
     1              ' CONVERGE IN ',I2,' ITERATIONS. ****'//)           00004700
C                                                                        00004710
  640 FORMAT(///32X,'THE CONVERGENCE CRITERIA IN THIS CASE WAS THREE ',  00004720
     1              'SUCCESSIVE SOLUTIONS'/42X,'WHICH AGREE TO WITHIN ',00004730
     2              'THE FOLLOWING TOLERANCES:'/48X,'EQUATION 1:',      00004740
     3              1PD25.15/48X,'EQUATION 2:',1PD25.15/48X,'EQUATION ',00004750
     4              '3:',1PD25.15/48X,'EQUATION 4:',1PD25.15/48X,       00004760
     5              'EQUATION 5:',1PD25.15/48X,'EQUATION 6:',1PD25.15)  00004770
C                                                                        00004780
  645 FORMAT(///40X,'THE INTEGRATOR ACHIEVED CONVERGENCE IN ',I2,        00004790
     1              ' ITERATIONS.'/42X,'THE MAXIMUM NUMBER OF ',        00004800
     2              'ITERATIONS ALLOWED WAS ',I3,'.'///39X,             00004810
     3              'THE COEFFICIENTS OF THE INTEGRAND AND INTE',       00004820
     4              'GRATED SERIES'/50X,'ARE SHOWN ON THE FOLLOW',      00004830
     5              'ING PAGES.')                                       00004840
C                                                                        00004850
  650 FORMAT(///33X,'THE EQUATIONS BEING INTEGRATED ARE THE COWELL ',    00004860
     1              'EQUATIONS OF MOTION.')                             00004870
C                                                                        00004880
  655 FORMAT(///18X,'THE EQUATIONS BEING INTEGRATED ARE THE KEPLERIAN', 00004890
     1              ' VARIATION OF PARAMETER EQUATIONS (MEAN ANOMALY).')00004900
C                                                                        00004910
  660 FORMAT(///16X,'THE EQUATIONS BEING INTEGRATED ARE THE EQUINO',     00004920
     1              'CTIAL VARIATION OF PARAMETER EQUATIONS (MEAN ',    00004930
     2              'LONGITUDE).')                                      00004940
C                                                                        00004950
  665 FORMAT('1',131('*')/1X,40('*'),51X,40('*')/1X,40('*'),             00004960
     1       '   BEGIN PICARD INTEGRATION METHOD FOR THIS STEP    ',    00004970
     2       40('*')/1X,40('*'),51X,40('*')/1X,131('*')/////)           00004980
C                                                                        00004990
  670 FORMAT(///27X,'THE ARC LENGTH OF THIS INTEGRATION STEP WAS',       00005000
     1       1PD23.15,' SECONDS.')                                       00005010
C                                                                        00005020
  690 FORMAT(27X,'A(',I2,',',I1,') =',1PD25.15,25X,'B(',I2,',',I1,') =',00005030
     1              1PD25.15)                                           00005040
C                                                                        00005050
  800 FORMAT('1'////10X,'THE IEQSET INPUT FLAG MUST BE SET TO 1, 2 OR',  00005060
     1              ' 3.'/10X,'IEQSET = 1 IMPLIES INTEGRATION OF THE ', 00005070
     2              'COWELL EQUATIONS OF MOTION.'/10X,'IEQSET = 2 ',    00005080
     3              'IMPLIES INTEGRATION OF THE KEPLERIAN VOP ',        00005090
     4              'EQUATIONS.'/10X,'IEQSET = 3 IMPLIES INTEGRATION ', 00005100
     5              'OF THE EQUINOCTIAL VOP EQUATIONS.'/10X,'MODIFY ',  00005110
     6              'IEQSET INPUT PARAMETER AND RETRY.')                00005120
C                                                                        00005130
```

```
VTEST                JSS1493.PRLLEL.ORBDET.FORT

  810 FORMAT('1'////10X,'THE MAXIMUM ORDER CHEBYSHEV POLYNOMIAL IN TH', 00005140
     1                  'IS PROGRAM IS RESTRICTED TO BE 68 OR LESS.'/  00005150
     2                  10X,'MODIFY M INPUT PARAMETER AND RETRY.')     00005160
C                                                                      00005170
  805 FORMAT('1'////10X,'THE VALUE OF THE TOTAL NUMBER OF ITERATIONS ', 00005180
     1                  'ALLOWED MUST BE GREATER THAN TWO BECAUSE ',   00005190
     2                  'THE PROGRAM'/10X,'THE PROGRAM NEEDS THREE ',  00005200
     3                  'SUCCESSIVE SOLUTIONS THAT AGREE TO WITHIN ',  00005210
     4                  'THE CONVERGENCE CRITERIA.'/10X,'MODIFY THE ',  00005220
     5                  'INPUT PARAMETER ITERM AND RETRY.')            00005230
C                                                                      00005240
  815 FORMAT('1'////10X,'THE LARGEST SUBSCRIPT N ON THE CHEBYSHEV POI', 00005250
     1                  'NTS OF ORTHOGONALITY IS RESTRICTED TO BE 69 ', 00005260
     2                  'OR LESS.'/10X,'MODIFY N INPUT PARAMETER AND ', 00005270
     3                  'RETRY.')                                      00005280
C                                                                      00005290
  820 FORMAT('1'////20X,'THE PROBLEM IS IMPROPERLY POSED.'//20X,'THE ', 00005300
     1                  'LOWER LIMIT OF INTEGRATION IS GREATER THAN ', 00005310
     2                  'THE UPPER LIMIT.'/20X,'MODIFY INPUT ',        00005320
     3                  'PARAMETERS AND RETRY.')                       00005330
C                                                                      00005340
      END                                                              00005350

******** END OF MEMBER VTEST      535 RECORDS ********
```

XEPOCH JSS1493.PRLLEL.ORBDET.FORT

```
      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK XEPOCH  00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/XEPOCH/   EPST(6)      ,   EPAK(26)     ,   EPAE(26)     , 00000120
     *                 PEODKO(6,6)  ,   PEODXO(6,6)                   00000130
C                                                                       00000140
C                                                                       00000150
C                      #LOCA-                                           00000160
C            DIMEN-     TION                                           00000170
C  VARIABLE   SION      (REAL)                 DESCRIPTION             00000180
C  ________  _______   ________  ____________________________________  00000190
C                                                                       00000200
C      EPST        6          1  ARRAY (6X1) CONTAINING THE EPOCH STATE 00000210
C                                COMPONENTS                             00000220
C      EPAK       26          7  ARRAY (26X1) CONTAINING THE EPOCH      00000230
C                                KEPLERIAN ELEMENTS AND 20 AUXLIARY     00000240
C                                QUANTITIES PRODUCED BY SUBROUTINE      00000250
C                                'ELMRPT'                               00000260
C      EPAE       26         33  ARRAY (26X1) CONTAINING THE EPOCH      00000270
C                                EQUINOCTIAL ELEMENTS AND 20 AUXLIARY   00000280
C                                QUANTITIES PRODUCED BY SUBROUTINE      00000290
C                                'ELMRPT'                               00000300
C    PEODKO      6,6         59  ARRAY (6X6) CONTAINING THE PARTIAL     00000310
C                                DERIVATIVES OF THE EPOCH EQUINOCTIAL   00000320
C                                ELEMENTS WITH RESPECT TO THE EPOCH     00000330
C                                KEPLERIAN ELEMENTS                     00000340
C    PEODXO      6,6         95  ARRAY (6X6) CONTAINING THE PARTIAL     00000350
C                                DERIVATIVES OF THE EPOCH EQUINOCTIAL   00000360
C                                ELEMENTS WITH RESPECT TO THE EPOCH     00000370
C                                POSITION AND VELOCITY VECTORS          00000380
C                                                                       00000390
      DATA EPST   / 6*0.D0      /                                       00000400
C                                                                       00000410
      DATA EPAK   / 26*0.D0     /                                       00000420
C                                                                       00000430
      DATA EPAE   / 26*0.D0     /                                       00000440
C                                                                       00000450
      DATA PEODKO / 36*0.D0     /                                       00000460
C                                                                       00000470
      DATA PEODXO / 36*0.D0     /                                       00000480
C                                                                       00000490
      END
```

******** END OF MEMBER XEPOCH 49 RECORDS ********

```
XESTMT              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK XESTMT  00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/XESTMT/   XAP(6)       ,   SCLFAC(6)    ,   EPSILN     ,  00000120
     *                 ALFA1        ,   STPSZ        ,   C1DIM      ,  00000130
     *                 CMTXAP(6,6)  ,   XMIN(6)                        00000140
C                                                                       00000150
C                      #LOCA-                                           00000160
C             DIMEN-    TION                                            00000170
C  VARIABLE    SION     (REAL)              DESCRIPTION                 00000180
C  ________   ______   _______   ____________________________________   00000190
C                                                                       00000200
C       XAP        6         1   ARRAY (6X1) CONTAINING THE A PRIORI     00000210
C                                SOLVE-FOR VECTOR COMPONENTS             00000220
C    SCLFAC        6         7   ARRAY (6X1) CONTAINING THE SET OF       00000230
C                                SCALE FACTORS TO BE APPLIED TO THE      00000240
C                                BASIS VECTORS SPANNING THE PARAMETER    00000250
C                                SPACE TO PROPERLY ADJUST THEIR MAG-     00000260
C                                NITUDES                                 00000270
C    EPSILN        1        13   CONVERGENCE CRITERION FOR THE PVM       00000280
C                                METHOD                                  00000290
C     ALFA1        1        14   STARTING VALUE OF THE INDEPENDENT       00000300
C                                VARIABLE ALPHA IN THE UNIVARIATE        00000310
C                                SEARCH ALGORITHM                        00000320
C     STPSZ        1        15   INITIAL ESTIMATE OF THE SEARCH STEP-    00000330
C                                SIZE IN THE UNIVARIATE SEARCH ALGO-     00000340
C                                RITHM                                   00000350
C     C1DIM        1        16   CONVERGENCE CRITERION FOR THE UNI-      00000360
C                                VARIATE SEARCH ALGORITHM                00000370
C    CMTXAP      6,6        17   ARRAY (6X6) CONTAINING THE A PRIORI     00000380
C                                ESTIMATION ERROR COVARIANCE MATRIX      00000390
C      XMIN        6        53   ARRAY (6X1) CONTAINING THE MINIMIZING   00000400
C                                VALUE OF THE SOLVE-FOR VECTOR X         00000410
C                                                                       00000420
      DATA XAP    / 6*0.D0      /                                       00000430
C                                                                       00000440
      DATA SCLFAC / 6*0.D0      /                                       00000450
C                                                                       00000460
      DATA EPSILN / 1.D-03      /                                       00000470
C                                                                       00000480
      DATA ALFA1  / 0.D0        /                                       00000490
C                                                                       00000500
      DATA STPSZ  / 5.D-01      /                                       00000510
C                                                                       00000520
      DATA C1DIM  / 1.D-03      /                                       00000530
C                                                                       00000540
      DATA CMTXAP / 36*0.D0     /                                       00000550
C                                                                       00000560
      DATA XMIN   / 6*0.D0      /                                       00000570
```

```
XESTMT              JSS1493.PRLLEL.ORBDET.FORT

C                                                                       00000580
      END                                                               00000590

******** END OF MEMBER XESTMT      59 RECORDS ********
```

```
XPTLEQ              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 01/01/80      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C        TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK XPTLEQ    00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C        CAPTAIN JEFFREY S. SHAVER, USAF                                00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/XPTLEQ/   PRDEQ(3,6)     ,   PVDEQ(3,6)     ,   PXDEQ(6,6) 00000120
C                                                                       00000130
C                      #LOCA-                                           00000140
C             DIMEN-     TION                                           00000150
C  VARIABLE    SION     (REAL)              DESCRIPTION                 00000160
C  ________   _______   _______   ____________________________________  00000170
C                                                                       00000180
C     PRDEQ       3,6         1   ARRAY (3X6) CONTAINING THE PARTIAL    00000190
C                                 DERIVATIVES OF THE CURRENT POSITION   00000200
C                                 VECTOR WITH RESPECT TO THE CURRENT    00000210
C                                 EQUINOCTIAL ELEMENTS                  00000220
C     PVDEQ       3,6        19   ARRAY (3X6) CONTAINING THE PARTIAL    00000230
C                                 DERIVATIVES OF THE CURRENT VELOCITY   00000240
C                                 VECTOR WITH RESPECT TO THE CURRENT    00000250
C                                 EQUINOCTIAL ELEMENTS                  00000260
C     PXDEQ       6,6        37   ARRAY (3X6) CONTAINING THE PARTIAL    00000270
C                                 DERIVATIVES OF THE CURRENT POSITION   00000280
C                                 AND VELOCITY VECTORS WITH RESPECT TO  00000290
C                                 THE CURRENT EQUINOCTIAL ELEMENTS      00000300
C                                                                       00000310
      DATA PRDEQ  / 18*0.D0       /                                     00000320
C                                                                       00000330
      DATA PVDEQ  / 18*0.D0       /                                     00000340
C                                                                       00000350
      DATA PXDEQ  / 36*0.D0       /                                     00000360
C                                                                       00000370
      END                                                               00000380

******** END OF MEMBER XPTLEQ      38 RECORDS ********
```

ZNLCOF JSS1493.PRLLEL.ORBDET.FORT

```
      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 08/07/79      **** CHECKED OUT ****                    00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VARIABLES IN THE NAMED COMMON BLOCK ZNLCOF  00000060
C                                                                       00000070
C     PROGRAMMING                                                       00000080
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000090
C                                                                       00000100
      IMPLICIT REAL*8(A-H,O-Z)                                          00000110
      COMMON/ZNLCOF/  SHJ(8)  ,  IZNLS                                  00000120
C                                                                       00000130
C                       #LOCA-                                          00000140
C              DIMEN-     TION                                         00000150
C  VARIABLE     SION     (REAL)              DESCRIPTION                00000160
C  ________    _____    _______  ____________________________________  00000170
C                                                                       00000180
C  SHJ             8         1   ARRAY (8X1) CONTAINING THE ZONAL      00000190
C                                HARMONIC COEFFICIENTS J  THROUGH J     00000200
C                                                       1         8     00000210
C                                (NOTE:  IT IS ASSUMED THAT J = 0)      00000220
C                                                          1            00000230
C                                                                       00000240
C                       #LOCA-                                          00000250
C              DIMEN-     TION                                         00000260
C  VARIABLE     SION    INTEGER              DESCRIPTION                00000270
C  ________    _____    _______  ____________________________________  00000280
C                                                                       00000290
C  IZNLS           1         1   FLAG TO INDICATE THAT ZONAL HARMONICS  00000300
C                                ARE TO BE INCLUDED IN THE FORCE MODEL: 00000310
C                                IZNLS<2 => NO ZONALS IN FORCE MODEL    00000320
C                                IZNLS>1 => ZONALS THROUGH ORDER IZNLS  00000330
C                                           ARE INCLUDED IN THE FORCE   00000340
C                                           MODEL                       00000350
C                                                                       00000360
      DATA SHJ      / 0.000000D 00, 0.108265D-02,-0.254503D-05,         00000370
     *                -0.167150D-05,-0.206721D-06, 0.640062D-06,         00000380
     *                -0.407213D-06,-0.329265D-06/                       00000390
C                                                                       00000400
      DATA IZNLS    /0            /                                     00000410
C                                                                       00000420
      END                                                               00000430

******** END OF MEMBER ZNLCOF      43 RECORDS ********
```

APPENDIX E

SIMULATION OF OBSERVATIONAL DATA

The observational data for the estimation test cases of Chapter 3 were assumed to be scalar components of the inertial position and/or velocity vectors. The true values for these observations were generated by a high precision integration of a truth ephemeris for the particular test case under consideration. It then remained to corrupt these true observations with observation errors. It was assumed that these errors were normally distributed with zero means and standard deviations selectable by the user.

In order to provide the necessary simulated observational data a small amount of software had to be written. The primary simulation driver is routine 'MAINSM'. This routine reads in the true observational data in the format 4G20.13, as shown below:

Column	Format	Description
1-20	G20.13	Observation time (minutes from epoch)
21-40	G20.13	Observation type: 1.0 = "x" position component 2.0 = "y" position component 3.0 = "z" position component 4.0 = "x" velocity component 5.0 = "y" velocity component 6.0 = "z" velocity component
41-60	G20.13	Actual truth ephemeris observation quantity (km or km/sec)
61-80	G20.13	Standard deviation associated with the simulated observational error for the particular observation type (km or km/sec)

Unfortunately, no automatic interface exists to prepare the input data in the above format, so it must be prepared manually from the truth ephemeris integration printout.

After 'MAINSM' reads the input deck, which may contain up to 100 scalar observations, it calls subroutine 'ERRORS' to generate the normally distributed random errors with zero mean and the standard deviations indicated in columns 61-80 of each input card. Subroutine 'ERRORS', in turn, calls subroutine 'RNDNUM', which is a random number generator. These two subroutines are taken directly from the Fortran Scientific Subroutine Package, where they care called 'GAUSS' and 'RANDU' respectively.

The routines 'MAINSM', 'ERRORS' and 'RNDNUM' are concatenated as shown in Figure E.1 so they can be compiled together. This allows the package to be executed online at a TSO terminal. It should be noted that the number of observations (NOBS) is currently hardcoded in 'MAINSM' to be 60. If a different number of observations is required, of if it is desired to change the initializing integer for the random number generator (IX), then 'MAINSM' must be recompiled.

When 'MAINSM' has computed all the required simulated observations, the output of the data is done in such a format that the resulting output file (if running online) can be easily converted to a named common block. This is done using the editing capabilities of the TSO terminal. The resulting named common block is called OBSDTx.FORT, where the x identifier is user specified. An example of this common block is given in Figure E.2. Once this data set is prepared, it must be compiled and the resulting object module included in the Job Control Language linkage for the particular estimation run. The estimation algorithms then use the simulated observational data in this common block.

```
MAINSM              JSS1493.PRLLEL.UTLITY.FORT

C     MAINSM.FORT                                                        00000010
C                                                                        00000020
C     VERSION OF 09/15/79      **** CHECKED OUT ****                     00000030
C                                                                        00000040
C     PURPOSE                                                            00000050
C         TO CONTROL THE CREATION OF SIMULATED OBSERVATIONAL DATA         00000060
C         FOR USE IN THE ESTIMATION PROBLEM.                             00000070
C                                                                        00000080
C     METHOD                                                             00000090
C         MAINSM READS IN A SET OF DATA CARD IMAGES WHICH CONTAIN THE    00000100
C         FOLLOWING DATA:                                                00000110
C         COLUMNS  1-20:  OBSERVATION TIME (MIN)                         00000120
C         COLUMNS 21-40:  OBSERVATION TYPE                               00000130
C                         1 = X-POSITION COMPONENT                       00000140
C                         2 = Y-POSITION COMPONENT                       00000150
C                         3 = Z-POSITION COMPONENT                       00000160
C                         4 = X-VELOCITY COMPONENT                       00000170
C                         5 = Y-VELOCITY COMPONENT                       00000180
C                         6 = Z-VELOCITY COMPONENT                       00000190
C         COLUMNS 41-60:  ACTUAL OBSERVATION (KM OR KM/SEC)              00000200
C         COLUMNS 61-80:  STANDARD DEVIATION ASSOCIATED WITH THE         00000210
C                         SIMULATED MEASUREMENT ERROR FOR THE PAR-       00000220
C                         TICULAR OBSERVATION TYPE(KM OR KM/SEC)         00000230
C                                                                        00000240
C         THE SUBROUTINE ERRORS IS THEN CALLED WHICH GENERATES A         00000250
C         NORMALLY DISTRIBUTED RANDOM ERROR WITH THE GIVEN STANDARD      00000260
C         DEVIATION, WHICH IS ADDED TO THE ACTUAL OBSERVATION TO         00000270
C         GENERATE THE SIMULATED OBSERVATION.                            00000280
C                                                                        00000290
C         THIS PROGRAM IS DESIGNED TO BE RUN ON-LINE, SUCH THAT THE      00000300
C         OUTPUT IS WRITTEN TO A TEMPORARY DATA FILE.  THIS FILE IS      00000310
C         THEN APPENDED TO THE TSO FILE 'OBSTUF.FORT' BY USE OF THE      00000320
C         "INCLUDE" COMMAND.  THE RESULTING DATA FILE IS THEN COPIED     00000330
C         INTO THE DESIRED VERSION NAME OF THE COMMON BLOCK /OBSDAT/,    00000340
C         EDITED AS APPROPRIATE AND THEN COMPILED.  THIS NEW VERSION     00000350
C         OF /OBSDAT/ IS THEN ADDED TO THE JCL FOR THAT ESTIMATION RUN.  00000360
C                                                                        00000370
C                                                                        00000380
C * * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00000390
C                                                                        00000400
      IMPLICIT REAL*8(A-H,O-Z)                                           00000410
C                                                                        00000420
C * * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000430
C                                                                        00000440
      DIMENSION       C1(100)   ,   C2(100)   ,   C3(100)   ,           00000450
     *                C4(100)   ,   C5(100)   ,   C6(100)   ,           00000460
     *                CMTX(100)                                         00000470
C                                                                        00000480
C     INITIALIZE AUXILIARY PARAMETERS:  NOBS IS SELECTABLE FOR EACH RUN  00000490
C                                                                        00000500
      NOBS  = 60                                                         00000510
      AMEAN = 0.D0                                                       00000520
      IX    = 9935917                                                    00000530
C                                                                        00000540
C     READ DATA CARDS; COMPUTE THE RANDOM NORMALLY DISTRIBUTED ERROR;    00000550
C     COMPUTE THE SIMULATED OBSERVATION; COMPUTE THE DIAGONAL ELEMENTS   00000560
C     OF THE INVERSE OF THE COVARIANCE MATRIX OF THE MEASUREMENT ERRORS  00000570
```

Figure E.1. Data Simulation Software Listing (Page 1 of 4)

```
MAINSM              JSS1493.PRLLEL.UTLITY.FORT

C                                                                       00000580
      DO  10   I = 1,NOBS                                               00000590
      READ(5,100)  C1(I),C2(I),C3(I),C5(I)                              00000600
      CALL ERRORS(IX,C5(I),AMEAN,C4(I))                                 00000610
      C6(I) = C3(I) + C4(I)                                             00000620
      CMTX(I) = (1.D0/C5(I))**2                                         00000630
   10 CONTINUE                                                          00000640
C                                                                       00000650
C     WRITE OUT THE FIRST COLUMN IN 'OBSFIL':  OBSERVATION TIME         00000660
C                                                                       00000670
      WRITE(6,120)                                                      00000680
      DO  20   I = 1,100,3                                              00000690
      WRITE(6,110)  C1(I),C1(I+1),C1(I+2)                               00000700
   20 CONTINUE                                                          00000710
C                                                                       00000720
C     WRITE THE SECOND COLUMN IN 'OBSFIL':  OBSERVATION TYPE            00000730
C                                                                       00000740
      WRITE(6,120)                                                      00000750
      DO  30   I = 1,100,3                                              00000760
      WRITE(6,110)  C2(I),C2(I+1),C2(I+2)                               00000770
   30 CONTINUE                                                          00000780
C                                                                       00000790
C     WRITE OUT THE THIRD COLUMN IN 'OBSFIL':  ACTUAL OBSERVATION       00000800
C                                                                       00000810
      WRITE(6,120)                                                      00000820
      DO  40   I = 1,100,3                                              00000830
      WRITE(6,110)  C3(I),C3(I+1),C3(I+2)                               00000840
   40 CONTINUE                                                          00000850
C                                                                       00000860
C     WRITE OUT THE FOURTH COLUMN IN 'OBSFIL':  OBSERVATION ERROR       00000870
C                                                                       00000880
      WRITE(6,120)                                                      00000890
      DO  50   I = 1,100,3                                              00000900
      WRITE(6,110)  C4(I),C4(I+1),C4(I+2)                               00000910
   50 CONTINUE                                                          00000920
C                                                                       00000930
C     WRITE OUT THE FIFTH COLUMN IN 'OBSFIL':  STANDARD DEVIATION       00000940
C                                                                       00000950
      WRITE(6,120)                                                      00000960
      DO  60   I = 1,100,3                                              00000970
      WRITE(6,110)  C5(I),C5(I+1),C5(I+2)                               00000980
   60 CONTINUE                                                          00000990
C                                                                       00001000
C     WRITE OUT THE SIXTH COLUMN IN 'OBSFIL':  SIMULATED OBSERVATION    00001010
C                                                                       00001020
      WRITE(6,120)                                                      00001030
      DO  70   I = 1,100,3                                              00001040
      WRITE(6,110)  C6(I),C6(I+1),C6(I+2)                               00001050
   70 CONTINUE                                                          00001060
C                                                                       00001070
C     WRITE OUT THE DIAGONAL ELEMENTS OF THE INVERSE OF THE COVARIANCE   00001080
C     MATRIX OF THE MEASUREMENT ERRORS                                  00001090
C                                                                       00001100
      WRITE(6,120)                                                      00001110
      WRITE(6,120)                                                      00001120
      DO  80   I = 1,100,3                                              00001130
      WRITE(6,110)  CMTX(I),CMTX(I+1),CMTX(I+2)                         00001140
```

Figure E.1. Data Simulation Software Listing (Page 2 of 4)

```
MAINSM              JSS1493.PRLLEL.UTLITY.FORT

   80 CONTINUE                                                          00001150
      STOP                                                              00001160
C                                                                       00001170
C * * * * * * * * *        FORMAT STATEMENTS        * * * * * * * * * * 00001180
C                                                                       00001190
  100 FORMAT(4G20.13)                                                   00001200
C                                                                       00001210
  110 FORMAT(5X,'* ',3(1PD19.9,' ,'))                                   00001220
C                                                                       00001230
  120 FORMAT(/)                                                         00001240
C                                                                       00001250
      END                                                               00001260
      SUBROUTINE ERRORS(IX,S,AM,V)                                      00001270
C                                                                       00001280
C     VERSION OF 09/15/79      **** CHECKED OUT ****                    00001290
C                                                                       00001300
C     PURPOSE                                                           00001310
C         COMPUTES A NORMALLY DISTRIBUTED RANDOM NUMBER WITH A GIVEN    00001320
C         MEAN AND STANDARD DEVIATION.                                  00001330
C                                                                       00001340
C     CALLING SEQUENCE                                                  00001350
C         CALL ERRORS(IX,S,AM,V)                                        00001360
C                                                                       00001370
C     CALLING SEQUENCE PARAMETERS                                       00001380
C         IX        I   IX MUST CONTAIN AN ODD INTEGER NUMBER WITH NINE 00001390
C                       OR LESS DIGITS ON THE FIRST ENTRY TO SUBROUTINE.00001400
C                       THEREAFTER, IT WILL CONTAIN A UNIFORMLY DISTRI- 00001410
C                       BUTED INTEGER RANDOM NUMBER GENERATED BY THE    00001420
C                       SUBROUTINE FOR THE NEXT ENTRY TO THE SUBROUTINE.00001430
C         S         I   THE DESIRED STANDARD DEVIATION OF THE NORMAL    00001440
C                       DISTRIBUTION.                                   00001450
C         AM        I   THE DESIRED MEAN OF THE NORMAL DISTRIBUTION.    00001460
C         V         O   THE VALUE OF THE COMPUTED NORMAL RANDOM VARIABLE.00001470
C                                                                       00001480
C     EXTERNAL REFERENCES                                               00001490
C         SUBROUTINES                                                   00001500
C              RNDNUM                                                   00001510
C                                                                       00001520
C     ANALYSIS                                                          00001530
C         FORTRAN SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE GAUSS       00001540
C                                                                       00001550
C                                                                       00001560
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00001570
C                                                                       00001580
      IMPLICIT REAL*8(A-H,O-Z)                                          00001590
      A = 0.D0                                                          00001600
      DO  10   I = 1,12                                                 00001610
      CALL RNDNUM(IX,IY,YFL)                                            00001620
      IX = IY                                                           00001630
   10 A = A + YFL                                                       00001640
      V = (A-6.D0) * S + AM                                             00001650
      RETURN                                                            00001660
      END                                                               00001670
      SUBROUTINE RNDNUM(IX,IY,YFL)                                      00001680
C                                                                       00001690
C     VERSION OF 09/15/79      **** CHECKED OUT ****                    00001700
C                                                                       00001710
```

Figure E.1. Data Simulation Software Listing (Page 3 of 4)

```
MAINSM              JSS1493.PRLLEL.UTLITY.FORT

C     PURPOSE                                                          00001720
C         COMPUTES UNIFORMLY DISTRIBUTED RANDOM REAL NUMBERS BETWEEN   00001730
C         0 AND 1.0 AND RANDOM INTEGERS BETWEEN ZERO AND 2**31.        00001740
C         EACH ENTRY USES AS INPUT AN INTEGER RANDOM NUMBER AND PRO-   00001750
C         DUCES A NEW INTEGER AND REAL RANDOM NUMBER.                  00001760
C                                                                      00001770
C     CALLING SEQUENCE                                                 00001780
C         CALL RNDNUM(IX,IY,YFL)                                       00001790
C                                                                      00001800
C     CALLING SEQUENCE PARAMETERS                                      00001810
C         IX        I   FOR THE FIRST ENTRY, THIS MUST CONTAIN ANY ODD 00001820
C                       INTEGER NUMBER WITH NINE OR LESS DIGITS.  AFTER 00001830
C                       THE FIRST ENTRY, IX SHOULD BE THE PREVIOUS VALUE 00001840
C                       OF IY COMPUTED BY THIS SUBROUTINE.             00001850
C         IY        O   A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR 00001860
C                       THE NEXT ENTRY TO THIS SUBROUTINE.  THE RANGE  00001870
C                       OF THIS NUMBER IS BETWEEN ZERO AND 2**31.      00001880
C         YFL       O   THE RESULTANT UNIFORMLY DISTRIBUTED FLOATING   00001890
C                       POINT RANDOM NUMBER IN THE RANGE 0 TO 1.0.     00001900
C                                                                      00001910
C     EXTERNAL REFERENCES                                              00001920
C         NONE                                                         00001930
C                                                                      00001940
C     ANALYSIS                                                         00001950
C         FORTRAN SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE RANDU      00001960
C                                                                      00001970
C                                                                      00001980
C * * * * * * * * *          START PROGRAM          * * * * * * * * * * 00001990
C                                                                      00002000
      IMPLICIT REAL*8(A-H,O-Z)                                         00002010
      IY = IX * 65539                                                  00002020
      IF (IY)   10,20,20                                               00002030
   10 IY = IY + 2147433647 + 1                                         00002040
   20 YFL = IY                                                         00002050
      YFL = YFL * .4656613D-09                                         00002060
      RETURN                                                           00002070
      END                                                              00002080

******** END OF MEMBER MAINSM     208 RECORDS ********
```

Figure E.1. Data Simulation Software Listing (Page 4 of 4)

```
OBSDT9              JSS1493.PRLLEL.ORBDET.FORT

      BLOCK DATA                                                        00000010
C                                                                       00000020
C     VERSION OF 09/15/79     **** CHECKED OUT ****                     00000030
C                                                                       00000040
C     PURPOSE                                                           00000050
C         TO INITIALIZE THE VALUES STORED IN THE LABELLED COMMON        00000060
C         BLOCK /OBSDAT/ FOR THE ESTIMATION PROBLEM                     00000070
C                                                                       00000080
C     PROGRAMMING                                                       00000090
C         CAPTAIN JEFFREY S. SHAVER, USAF                               00000100
C                                                                       00000110
      IMPLICIT REAL*8(A-H,O-Z)                                          00000120
C                                                                       00000130
C * * * * * * * * *          COMMON BLOCK           * * * * * * * * * * 00000140
C                                                                       00000150
      COMMON/OBSDAT/  OBSFIL(100,6)         ,       OBSPAR(100,6)     , 00000160
     *                OBSCMP(100,6)         ,       RES(100)          , 00000170
     *                CMTXMS(100)           ,       NOBS                00000180
C                                                                       00000190
C * * * * * * * * *      DIMENSION STATEMENTS       * * * * * * * * * * 00000200
C                                                                       00000210
      DIMENSION       C11(48),C12(52),C21(48),C22(52),C31(48),C32(52),  00000220
     *                C41(48),C42(52),C51(48),C52(52),C61(48),C62(52),  00000230
     *                C71(48),C72(52)                                   00000240
C                                                                       00000250
C * * * * * * * * *    EQUIVALENCE STATEMENTS       * * * * * * * * * * 00000260
C                                                                       00000270
      EQUIVALENCE     (OBSFIL(1,1) , C11(1)) , (OBSFIL(49,1) , C12(1)) ,00000280
     *                (OBSFIL(1,2) , C21(1)) , (OBSFIL(49,2) , C22(1)) ,00000290
     *                (OBSFIL(1,3) , C31(1)) , (OBSFIL(49,3) , C32(1)) ,00000300
     *                (OBSFIL(1,4) , C41(1)) , (OBSFIL(49,4) , C42(1)) ,00000310
     *                (OBSFIL(1,5) , C51(1)) , (OBSFIL(49,5) , C52(1)) ,00000320
     *                (OBSFIL(1,6) , C61(1)) , (OBSFIL(49,6) , C62(1))  00000330
C                                                                       00000340
      EQUIVALENCE     (CMTXMS(1)   , C71(1)) , (CMTXMS(49)   , C72(1))  00000350
C                                                                       00000360
C * * * * * * * * *          DATA STATEMENTS        * * * * * * * * * * 00000370
C                                                                       00000380
C     INITIALIZE ARRAY OBSPAR:  THE ARRAY OF PARTIAL DERIVATIVES OF THE 00000390
C     SCALAR OBSERVATION WITH RESPECT TO THE SOLVE-FOR PARAMETERS       00000400
C                                                                       00000410
      DATA OBSPAR  /  600*0.D0  /                                       00000420
C                                                                       00000430
C     INITIALIZE ARRAY OBSCMP:  THE ARRAY OF THE POSITION AND VELOCITY  00000440
C     COMPONENTS AT THE OBSERVATION TIMES                               00000450
C                                                                       00000460
      DATA OBSCMP  /  600*0.D0  /                                       00000470
C                                                                       00000480
C     INITIALIZE ARRAY RES:  THE ARRAY OF THE OBSERVATION RESIDUALS     00000490
C                                                                       00000500
      DATA RES     /  100*0.D0  /                                       00000510
C                                                                       00000520
C     INITIALIZE ARRAY OBSFIL:  THE ARRAY OF THE SIMULATED OBSERVATION  00000530
C     DATA IN THE FOLLOWING FORMAT:                                     00000540
C         COLUMN 1:  OBSERVATION TIMES (MIN)                          00000550
C         COLUMN 2:  OBSERVATION TYPES                                00000560
C         COLUMN 3:  ACTUAL OBSERVATIONS                              00000570
```

Figure E.2. Named Common Block OBSDAT for Storage of Simulated Observational Data (Page 1 of 6)

```
OBSDT9              JSS1493.PRLLEL.ORBDET.FORT

C         COLUMN 4:  OBSERVATION ERRORS                                  00000580
C         COLUMN 5:  STANDARD DEVIATIONS                                 00000590
C         COLUMN 6:  SIMULATED OBSERVATIONS                              00000600
C                                                                        00000610
      DATA C11  /                                                        00000620
     *     3.0000000000D+00 ,    6.0000000000D+00 ,    9.0000000000D+00 , 00000630
     *     1.2000000000D+01 ,    1.5000000000D+01 ,    1.8000000000D+01 , 00000640
     *     2.1000000000D+01 ,    2.4000000000D+01 ,    2.7000000000D+01 , 00000650
     *     3.0000000000D+01 ,    3.3000000000D+01 ,    3.6000000000D+01 , 00000660
     *     3.9000000000D+01 ,    4.2000000000D+01 ,    4.5000000000D+01 , 00000670
     *     4.8000000000D+01 ,    5.1000000000D+01 ,    5.4000000000D+01 , 00000680
     *     5.7000000000D+01 ,    6.0000000000D+01 ,    6.3000000000D+01 , 00000690
     *     6.6000000000D+01 ,    6.9000000000D+01 ,    7.2000000000D+01 , 00000700
     *     7.5000000000D+01 ,    7.8000000000D+01 ,    8.1000000000D+01 , 00000710
     *     8.4000000000D+01 ,    8.7000000000D+01 ,    9.0000000000D+01 , 00000720
     *     9.3000000000D+01 ,    9.6000000000D+01 ,    9.9000000000D+01 , 00000730
     *     1.0200000000D+02 ,    1.0500000000D+02 ,    1.0800000000D+02 , 00000740
     *     1.1100000000D+02 ,    1.1400000000D+02 ,    1.1700000000D+02 , 00000750
     *     1.2000000000D+02 ,    1.2300000000D+02 ,    1.2600000000D+02 , 00000760
     *     1.2900000000D+02 ,    1.3200000000D+02 ,    1.3500000000D+02 , 00000770
     *     1.3800000000D+02 ,    1.4100000000D+02 ,    1.4400000000D+02 / 00000780
C                                                                        00000790
      DATA C12  /                                                        00000800
     *     1.4700000000D+02 ,    1.5000000000D+02 ,    1.5300000000D+02 , 00000810
     *     1.5600000000D+02 ,    1.5900000000D+02 ,    1.6200000000D+02 , 00000820
     *     1.6500000000D+02 ,    1.6800000000D+02 ,    1.7100000000D+02 , 00000830
     *     1.7400000000D+02 ,    1.7700000000D+02 ,    1.8000000000D+02 , 00000840
     *     0.0              ,    0.0              ,    0.0              , 00000850
     *     0.0              ,    0.0              ,    0.0              , 00000860
     *     0.0              ,    0.0              ,    0.0              , 00000870
     *     0.0              ,    0.0              ,    0.0              , 00000880
     *     0.0              ,    0.0              ,    0.0              , 00000890
     *     0.0              ,    0.0              ,    0.0              , 00000900
     *     0.0              ,    0.0              ,    0.0              , 00000910
     *     0.0              ,    0.0              ,    0.0              , 00000920
     *     0.0              ,    0.0              ,    0.0              , 00000930
     *     0.0              ,    0.0              ,    0.0              , 00000940
     *     0.0              ,    0.0              ,    0.0              , 00000950
     *     0.0              ,    0.0              ,    0.0              , 00000960
     *     0.0              ,    0.0              ,    0.0              , 00000970
     *     0.0              /                                           00000980
C                                                                        00000990
      DATA C21  /                                                        00001000
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001010
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001020
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001030
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001040
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001050
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001060
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001070
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001080
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001090
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001100
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001110
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001120
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001130
     *     1.0000000000D+00 ,    2.0000000000D+00 ,    3.0000000000D+00 , 00001140
```

Figure E.2. Named Common Block OBSDAT for Storage of Simulated Observational Data (Page 2 of 6)

```
     *.    1.000000000D+00 ,    2.000000000D+00 ,    3.000000000D+00 ,  00001150
     *     1.000000000D+00 ,    2.000000000D+00 ,    3.000000000D+00 /  00001160
C                                                                       00001170
      DATA C22  /                                                       00001180
     *     1.000000000D+00 ,    2.000000000D+00 ,    3.000000000D+00 ,  00001190
     *     1.000000000D+00 ,    2.000000000D+00 ,    3.000000000D+00 ,  00001200
     *     1.000000000D+00 ,    2.000000000D+00 ,    3.000000000D+00 ,  00001210
     *     1.000000000D+00 ,    2.000000000D+00 ,    3.000000000D+00 ,  00001220
     *     0.0             ,    0.0             ,    0.0             ,  00001230
     *     0.0             ,    0.0             ,    0.0             ,  00001240
     *     0.0             ,    0.0             ,    0.0             ,  00001250
     *     0.0             ,    0.0             ,    0.0             ,  00001260
     *     0.0             ,    0.0             ,    0.0             ,  00001270
     *     0.0             ,    0.0             ,    0.0             ,  00001280
     *     0.0             ,    0.0             ,    0.0             ,  00001290
     *     0.0             ,    0.0             ,    0.0             ,  00001300
     *     0.0             ,    0.0             ,    0.0             ,  00001310
     *     0.0             ,    0.0             ,    0.0             ,  00001320
     *     0.0             ,    0.0             ,    0.0             ,  00001330
     *     0.0             ,    0.0             ,    0.0             ,  00001340
     *     0.0             ,    0.0             ,    0.0             ,  00001350
     *     0.0             /                                           00001360
C                                                                       00001370
      DATA C31  /                                                       00001380
     *    -1.052865426D+03 ,    6.249087435D+03 ,   -2.076334303D+03 ,  00001390
     *    -3.852438824D+03 ,    3.741407819D+03 ,   -3.983069282D+03 ,  00001400
     *    -5.263716575D+03 ,   -1.085710855D+02 ,   -4.507170142D+03 ,  00001410
     *    -4.890336326D+03 ,   -3.933909152D+03 ,   -3.566619644D+03 ,  00001420
     *    -2.962788131D+03 ,   -6.554964057D+03 ,   -1.529624056D+03 ,  00001430
     *    -1.375449132D+02 ,   -7.217313287D+03 ,    9.635928686D+02 ,  00001440
     *     2.728550904D+03 ,   -5.741586601D+03 ,    3.166155721D+03 ,  00001450
     *     4.769534966D+03 ,   -2.522102722D+03 ,    4.380689405D+03 ,  00001460
     *     5.300761797D+03 ,    1.496455359D+03 ,    4.144711033D+03 ,  00001470
     *     4.043945856D+03 ,    4.981532253D+03 ,    2.444062529D+03 ,  00001480
     *     1.336222274D+03 ,    6.630681144D+03 ,   -1.630980194D+02 ,  00001490
     *    -1.868929543D+03 ,    5.781849670D+03 ,   -2.710888286D+03 ,  00001500
     *    -4.380947306D+03 ,    2.791281867D+03 ,   -4.277098376D+03 ,  00001510
     *    -5.336568184D+03 ,   -1.189116505D+03 ,   -4.383674445D+03 ,  00001520
     *    -4.514267724D+03 ,   -4.797769487D+03 ,   -3.088682029D+03 ,  00001530
     *    -2.278359495D+03 ,   -6.942902703D+03 ,   -8.541686430D+02 /  00001540
C                                                                       00001550
      DATA C32  /                                                       00001560
     *     6.424412008D+02 ,   -7.028243158D+03 ,    1.634131161D+03 ,  00001570
     *     3.371783991D+03 ,   -5.023817681D+03 ,    3.627075762D+03 ,  00001580
     *     5.071868049D+03 ,   -1.477360591D+03 ,    4.472513638D+03 ,  00001590
     *     5.141286757D+03 ,    2.538318795D+03 ,    3.811222411D+03 ,  00001600
     *     0.0             ,    0.0             ,    0.0             ,  00001610
     *     0.0             ,    0.0             ,    0.0             ,  00001620
     *     0.0             ,    0.0             ,    0.0             ,  00001630
     *     0.0             ,    0.0             ,    0.0             ,  00001640
     *     0.0             ,    0.0             ,    0.0             ,  00001650
     *     0.0             ,    0.0             ,    0.0             ,  00001660
     *     0.0             ,    0.0             ,    0.0             ,  00001670
     *     0.0             ,    0.0             ,    0.0             ,  00001680
     *     0.0             ,    0.0             ,    0.0             ,  00001690
     *     0.0             ,    0.0             ,    0.0             ,  00001700
     *     0.0             ,    0.0             ,    0.0             ,  00001710
```

Figure E.2. Named Common Block OBSDAT for Storage of Simulated Observational Data (Page 3 of 6)

```
     *     0.0              ,    0.0              ,    0.0              ,  00001720
     *     0.0              ,    0.0              ,    0.0              ,  00001730
     *     0.0              /                                             00001740
C                                                                        00001750
      DATA C41  /                                                        00001760
     *     2.6815338520-02 ,   -3.7631204080-02 ,   -1.9022303870-02 ,  00001770
     *     1.5245568040-02 ,   -3.2659001540-02 ,    1.8759458540-03 ,  00001780
     *    -1.5705169840-02 ,   -5.4373028610-03 ,    1.4482898090-02 ,  00001790
     *    -3.5031384640-02 ,   -6.0611496020-03 ,   -3.5371868280-02 ,  00001800
     *    -1.3540022270-02 ,    1.0267491800-02 ,    8.3681699180-02 ,  00001810
     *     1.5712777460-02 ,   -3.1631140130-02 ,   -5.2029213610-04 ,  00001820
     *    -6.6039902770-03 ,   -2.4269989620-02 ,   -6.0545540500-03 ,  00001830
     *     3.4611639230-02 ,    2.4381860740-02 ,   -1.8898903560-02 ,  00001840
     *     1.2429382940-02 ,    3.8505203010-02 ,    4.0710301860-02 ,  00001850
     *    -6.3983629640-03 ,   -5.4618310360-02 ,    3.7015418700-02 ,  00001860
     *    -3.2982291560-02 ,    1.6353777150-02 ,   -2.9004478630-02 ,  00001870
     *     6.1262090200-03 ,   -4.5146783680-02 ,    2.9573102310-02 ,  00001880
     *    -1.6984658680-02 ,   -2.4476196740-02 ,   -2.1063428770-02 ,  00001890
     *    -7.2555035580-03 ,   -2.0999216240-02 ,    5.6872915410-02 ,  00001900
     *     3.0366203150-02 ,    3.3180052820-02 ,   -2.3661183590-02 ,  00001910
     *     5.6621712900-02 ,   -3.6451484250-03 ,    2.0620384140-03 /  00001920
C                                                                        00001930
      DATA C42  /                                                        00001940
     *     2.3252684030-03 ,    1.8488182810-02 ,   -3.7942611240-02 ,  00001950
     *     1.6613843250-02 ,   -5.0589684860-02 ,   -5.3328056500-02 ,  00001960
     *    -4.5409423800-02 ,   -8.0814100400-04 ,    3.3468989770-02 ,  00001970
     *     6.1954844460-03 ,   -4.4053844270-02 ,   -1.4080648330-02 ,  00001980
     *     0.0              ,    0.0              ,    0.0              ,  00001990
     *     0.0              ,    0.0              ,    0.0              ,  00002000
     *     0.0              ,    0.0              ,    0.0              ,  00002010
     *     0.0              ,    0.0              ,    0.0              ,  00002020
     *     0.0              ,    0.0              ,    0.0              ,  00002030
     *     0.0              ,    0.0              ,    0.0              ,  00002040
     *     0.0              ,    0.0              ,    0.0              ,  00002050
     *     0.0              ,    0.0              ,    0.0              ,  00002060
     *     0.0              ,    0.0              ,    0.0              ,  00002070
     *     0.0              ,    0.0              ,    0.0              ,  00002080
     *     0.0              ,    0.0              ,    0.0              ,  00002090
     *     0.0              ,    0.0              ,    0.0              ,  00002100
     *     0.0              ,    0.0              ,    0.0              ,  00002110
     *     0.0              /                                             00002120
C                                                                        00002130
      DATA C51  /                                                        00002140
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002150
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002160
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002170
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002180
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002190
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002200
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002210
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002220
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002230
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002240
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002250
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002260
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002270
     *     3.0000000000-02 ,    3.0000000000-02 ,    3.0000000000-02 ,  00002280
```

Figure E.2. Named Common Block OBSDAT for Storage of Simulated Observational Data (Page 4 of 6)

```
     *     3.000000000D-02 ,    3.000000000D-02 ,    3.000000000D-02 ,  00002290
     *     3.000000000D-02 ,    3.000000000D-02 ,    3.000000000D-02 /  00002300
C                                                                       00002310
      DATA C52  /                                                       00002320
     *     3.000000000D-02 ,    3.000000000D-02 ,    3.000000000D-02 ,  00002330
     *     3.000000000D-02 ,    3.000000000D-02 ,    3.000000000D-02 ,  00002340
     *     3.000000000D-02 ,    3.000000000D-02 ,    3.000000000D-02 ,  00002350
     *     3.000000000D-02 ,    3.000000000D-02 ,    3.000000000D-02 ,  00002360
     *     0.0             ,    0.0             ,    0.0             ,  00002370
     *     0.0             ,    0.0             ,    0.0             ,  00002380
     *     0.0             ,    0.0             ,    0.0             ,  00002390
     *     0.0             ,    0.0             ,    0.0             ,  00002400
     *     0.0             ,    0.0             ,    0.0             ,  00002410
     *     0.0             ,    0.0             ,    0.0             ,  00002420
     *     0.0             ,    0.0             ,    0.0             ,  00002430
     *     0.0             ,    0.0             ,    0.0             ,  00002440
     *     0.0             ,    0.0             ,    0.0             ,  00002450
     *     0.0             ,    0.0             ,    0.0             ,  00002460
     *     0.0             ,    0.0             ,    0.0             ,  00002470
     *     0.0             ,    0.0             ,    0.0             ,  00002480
     *     0.0             ,    0.0             ,    0.0             ,  00002490
     *     0.0             /                                           00002500
C                                                                       00002510
      DATA C61  /                                                       00002520
     *    -1.052838611D+03 ,    6.249049804D+03 ,   -2.076353325D+03 ,  00002530
     *    -3.852423578D+03 ,    3.741375160D+03 ,   -3.983067406D+03 ,  00002540
     *    -5.263732280D+03 ,   -1.085765228D+02 ,   -4.507155659D+03 ,  00002550
     *    -4.890371357D+03 ,   -3.938915213D+03 ,   -3.566655016D+03 ,  00002560
     *    -2.962801671D+03 ,   -6.554953790D+03 ,   -1.529540374D+03 ,  00002570
     *    -1.375292004D+02 ,   -7.217344918D+03 ,    9.635923483D+02 ,  00002580
     *     2.728544300D+03 ,   -5.741610871D+03 ,    3.166149666D+03 ,  00002590
     *     4.769569578D+03 ,   -2.522078340D+03 ,    4.380670506D+03 ,  00002600
     *     5.300774226D+03 ,    1.496493864D+03 ,    4.144751743D+03 ,  00002610
     *     4.043939458D+03 ,    4.981477635D+03 ,    2.444099544D+03 ,  00002620
     *     1.335189292D+03 ,    6.630697498D+03 ,   -1.631278239D+02 ,  00002630
     *    -1.868923417D+03 ,    5.781804523D+03 ,   -2.710858713D+03 ,  00002640
     *    -4.380964291D+03 ,    2.791257391D+03 ,   -4.277119439D+03 ,  00002650
     *    -5.336575440D+03 ,   -1.189137504D+03 ,   -4.383617572D+03 ,  00002660
     *    -4.514237358D+03 ,   -4.797736307D+03 ,   -3.088705690D+03 ,  00002670
     *    -2.278302873D+03 ,   -6.942906348D+03 ,   -8.541665810D+02 /  00002680
C                                                                       00002690
      DATA C62  /                                                       00002700
     *     6.424435261D+02 ,   -7.028224670D+03 ,    1.634093218D+03 ,  00002710
     *     3.371800605D+03 ,   -5.023867671D+03 ,    3.627022434D+03 ,  00002720
     *     5.071822640D+03 ,   -1.477361399D+03 ,    4.472547107D+03 ,  00002730
     *     5.141292952D+03 ,    2.538274741D+03 ,    3.811208330D+03 ,  00002740
     *     0.0             ,    0.0             ,    0.0             ,  00002750
     *     0.0             ,    0.0             ,    0.0             ,  00002760
     *     0.0             ,    0.0             ,    0.0             ,  00002770
     *     0.0             ,    0.0             ,    0.0             ,  00002780
     *     0.0             ,    0.0             ,    0.0             ,  00002790
     *     0.0             ,    0.0             ,    0.0             ,  00002800
     *     0.0             ,    0.0             ,    0.0             ,  00002810
     *     0.0             ,    0.0             ,    0.0             ,  00002820
     *     0.0             ,    0.0             ,    0.0             ,  00002830
     *     0.0             ,    0.0             ,    0.0             ,  00002840
     *     0.0             ,    0.0             ,    0.0             ,  00002850
```

Figure E.2. Named Common Block OBSDAT for Storage of Simulated Observational Data (Page 5 of 6)

```
OBSDT9              JSS1493.PRLLEL.ORBDET.FORT

     *     0.0             ,    0.0             ,    0.0             ,  00002860
     *     0.0             ,    0.0             ,    0.0             ,  00002870
     *     0.0            /                                             00002880
C                                                                       00002890
C     INITIALIZE ARRAY CMTXMS:  THE ARRAY OF THE DIAGONAL ELEMENTS OF   00002900
C     THE INVERSE OF THE MEASUREMENT COVARIANCE MATRIX                  00002910
C                                                                       00002920
      DATA C71  /                                                       00002930
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00002940
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00002950
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00002960
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00002970
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00002980
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00002990
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003000
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003010
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003020
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003030
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003040
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003050
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003060
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003070
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003080
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 /  00003090
C                                                                       00003100
      DATA C72  /                                                       00003110
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003120
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003130
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003140
     *     1.111111111D+03 ,    1.111111111D+03 ,    1.111111111D+03 ,  00003150
     *     0.0             ,    0.0             ,    0.0             ,  00003160
     *     0.0             ,    0.0             ,    0.0             ,  00003170
     *     0.0             ,    0.0             ,    0.0             ,  00003180
     *     0.0             ,    0.0             ,    0.0             ,  00003190
     *     0.0             ,    0.0             ,    0.0             ,  00003200
     *     0.0             ,    0.0             ,    0.0             ,  00003210
     *     0.0             ,    0.0             ,    0.0             ,  00003220
     *     0.0             ,    0.0             ,    0.0             ,  00003230
     *     0.0             ,    0.0             ,    0.0             ,  00003240
     *     0.0             ,    0.0             ,    0.0             ,  00003250
     *     0.0             ,    0.0             ,    0.0             ,  00003260
     *     0.0             ,    0.0             ,    0.0             ,  00003270
     *     0.0             ,    0.0             ,    0.0             ,  00003280
     *     0.0            /                                             00003290
C                                                                       00003300
C     INITIALIZE THE NUMBER OF OBSERVATIONS, NOBS                       00003310
C                                                                       00003320
      DATA NOBS    /  60       /                                       00003330
C                                                                       00003340
      END                                                               00003350

******** END OF MEMBER OBSDT9      335 RECORDS ********
```

Figure E.2. Named Common Block OBSDAT for Storage of Simulated Observational Data (Page 6 of 6)

Although this method is somewhat cumbersome, it is simplified by the availability of TSO terminals and their editing capabilities. The method proved quite adequate for this study.

REFERENCES

1. ADCOM DCD 1, Space Computational Center Program Documentation: Overview, USAF Aerospace Defense Command, Peterson AFB, Colorado, 1 May 1979.

2. ADCOM DCD 3-1.1, Space Computational Center Program Documentation (Astrodynamic Support): SPEPHX, USAF Aerospace Defense Command, Peterson AFB, Colorado, 25 March 1977.

3. ADCOM DCD 4-2.12, Space Computational Center Program Documentation: Special Support Routines, USAF Aerospace Defense Command, Peterson AFB, Colorado, 20 August 1976.

4. Adimurthy, V., and K. V. Joy, Estimation of Orbital Elements Using Methods of Nonlinear Optimization, Indian Space Research Organization, Bangalore, India, Scientific Note ISRO-VSSC-SN-02-77, March 1977.

5. Adimurthy, V., and K. V. Joy, "Extension of Region of Convergence in Orbit Determination Using Function-Minimization Techniques," Journal of Guidance and Control, Vol. 1, No. 6, November-December 1978.

6. Arsenault, J. L., J. D. Enright and C. Purcell, General Perturbation Techniques for Satellite Orbit Prediction Study: Volume I, Study and Results, Technical Documentary Report No. AL-TDR-64-70, Volume I, Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio, April 1964.

7. Aviation Week and Space Technology, "Antisatellite Effort Decision Awaited," 24 January 1977, p. 19.

8. _____, "Dual Mission Complicates Data Relay Satellite Work," 17 October 1977, pp. 155-157.

9. _____, "Earth Resources Benefits to Expand," 17 October 1977, pp. 42-47.

10. Aviation Week and Space Technology, "Killers Acknowledged," 14 November 1977, p. 23.

11. _____, "Killer Technology," 20 March 1977, p. 54.

12. _____, "Landsat Imagery - A $3 Million Business," 17 October 1977, p. 48.

13. _____, "Multimillion-Dollar Crop Forecast Business Seen," 17 October 1977, p. 61.

14. _____, "New Techniques Spur Landsat Data Use," 17 October 1977, pp. 48-65.

15. _____, "NOAA, Space Agency Defining Weather Satellite System for Deployment by 1985," 17 October 1977, pp. 140-145.

16. _____, "Soviet Killer Satellite System Confirmed," 10 October 1977, p. 18.

17. _____, "Soviets Duplicate Killer Satellite Mission," 30 May 1977, p. 15.

18. _____, "TDRS to Perform Dual Space Missions," 17 October 1977, pp. 97-98.

19. _____, Washington Roundup, "Antisatellite Bid," 7 November 1977, p. 17.

20. _____, Washington Roundup, "Killer Talks," 27 March 1978, p. 13.

21. _____, Washington Roundup, "Killer Talks," 20 November 1977, p. 13.

22. _____, Washington Roundup, "Study Pushed," 6 March 1978, p. 11.

23. Baer, J. L., "A Survey of Some Theoretical Aspects of Multiprocessing," ACM Computing Surveys, Volume 5, Number 1, March 1973, pp. 31-80.

24. Baer, J. L., "Multiprocessing Systems," IEEE Transactions on Computers, Volume C-25, Number 12, December 1976, pp. 1271-1277.

25. Baer, J. L., and C. S. Ellis, "Model, Design, and Evaluation of a Compiler for a Parallel Processing Environment," IEEE Transactions on Software Engineering, Volume SE-3, Number 6, November 1977, pp. 394-405.

26. Bard, Y., "Comparison of Gradient Methods for the Solution of Nonlinear Parameter Estimation Problems," SIAM Journal of Numerical Analysis, Volume 7, Number 1, March 1970, pp. 157-186.

27. Baskett, F., and T. W. Keller, "An Evaluation of the CRAY-1 Computer," High Speed Computer and Algorithm Organization, D. J. Kuck, et al., Editors, Academic Press, Inc., New York, 1977.

28. Battin, R. H., Astronautical Guidance, McGraw-Hill Book Company, New York, 1964.

29. Battin, R. H., and D. C. Fraser, Space Guidance and Navigation, AIAA Professional Study Series.

30. Battin, R. H., S. R. Croopnick and J. E. Lenox, "The Epoch State Navigation Filter," Journal of Guidance and Control, Volume 1, Number 5, September-October 1978.

31. Baudet, G. M., R. P. Brent, and H. T. Kung, "Parallel Execution of a Sequence of Tasks on an Asynchronous Multiprocessor," June 1977, AD-A042505, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

32. Bell, C. G., R. Chen and S. Rege, "Effect of Technology on Near Term Computer Structures," Computer, March-April 1972, pp. 29-38.

33. Brent, R. P. and H. T. Kung, "Fast Algorithms for Manipulating Formal Power Series," Journal of the ACM, Volume 25, Number 4, October 1978, pp. 581-595.

34. Broucke, R. A., "Algorithm 446: Ten Subroutines for the Manipulation of Chebyshev Series [C1]," Communications of the ACM, Volume 16, Number 4, April 1973, pp. 254-256.

35. Broucke, R. A., "Construction of Rational and Negative Powers of a Formal Series," Communications of the ACM, Volume 14, Number 1, January 1971, pp. 32-35.

36. Broucke, R. A., "On the Matrizant of the Two-Body Problem," Astronomy and Astrophysics, Volume 6, 1970, pp. 173-182.

37. Broucke, R. A., "Perturbations in Rectangular Coordinates by Iteration," Celestial Mechanics, Volume 1, Number 1, June 1969, pp. 110-126.

38. Broucke, R., and P. Cefola, "On the Equinoctial Orbital Elements," Celestial Mechanics, Volume 5, Number 3, 1972, pp. 303-310.

39. Brouwer, D., "Solution of the Problem of Artificial Satellite Theory Without Drag," The Astronomical Journal, Volume 64, Number 1274, November 1959.

40. Brouwer, D., and G. Hori, "Theoretical Evaluation of Atmospheric Drag Effects in the Motion of an Artificial Satellite," The Astronomical Journal, Volume 66, Number 5, June 1961.

41. Brouwer, D., and G. M. Clemence, Methods of Celestial Mechanics, Academic Press, New York, 1961.

42. Bucy, R. S., and K. D. Senne, New Frontiers in Nonlinear Filtering, Lincoln Laboratory Technical Note 1978-16, Lincoln Laboratory, Lexington, MA, 26 May 1978, also ESD-TR-78-98.

43. Burkhead, F., Parallel Processing of Recursive Functions, M.S. Thesis, Naval Postgraduate School, Monterey, CA, June 1977.

44. Calahan, D. A., "Vector Processing in Simulation," AD-A041747, University of Michican, Ann Arbor, Michigan, April 1977.

45. Carpenter, L., Planetary Perturbations in Chebyshev Series, NASA Technical Note TN D-3168, Goddard Space Flight Center, Greenbelt, Maryland, January 1966.

46. Cappellari, J. O., C. E. Velez, and A. J. Fuchs, Editors, "Mathematical Theory of the Goddard Trajectory Determination System," X-582-76-77, Goddard Space Flight Center, Geenbelt, Maryland, April 1976.

47. Cefola, P. J., "A Recursive Formulation for the Tesseral Disturbing Function in Equinoctial Variables," No. 76-839, AIAA/AAS Astrodynamics Conference, San Diego, Calif., August 1976.

48. Cefola, P., and R. Broucke, "On the Formulation of the Gravitational Potential in Terms of Equinoctial Variables," No. 75-9, AIAA Aerospace Sciences Meeting, Pasadena, Calif., January 1975.

49. Cefola, P., A. Long, and G. Holloway, "The Long-Term Prediction of Artificial Satellite Orbits," No. 74-170, AIAA Aerospace Sciences Meeting, Washington, D.C., February 1974.

50. Chan, F. K., H. Siddalingaiah, and P. S. Desai, The Hamilton Orbit Generator, 3000-09000-OITM, Computer Sciences Corporation, September 1974.

51. Chen, S., and D. J. Kuck, "Time and Parallel Processor Bounds for Linear Recurrence Systems," IEEE Transactions on Computers, Volume C-24, Number 7, July 1975, pp. 701-717.

52. Clenshaw, C. W., and H. J. Norton, "The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series," Computer Journal, Volume 6, 1963, pp. 88-92.

53. Coddington, E. A., and N. Levinson, Theory of Ordinary Differential Equations, McGraw-Hill Book Co., Inc., New York, 1955.

54. Collins, S. K., "Long Term Prediction of Super-Synchronous Orbits Over Very Long Time Spans," pending Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

55. Conte, S. D., and C. de Boor, Elementary Numerical Analysis: An Algorithmic Approach, McGraw-Hill Book Co., Inc., 1965.

56. Corio, A. J., "The Use of Chebyshev Polynomials for Satellite Ephemerides," COMSAT Technical Review, Volume 3, Number 2, Fall 1973, pp. 411-418.

57. Covault, Craig, "NASA Plans New Space Missions," Aviation Week and Space Technology, April 17, 1978, pp. 14-16.

58. Covault, Craig, "Potential in Space Awaits Funds," Aviation Week and Space Technology, March 21, 1977, pp. 59-61.

59. Dahlquist, G., A. Björck, and N. Anderson (Translator), Numerical Methods, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

60. Danby, J. M. A., Fundamentals of Celestial Mechanics, The Macmillan Co., New York, 1962.

61. Deprit, A., H. Pickard and W. Poplarchek, "Compression of Ephemerides by Discrete Chebyshev Approximations," NAVIGATION: Journal of the Institute of Navigation, Volume 26, Number 1, Spring 1979, pp. 1-11.

62. DeWolf, J. B., "Requirements Specification and Preliminary Design for Real-Time Systems," presented at COMPSAC '77, 8-11 November 1977, Chicago, Illinois.

63. Doggett, L. E., "The Use of Chebyshev Polynomials in Generating a High Precision Theory of Mars," Recent Advances in Engineering Science, Volume 8, pp. 439-446, also A77-39922.

64. Dooling, Dave, "Space Factories in 1997, Part 1," Spaceflight, Volume 19, Number 10, October 1977, pp. 342-347.

65. Dooling, Dave, "Space Factories in 1997, Part 2," Spaceflight, Volume 19, Number 11, November 1977, pp. 390-392.

66. Dowd, Douglas L., "Adaptive Estimation of Atmospheric Drag on Near Earth Satellites," Ph.D. Dissertation, The University of Texas at Austin, Austin, Texas, December 1977.

67. Dyer, P., and S. R. McReynolds, The Computation and Theory of Optimal Control, Academic Press, New York, 1970.

68. Ekelund, J. E., "The JPL Orbit Determination Software System," No. 79-111, AAS/AIAA Astrodynamics Conference, Provincetown, MA, June 1979.

69. Edgard, Robert, "Landsats -- Spacecraft Exploring Earth," Spaceflight, Volume 17, Numbers 8 and 9, August-September 1975, pp. 312-322.

70. Edwards, J. A., "Investigation of the Epoch State Filter," M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA, February 1972.

71. Elson, B. M., "Computer Development Time Being Cut," Aviation Week and Space Technology, February 26, 1979, pp. 67-70.

72. Elson, Benjamin M., "Space-Based Solar Power Study Near Completion," Aviation Week and Space Technology, September 19, 1977, pp. 58-69.

73. Elson, B. M., "Supercomputer Sought to Aid Design," Aviation Week and Space Technology, April 24, 1978, pp. 123-128.

74. Enslow, P. H., Jr., "Multiprocessor Organization -- A Survey," ACM Computing Surveys, Volume 9, Number 1, March 1977, pp. 103-129.

75. Feagin, T., Description of the Numerical Integration of the Equations of Motion of a Space Vehicle Using Chebyshev Series, Goddard Space Flight Center, Report X582-73-176, June 1973, also N73-22613.

76. Feng, T., Editor, Parallel Processing: Proceedings of the Sagamore Computer Conference, August 20-23, 1974, Springer-Verlag, New York, 1975.

77. Flanders, P. M., D. J. Hunt, S. F. Reddaway, and D. Parkinson, "Efficient High Speed Computing with the Distributed Array Processor," High Speed Computing and Algorithm Organization, D. J. Kuck, et al., Editors, Academic Press, Inc., New York, 1977.

78. Flynn, Michael J., "Some Computer Organizations and Their Effectiveness," IEEE Transactions on Computers, Volume C-21, Number 9, September 1972, pp. 948-960.

79. Flynn, M. J., "Very High-Speed Computing Systems," Proceedings of the IEEE, Volume 54, December 1966, pp. 1901-1909.

80. Fox, L., and I. B. Parker, Chebyshev Polynomials in Numerical Analysis, Oxford University Press, London, 1968.

81. Fung, L., "A Massively Parallel Processing Computer," High Speed Computer and Algorithm Organization, D. J. Kuck, et al., Editors, Academic Press, Inc., New York, 1977.

82. Green, A. J., "Orbit Determination and Prediction Processes for Low Altitude Satellites," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, December 1979.

83. Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Co., Inc., New York, 1962.

84. Henrici, P., Discrete Variable Methods in Ordinary Differential Equations, John Wiley and Sons, Inc., New York, 1962.

85. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill Book Co., Inc., New York, 1974.

86. Hujsak, R. S., Fortran SGP/SGP4/DP4 Program Description, Aerospace Data Analysis Program Number 77-1, Office of Astrodynamics, HQ Aerospace Defense Command, Peterson AFB, Colorado, 6 April 1977.

87. Kaufman, H., Parallel Computation for Developing Nonlinear Control Procedures, Interim Scientific Report For AFOSR Grant No. AFOSR 77-3418, Sept. 1, 1977-Sept. 30, 1978, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, New York.

88. Deleted

89. Keller, J. D., and A. Jameson, "Preliminary Study of the Use of the STAR-100 Computer for Transonic Flow Calculations," presented at AIAA 16th Aerospace Sciences Meeting, 16-18 January 1978, Huntsville, Alabama.

90. Kenden, A., "U.S. Reconnaissance Satellite Programmes," Spaceflight, Volume 20, Number 7, July 1978, pp. 243-262.

91. Klass, Philip J., Secret Sentries in Space, Random House, Inc., New York, 1970.

92. Koskela, P. E., Astrodynamic Analysis for the Advanced Orbit/Ephemeris Subsystem, Aeronutronic Publication No. U-4180, Philco-Ford Corp., Newport Beach, CA, 1 September 1967.

93. Kozai, Y., "Analytical Orbital Theories for Satellites," The Use of Artificial Satellites for Geodesy and Geodynamics, Proceedings of the International Symposium on the Use of Artificial Satellites for Geodesy and Geodynamics, Athens, Greece, May 1973, pp. 237-242.

94. Kozai, Y., "The Motion of a Close Earth Satellite," The Astronomical Journal, Volume 64, Number 1274, November 1959, pp. 367-277.

95. Kuck, D. J., "A Survey of Parallel Machine Organization and Programming," ACM Computing Surveys, Volume 9, Number 1, March 1977, pp. 29-59.

96. Kuck, D. J., D. H. Lawrie and A. H. Sameh, Editors, High Speed Computer and Algorithm Organization, Academic Press, Inc., New York, 1977.

97. Kuester, J. L., and J. H. Mize, Optimization Methods with Fortran, McGraw-Hill Book Co., Inc., New York, 1973.

98. Kung, H. T., "Synchronized and Asynchronous Parallel Algorithms for Multiprocessors," Algorithms and Complexity: New Directions and Recent Results, J. F. Traub, Editor, Academic Press, Inc., New York, 1976, pp. 153-200.

99. Kung, H. T., and J. F. Traub, "All Algebraic Functions Can Be Computed Fast," Journal of the ACM, Volume 25, Number 2, April 1978, pp. 245-260.

100. Kurzhals, P. R., "Highlights 1979 -- Computer Systems," Astronautics and Aeronautics, Volume 17, Number 12, December 1979, pp. 44-58.

101. Lane, Max H., "The Development of an Artificial Satellite Theory Using a Power-Law Atmospheric Density Representation," AIAA Paper No. 65-35, AIAA 2nd Aerospace Sciences Meeting, New York, 1965.

102. Lane, M. H., P. M. Fitzpatrick and J. J. Murphy, On the Representation of Air Density in Satellite Deceleration Equations by Power Functions with Integral Exponents, Technical Documentary Report No. APGC-TDR-62-15, Air Proving Ground Center, Eglin AFB, Florida, March 1962.

103. Larson, R. E., and E. Tse, "Parallel Processing Algorithms for Modal Trajectory Estimation," Thirteenth Joint Automatic Control Conference of the American Automatic Control Council, Stanford, California, August 16-18, 1972.

104. Long, A. C., and W. D. McClain, "Optimal Perturbation Models for Averaged Orbit Generation," No. 76-815, AIAA/AAS Astrodynamics Conference, San Diego, CA, August 1976.

105. Long, A., W. D. McClain and P. J. Cefola, Mathematical Specifications for the Earth Satellite Mission Analysis Program (ESMAP), June 1975, CSC/SD-75/6025, Computer Sciences Corporation, Silver Spring, Maryland, NASA Contract NAS5-1199.

106. Lutzky, D., and C. Uphoff, "Short Periodic Variations and Second-Order Numerical Averaging," No. 75-11, AIAA Aerospace Sciences Meeting, Pasadena, CA, January 1975.

107. Lyddane, R. H., "Small Eccentricities or Inclinations in the Brouwer Theory of the Artificial Satellite," The Astronomical Journal, Volume 68, Number 8, October 1963, pp. 555-558.

108. Major, P. E., and F. R. Hoots, Primary General Perturbation Theories Used in the Space Defense Center, Analysis Memorandum 75-2, Office of Astrodynamic Applications, HQ Fourteenth Aerospace Force, Ent AFB, Colorado, 7 May 1975.

109. Martin, H. G., "A Discourse on a New Super Computer, PEPE," High Speed Computer and Algorithm Organization, D. J. Kuck, et al., Editors, Academic Press, New York, 1977.

110. McClain, W. D., A Recursively Formulated First-Order Semianalytic Artificial Satellite Theory Based on the Generalized Method of Averaging, Volume 1, November 1977, CSC/TR-77/6010, Computer Sciences Corporation, Silver Spring, Maryland, NASA-CR-156782, N78-28147, 1978.

111. McClain, W. D., A Recursively Formulated First-Order Semianalytic Artificial Satellite Theory Based on the Generalized Method of Averaging, Volume 2, May 1978, CSC/TR-78/6001, Computer Sciences Corporation, Silver Spring, Maryland, NASA-CR-156783, N78-28148, 1978.

112. McClain, W. D., et al., "Development and Evaluation of a Hybrid Averaged Orbit Generator," No. 78-1382, AIAA/AAS Astrodynamics Conference, Palo Alto, CA, August 1978.

113. McCuskey, S. W., Introduction to Celestial Mechanics, Addison-Wesley, Reading, MA, 1963.

114. Miller, R. E., "A Comparison of Some Theoretical Models of Parallel Computation," IEEE Transactions on Computers, Volume C-22, Number 8, August 1973, pp. 710-717.

115. Miranker, W. L., "A Survey of Parallelism in Numerical Analysis," SIAM Review, Volume 13, Number 4, October 1971, pp. 524-547.

116. Miranker, W. L., and W. M. Liniger, "Parallel Methods for the Numerical Integration of Ordinary Differential Equations," Mathematics of Computation, Volume 21, 1967, pp. 303-320.

117. Morgan, W. L., "Satellite Characteristics Summary," Acta Astronautica, Volume 5, 1978, pp. 455-466.

118. Musen, P., and L. Carpenter, "On the General Planetary Perturbations in Rectangular Coordinates," Journal of Geophysical Research, 68(9), 1 May 1963, pp. 2727-2734.

119. Myers, K. A., "Filtering Theory Methods and Applications to the Orbit Determination Problem for Near-Earth Satellites," Ph.D. Dissertation, University of Texas, Austin, Texas, January 1974.

120. Noe, J. D., and G. J. Nutt, "Macro E-Nets for Representation of Parallel Systems," IEEE Transactions on Computers, Volume C-22, Number 8, August 1973, pp. 718-727.

121. Nutt, G. J., "A Parallel Processor Operating System Comparison," IEEE Transactions on Software Engineering, Volume SE-3, Number 6, November 1977, pp. 467-475.

122. O'Toole, J. W., Celest Computer Program for Computing Satellite Orbits, Naval Surface Weapons Center, Dahlgren Laboratory, Dahlgren, Virginia, NSWC/DL TR-3565, October 1976.

123. Pirard, T., "Chinese 'Secrets' Orbiting the Earth," Spaceflight, Volume 19, Number 10, October 1977, pp. 355-361.

124. Pollard, H., Mathematical Introduction to Celestial Mechanics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1966.

125. Pon, W. Y., "Mathematical Theory for the Advanced Orbit Determination and Ephemeris Generation System," September 1973, Data Dynamics, Inc., Cupertino, CA.

126. Principato, R., "A Survey of Specification Techniques for Parallel Processes," 25 January 1977, C-4813, C. S. Draper Laboratory, Inc., Cambridge, MA.

127. Prislin, R. H., D. C. Walker, and R. J. Mercer, TRACE66 Trajectory Analysis and Orbit Determination Program -- Volume I: General Program Objectives, Description and Summary, Aerospace Report Number TR-0059(9320) -1, Volume 1, The Aerospace Corporation, Los Angeles, CA, 15 August 1971, also Air Force Report No. SAMSO-TR-71-141, Volume I.

128. Ramamoorthy, C. V., and H. F. Li, "Pipeline Architecture," ACM Computing Surveys, Volume 9, Number 1, March 1977, pp. 61-102.

129. Russell, R. M., "The CRAY-1 Computer System," Communications of the ACM, Volume 21, Number 1, January 1978, pp. 63-72.

130. Salkeld, Robert, and Donald W. Patterson, "Space Transportation -- New Headings for the Future," Astronautics and Aeronautics, Volume 16, Number 4, April 1978, pp. 28-31.

131. Sameh, A. H., and R. P. Brent, "Solving Triangular Systems on a Parallel Computer," SIAM Journal of Numerical Analysis, Volume 14, Number 6, December 1977, pp. 1101-1113.

132. Sameh, A. H., and D. J. Kuck, "On Stable Parallel Linear System Solvers," Journal of the ACM, Volume 25, Number 1, January 1978, pp. 81-91.

133. Deleted

134. Sheldon, C. S., "Soviet Space Activities in 1977," Air Force Magazine, Volume 61, Number 3, March 1978, pp. 74-77.

135. Sherrill, T. J., "Development of a Satellite Drag Theory Based on the Vinti Formulation," Ph.D. Dissertation, University of Calif., Berkeley, CA, 1966.

136. Smart, Jacob E., "Strategic Implications of Space Activities," Strategic Review, Volume II, Number 4, Fall 1974, pp. 19-24.

137. Spaceflight, "Extra-Terrestrial Communities," Volume 18, Number 4, April 1976, pp. 130-134.

138. Stokes, R. A., "Burroughs Scientific Processor," High Speed Computer and Algorithm Organization, D. J. Kuck, et al., Editors, Academic Press, Inc., New York, 1977.

139. Straeter, T. A., A Parallel Variable Metric Optimization Algorithm, NASA Langley Research Center, Hampton, VA, NASA TN D-7329, December 1973.

140. Tapley, B., A. R. Neto and B. E. Schutz, "Orbit Determination in the Presence of Atmospheric Errors," COSPAR IAUIUTAM Symposium, Sao Paulo, Brazil, Satellite Dynamics, Springer-Verlag, 1975.

141. Taylor, T. D., and G. F. Widhopf, "A Trend Toward Inexpensive Computer Simulations," Astronautics and Aeronautics, Volume 17, Number 4, April 1979, pp. 34-41.

142. Thurber, K. J., "Parallel Processor Architectures -- Part 1: General Purpose Systems," Computer Design, Volume 18, Number 1, January 1979, pp. 89-97.

143. Thurber, K. J., "Parallel Processor Architectures -- Part 2: Special Purpose Systems," Computer Design, Volume 18, Number 2, February 1979, pp. 103-114.

144. Thurber, K. J., and L. D. Wald, "Associative and Parallel Processors," ACM Computing Surveys, Volume 7, Number 4, December 1977, pp. 215-255.

145. Toong, H. D., Multi-Microprocessor Systems, Center for Information Systems Research, Massachusetts Institute of Technology, Cambridge, MA.

146. Travassos, R., and H. Kaufman, Parallel Algorithms for Solving Nonlinear Two-Point Boundary Value Problems Which Arise in Optimal Control, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, New York.

147. Travassos, R., and H. Kaufman, The Impact of Parallel Computers on the Design of Nonlinear Flight Controllers, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, New York.

148. Uphoff, C., "Numerical Averaging in Orbit Prediction," AIAA Journal, Volume 11, Number 11, November 1973, pp. 1513-1516.

149. Vinti, J. P., "New Method of Solution for Unretarded Satellite Orbits," Journal of Research of the National Bureau of Standards, Volume 62B, Number 2, October-December 1959.

150. Vinti, J. P., "Theory of an Accurate Intermediary Orbit for Satellite Astronomy," Journal of Research of the National Bureau of Standards, Volume 65B, Number 3, July-September 1961.

151. Vinti, J. P., "Zonal Harmonic Perturbations of an Accurate Reference Orbit of an Artificial Satellite," Journal of Research of the National Bureau of Standards, Volume 67B, Number 4, October-December 1963.

152. Vinti, J. P., "Inclusion of the Third Zonal Harmonic in an Accurate Reference Orbit of an Artificial Satellite," Journal of Research of the National Bureau of Standards, Volume 70B, Number 1, January-March 1966.

153. Watson, J. S., G. D. Mistretta, and N. L. Bonavito, "An Analytic Method to Account for Drag in the Vinti Satellite Theory," Celestial Mechanics, Volume 11, Number 2, March 1975, pp. 145-177.

154. Yau, S. S., and H. S. Fung, "Associative Processor Architecture -- A Survey," ACM Computing Surveys, Volume 9, Number 1, March 1977, pp. 3-27.

155. Zavaleta, E. L., Goddard Trajectory Determination System Design Manual, March 1975, CSC/3000-27900-02TM, Computer Sciences Corporation, Silver Spring, Maryland.

156. Zeis, E. G., "A Computerized Algebraic Utility for the Construction of Nonsingular Satellite Theories," M.S. Thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1978.

157. Wu Lian-da, Wang Chang-bin and Tong-Fu, "A Semi-Numerical Method for Computing the Second-Order Perturbations of Artificial Satellites," Chinese Astronomy, 3 (1979), pp. 238-257.

BIOGRAPHY

Jeffrey Scott Shaver was born on 1 October 1948 in Coeur d'Alene, Idaho to Jack R. and Genevieve C. Shaver. He attended the public school system in his hometown, graduating from Coeur d'Alene High School in 1966. During his high school senior year, he was selected as a National Merit Scholarship recipient. In June 1966, he entered the United States Air Force Academy. He graduated in June 1970 as a Distinguished Graduate, majoring in Aeronautical Engineering and was commissioned as a Second Lieutenant in the U. S. Air Force. He attended Purdue University from June 1970 to January 1971, receiving a Master of Science degree in Aeronautics, Astronautics and Engineering Sciences. From January 1972 to May 1976 he served as a Software Systems Manager at the Headquarters, Air Force Satellite Control Facility (AFSCF), Space and Missile Systems Organization (SAMSO), Los Angeles, California. These duties included technical management of a large contractor effort for the design and development of real-time command, control, communication and telemetry processing computer programs in support of the world-wide AFSCF network of satellite tracking stations. His military decorations include the Airman's Medal (for valor), the Meritorious Service Medal and the Air Force Commendation Medal. In June 1976 he entered the Massachusetts Institute of Technology. He married Claire M. Knapp in May 1977. At present he is completing the requirements for the Doctor of Philosophy degree in the M.I.T. Interdepartmental Instrumentation Program. He has two step children, Jennifer and Jaime.