T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DİFERENSİYEL DENKLEM SİSTEMLERİNİN NÜMERİK İNTEGRASYONUNDA ADIM GENİŞLİĞİ STRATEJİLERİ

Gülnur ÇELİK KIZILKAN

DOKTORA TEZİ

MATEMATİK ANABİLİM DALI

Konya, 2009

ÖZET

Doktora Tezi

DİFERENSİYEL DENKLEM SİSTEMLERİNİN NÜMERİK İNTEGRASYONUNDA ADIM GENİŞLİĞİ STRATEJİLERİ

Gülnur ÇELİK KIZILKAN

Selçuk Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Danışman : Doç. Dr. Kemal AYDIN

2009, 73 + viii Sayfa

Jüri: Prof. Dr. Şaziye YÜKSEL
Prof. Dr. Dursun TAŞÇI
Prof. Dr. Haydar BULGAK
Doç. Dr. Kemal AYDIN
Yrd. Doç. Ayşe NALLI

Bu çalışmada, $X'(t) = AX(t)$ ve $X'(t) = AX(t) + \varphi(t, X)$ denklem sistemlerinin $[t_0 , T]$ aralığındaki nümerik integrasyonunda her bir adımda oluşan lokal hata kullanıcının belirlediği hata seviyesinden küçük olacak şekilde adım genişliği belirleyen stratejiler elde edilmiştir. Bu stratejilere göre seçilen adım genişlikleri ile nümerik çözüm hesaplayan algoritmalar verilmiştir. Algoritmalar nümerik örneklerle desteklenmiştir.

Anahtar Kelimeler: Cauchy problemi, nümerik integrasyon, adım genişliği stratejisi, lokal hata.

ABSTRACT

Ph.D. Thesis

STEP SIZE STRATEGIES ON THE NUMERICAL INTEGRATION OF THE SYSTEMS OF DIFFERENTIAL EQUATIONS

Gülnur ÇELİK KIZILKAN

Selçuk University
Graduate School of Natural and Applied Sciences
Department of Mathematics

Supervisor: Assoc.Prof.Dr. Kemal AYDIN

2009, 73 + viii Pages

Jury: Prof. Dr. Şaziye YÜKSEL
Prof. Dr. Dursun TAŞÇI
Prof. Dr. Haydar BULGAK
Assoc. Prof. Dr. Kemal AYDIN
Asist. Prof. Dr. Ayşe NALLI

In this study, the step size strategies have been obtained such that the local error is smaller than desired error level in the numerical integration of $X'(t) = AX(t)$ and $X'(t) = AX(t) + \varphi(t, X)$ equation systems in interval $[t_0 , T]$. The algorithms have been given that calculate step sizes and numerical solutions according to these strategies and numerical solutions. The algorithms have been supported by the numerical examples.

Key Words: Cauchy problem, numerical integration, step size strategy, local error.

ÖNSÖZ

Bu çalışma, Selçuk Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda **Doktora Tezi** olarak hazırlanmıştır.

Doktora çalışmasının her aşamasında büyük sabır ve titizlik gösteren ve yardımlarını esirgemeyen danışman hocam Sayın Doç. Dr. Kemal AYDIN'a, değerli katkılarını gördüğüm tez izleme komitesi üyeleri Prof. Dr. Şaziye YÜKSEL ve Prof. Dr. Dursun TAŞÇI 'ya, bu çalışmanın hazırlanmasında manevi katkılarıyla bana destek olan aileme teşekkür ederim.

Gülnur ÇELİK KIZILKAN

KULLANILAN SEMBOLLER

$\boldsymbol{R}$: Reel sayılar kümesi

$\boldsymbol{R^N}$: N boyutlu Öklit uzayı

$C^m(\Omega)$: Ω bölgesinde m. mertebeden türevlenebilen fonksiyonlar ailesi

t_i : Grid noktaları

h_i : i inci adımdaki adım genişliği

$X(t_i)$: $X'(t) = F(t, X)$, $X(t_0) = X_0$ Cauchy probleminin tam çözümünün t_i noktasındaki değeri

$Z(t)$: $Z'(t) = F(t, Z)$, $Z(t_{i-1}) = Y_{i-1}$ Cauchy probleminin $[t_{i-1}, t_i)$ aralığındaki çözümü

Y_i : Cauchy probleminin t_i noktasındaki nümerik metot kullanarak elde edilen yaklaşık çözümünün değeri

LE_i : i inci adımda oluşan lokal hata vektörü

$\| LE_i \|$: i inci adımda oluşan lokal hata

$\|x\|$: x vektörünün *Öklit* normu

$\|A\|$: A matrisinin normu

α : $A = (a_{ij}) \in \boldsymbol{R}^{N \times N}$ olmak üzere $\alpha = \max\limits_{1 \le i, j \le N} | a_{ij} |$ sayısı

β_{i-1} : $\max\limits_{1 \le j \le N}(\sup\limits_{t_{i-1} < \tau_i < t_i} | z_j(\tau_i) |) \le \beta_{i-1}$ olacak şekildeki reel sayı

γ_{i-1} : $X'(t) = AX(t) + \varphi(t, X)$, $X(t_0) = X_0$ Cauchy problemi için $\max\limits_{1 \le j \le N}(\sup\limits_{t_{i-1} < \tau_i < t_i} | \varphi_j(\tau_i, z(\tau_i)) | \le \gamma_{i-1}$ şartını sağlayan reel sayı

ζ_{i-1} : $X'(t) = AX(t) + \varphi(t, X)$, $X(t_0) = X_0$ Cauchy problemi için $\max\limits_{1 \le j \le N}(\sup\limits_{t_{i-1} < \tau_i < t_i} | \dfrac{d\varphi_j}{dt}(\tau_i, z(\tau_i)) | \le \varsigma_{i-1}$ şartını sağlayan reel sayı

h^* : Pratik adım genişliği parametresi

δ_L : İstenilen hata seviyesi

İÇİNDEKİLER

ÖZET .....……...……ii
ABSTRACT .....……...iii
ÖNSÖZ ……..……...iv
KULLANILAN SEMBOLLER ……v

1. GİRİŞ
1.1. Literatür Özeti ……1
1.2. Problemin Tanıtımı ……6
1.3. Tezin Yapısı ……7

2. BİRİNCİ MERTEBEDEN CAUCHY PROBLEMİ İÇİN ADIM GENİŞLİĞİ STRATEJİSİ
2.1. Lipschitz şartı, Lipschitz sabiti ……...9
2.2. Lokal Hata ……..9
2.3. Euler Metodu ve Hata Analizi ……..10
2.4. Adım Genişliği Seçimi ve Pratik Adım Genişliği ……10
2.5. Adım Genişliği Kontrolü (K) ……11
2.6. Hata Analizi Tabanlı Adım Genişliği İçin Algoritma ……...12

3. DİFERENSİYEL DENKLEM SİSTEMLERİ İÇİN ADIM GENİŞLİĞİ STRATEJİLERİ
3.1. Kayan Noktalı Sayılar ve Hesaplamalarda Kayan Nokta Aritmetiğinin Etkisi……...……...15
3.2. $F(t, X) = AX$ Durumunda Adım Genişliği Stratejileri ve Algoritmaları.….16
3.2.1. Strateji ve Algoritma …….....18

3.2.2. Strateji ve Algoritma ….……...23
3.2.3. Stratejisi ve Algoritma ……30
3.2.4. Strateji ve Algoritma ……...34
3.2.5. Stratejilerin *m*. Mertebeden Cauchy Problemlerine Uygulanması.... 39
3.2.6. Değerlendirme ……...41
3.2.7. Ek 1: Algoritmalar Prosedürler ...……..46
3.3. $F(t, X) = AX + \varphi(t, X)$ Durumunda Adım Genişliği Stratejileri ve Algoritmaları…...……..49
3.3.1. Strateji ve Algoritma …….....50
3.3.2. Strateji ve Algoritma ….……...54
3.3.3. Stratejisi ve Algoritma ……59
3.3.4. Değerlendirme…….....65
3.3.5. Ek 2: Örnekler İçin Prosedürler ……66

4. KAYNAKLAR ……72

1. GİRİŞ

Kütle transferi, ısı transferi, elektrik devresi, popülâsyonun artışı, av-avcı problemleri, kimyasal reaksiyonlar, astronomik nesnelerin hareketi gibi birçok problemin matematiksel modeli diferensiyel denklemlerle ifade edilir. Bu nedenle diferensiyel denklemlerin çözümünü hesaplamak önemlidir. Bazı denklemleri analitik olarak çözmek mümkün değildir. Mümkün olsa bile, analitik çözüme ulaşmak için birçok ön araştırma yapmak gerekebilir. Bu nedenle, analitik çözüm yerine denklemlerin nümerik çözümleri hesaplanır. Günümüz teknolojisinde, hemen hemen her alanda olduğu gibi, diferensiyel denklemlerin çözümleri için de bilgisayar kullanılmaktadır.

Bilgisayarla hesaplamalarda, kullanıcıdan kullanıcıya değişik öncelikler söz konusu olsa da, istenilen yakınlıkta çözüm hesaplamak her zaman önemli olmuştur. Bazen hızda önemli olabilmektedir. Genelde, kabul edilebilir bir hata seviyesi ve kısa hesaplama süresi birlikte nümerik hesaplamayı etkin kılar. Çözümün istenilen yakınlıkta ve hızlı hesaplanmasını etkileyen faktörlerden birisi *adım genişliği*dir. Hesaplamalarda, küçük adım genişliklerinin seçilmesi nümerik çözümün analitik çözüme daha yakın olmasını sağlar. Fakat, küçük adım genişlikleri hem çözümün hesaplama süresini hem de hesaplama hatalarını artırır. Dolayısıyla; adım genişlikleri problemin yapısına ve kullanılan nümerik metoda göre seçilmelidir.

1.1. Literatür Özeti

$$x' = f(t,x)\ ,\ x(t_0) = x_0\ ,\ t_0 \le t \le T \tag{1.1}$$

Cauchy problemini ele alalım. Birçok uygulamada $f(t,x)$ fonksiyonu açık (explicit) olarak verilmez. $f(t,x)$ açık olarak ifade edilse bile, Cauchy problemini analitik olarak çözmek mümkün olmayabilir. Bu nedenle nümerik integrasyona başlamadan önce;

1. Problemin gerçekten bir çözümünün olup olmadığı,

2. Eğer bir çözümü varsa, bu çözümün tek mi yoksa sonsuz çoklukta mı olduğu soruları akla gelir. Problemin çözümünün ne olduğunu bilmeden, sadece çözümün bir tek şekilde var olduğunu garanti eden teoremler bu sorulara cevap verir. Picard teoremi, (1.1) Cauchy probleminin varlık ve teklik teoremidir (Rao 1979, Brock ve Malliaris 1989, Estep 2002).

(1.1) Cauchy probleminin çözümünün varlık ve tekliği ile ilgili kavramlardan birisi de Lipschitz şartıdır (Rao 1979, Miranker 1981, Brock ve Malliaris 1989, Shampine ve ark. 1996). Lipschitz şartı, bilim ve mühendislikte, matematik analizde ve matematik modellemede sıklıkla kullanılır (Estep 2002). Sistemler için de Lipschitz şartı tanımlanmıştır (Gear 1971, Rao 1979, Shampine ve ark. 1996).

(1.1) Cauchy probleminin nümerik metotla hesaplanmasından kaynaklanan lokal hatası; $[t_{i-1}, t_i)$ aralığında $z(t_{i-1}) = y_{i-1}$ başlangıç şartını sağlayan $z' = f(t,z)$ probleminin çözümü $z(t)$ olmak üzere

$$|LE_i| = |y_i - z(t_i)|$$

şeklinde tanımlanır (Shampine ve ark. 1996, Loan 2000, Aydın ve ark. 2001). Sabit adım genişlikli Euler metodu için lokal hata analizi incelenmiştir (Loan 2000, Aydın ve ark. 2001).

(1.1) Cauchy probleminin çözümünün var ve tek olduğu bilinen bir bölge üzerinde, problemin nümerik integrasyonu için üç farklı adım genişliği stratejisi;

1. Lokal hata iterasyonun her bir adımında verilen δ_L hata seviyesinden küçük kalacak şekilde Euler metodu için,

$$h_i < \left(\frac{2\delta_L}{\max\limits_{\tau \in [t_{i-1}, t_i]} |f''(\tau)|}\right)^{\frac{1}{2}} \quad , (i = 1, 2, \ldots, n)$$

şeklinde (Çelik Kızılkan 2004);

2. $D = \{(t,x): |t - t_0| \le a, |x - x_0| \le b\}$ bölgesi üzerinde çözümü var ve tek olmak üzere Picard teoremi ard arda uygulanarak,

$$h_i = \min\{a, b/M\}$$

şeklinde (Çelik Kızılkan 2004);

3. (1.1) Cauchy probleminin D bölgesi üzerinde çözümünün varlığı bilinmiyorsa

- y_i; i-inci adımda elde edilen nümerik çözüm,
- $z(t)$; $z(t_{i-1}) = y_{i-1}$ şartını sağlayan $z' = f(t,z)$ probleminin çözümü,
- b_{i-1}; $| z - y_{i-1} |$ hatasının üst sınırı,
- $b_{0i-1} = \min\{b_{0i-2}, b_{i-1}\}$,
- $D_{i-1} = \{(t,z) : | t - t_{i-1} | \leq a, | z - y_{i-1} | \leq b_{0i-1}\}$,
- M_i; D_{i-1} bölgesi üzerinde $f(t,z)$ nin üst sınırı

olmak üzere,

$$h_i = \min\{a, b_{0i-1} / M_i\}$$

şeklinde verilmiştir (Çelik Kızılkan 2004, Çelik Kızılkan ve Aydın 2006).

Çelik Kızılkan (2004), Çelik Kızılkan ve Aydın (2006)' da verilen değişken adım genişliği stratejileri kullanılarak hem değişken adım genişliklerini hem de Cauchy probleminin yaklaşık çözümü hesap eden algoritmalar da verilmiştir.

2. mertebeden Runge- Kutta metodu için,

$$\max_{\tau \in [t_{i-1}, t_i]} | (f_{tt} + 2f.f_{tx} + f_x.f_t + f.f_x^2 + f^2 f_{xx})(\tau) | \leq M_{t_i}$$

olmak üzere, $\| LE_i \| < \delta_L$ olacak şekilde adım genişliği stratejisi

$$h_i \leq (\frac{12\delta_L}{M_{t_i}})^{\frac{1}{3}}, \text{(i=1, 2, … , } n)$$

şeklinde verilmiştir (Çelik Kızılkan ve Aydın 2005).

Lokal hata diferensiyel denklemin kararlığının bir ölçüsüdür. Lokal hata hızlı bir şekilde artıyorsa, problem kötü şartlıdır (ill- conditioned) ya da kararsızdır (Shampine ve ark. 1996). Bu nedenle, diferensiyel denklemlerin nümerik integrasyonunda adım genişliği seçimi ile ilgili çalışmalarda genellikle lokal hata dikkate alınmıştır.

Diferensiyel denklemlerin nümerik çözümlerinde kullanılan metotlarda, genellikle her bir adımda oluşan hata, kullanıcının belirlediği bir hata seviyesinden küçük kalacak şekilde adım genişliği seçilir. Çünkü hatanın tahmin ve kontrol edilmesi, nümerik çözümlerin güvenirliğini sağlar. Global hata, problem çözülmeden tahmin edilebilse de kontrol edilmesi mümkün olmadığından adım genişliğinin

seçiminde global hatanın dikkate alınması pratik değildir. Bunu yerine, genellikle lokal hatanın tahmini ile adım genişliği seçilmesi tercih edilir (Shampine 2005).

$$y''(t) - 2gy'(t) + (g^2 + w^2)y(t) = f(t, y(t))$$

şeklindeki ikinci mertebeden diferensiyel denklemler için sabit adım genişlikli Chebyshev metodu, değişken adım genişlikli olarak yeniden düzenlenmiş ve adım genişliği lokal kesme hatasının bir fonksiyonu şeklinde elde edilmiştir (Ramos ve Aguiar 2007).

k- adım Störmer- Cowell metodu;

$$y'' = f(x, y)\,,\ y(x_0) = y_0\,,\ y'(x_0) = y_0'$$

Cauchy probleminin nümerik integrasyonunda sıklıkla kullanılan metotlardan biridir. k- adım Störmer- Cowell metodunun değişken adım genişlikli versiyonu elde edilmiş ve adım genişliği lokal hataya bağlı olarak verilmiştir (Ramos ve Aguiar 2005).

Adım genişliği ile ilgili lokal hata dikkate alınmadan elde edilen stratejiler de vardır. (1.1) Cauchy problemi için s- adım Runge- Kutta metodu ve Runge- Kutta-Nyström metodu kullanılarak Picard iterasyonu uygulanmış ve optimal adım genişliği için bir formülasyon verilmiştir (Holsapple ve ark. 2007).

$$\frac{dy}{dt} = f(y)\ ,\quad t \in [t_0, T]$$

problemi için yeni bir değişken adım genişlikli metot elde edilmiş ve elde edilen yeni metot için bir kararlılık fonksiyonu tanımlanmıştır. Adım genişliği seçimi, kararlılık fonksiyonundan elde edilen bir parametreye bağlı olarak elde edilmiştir (Butcher ve Podhaisky 2006).

(1.1) Cauchy probleminin nümerik integrasyonunda değişken adım genişlikli lineer çok adım metotlarının yakınsaklık mertebesinin korunması için her bir adımda metodun katsayılarının yeniden hesaplanması gerekir. Metodun katsayıları, adım genişliklerinin oranlarının fonksiyonları şeklindedir. Bu nedenle, adım genişliklerinin seçimi tek şekilde olmayacağından adım genişliğinin değişimi yerine adım yoğunluk fonksiyonunun değişimi dikkate alınır ve her bir adımda adım genişliği bu adım yoğunluk fonksiyonu yardımıyla verilmiştir (Arévalo ve ark. 2007).

(1.1) Cauchy probleminin nümerik integrasyonunda Euler metodu için τ - kullanıcının belirleyeceği hata seviyesi, $\xi = \min_{[a,b]} |y''(x)| > 0$ ve $H = \sqrt{\frac{2\tau}{\xi}}$ olmak üzere adım genişliği;

$$h_i \approx \sqrt{\frac{\xi}{|y''(x_i)|}} H$$

şeklinde verilmiştir (Shampine ve ark. 1996).

$y = (y_j) \in \boldsymbol{R}^N$, $t \in \Omega$ $f(t,y) = (f_j) \in \boldsymbol{R}^N$ olmak üzere

$$y' = f(t,y),\ y(a) = c$$

diferensiyel denklem sisteminin nümerik integrasyonu için Euler metodu ve 2. mertebeden Runge- kutta metodu verilmiştir. 2. mertebeden Runge- Kutta metodu için *TOL* – kullanıcının belirlediği hata seviyesi, δ - relatif hata olmak üzere adım genişliği, ilk adımda adım genişliği

$$h = \frac{1}{4} \frac{\sum_j |y_j(a)|}{\sum_j |\frac{dy_j}{dt}(a)|}$$

ve daha sonraki adımlarda,

- eğer $\delta \le TOL$ ise $h_{next} = h.\min\{1.5, \sqrt{\frac{TOL}{1.2\delta}}\}$
- $\delta > TOL$ ise $h_{next} = h.\max\{0.1, \sqrt{\frac{TOL}{1.2\delta}}\}$

şeklinde verilmiştir (Dahlquist ve Björck 2008).

Literatürde, Cauchy problemlerinin nümerik integrasyonunda kullanılan Euler metodu, Runge- Kutta gibi iyi bilinen nümerik metotlara kolaylıkla ulaşılabilir (Gear 1971, Hairer ve Wanner 1991, Golub ve Ortega 1992, Shampine ve ark. 1996 v.s.).

Ayrıca; ikinci mertebeden başlangıç değer probleminin nümerik integrasyonu için doğruluğu düşük mertebeli metotlardan daha fazla olan sekizinci mertebeden Runge- Kutta Nyström metodu elde edilmiştir (García ve ark. 2004).

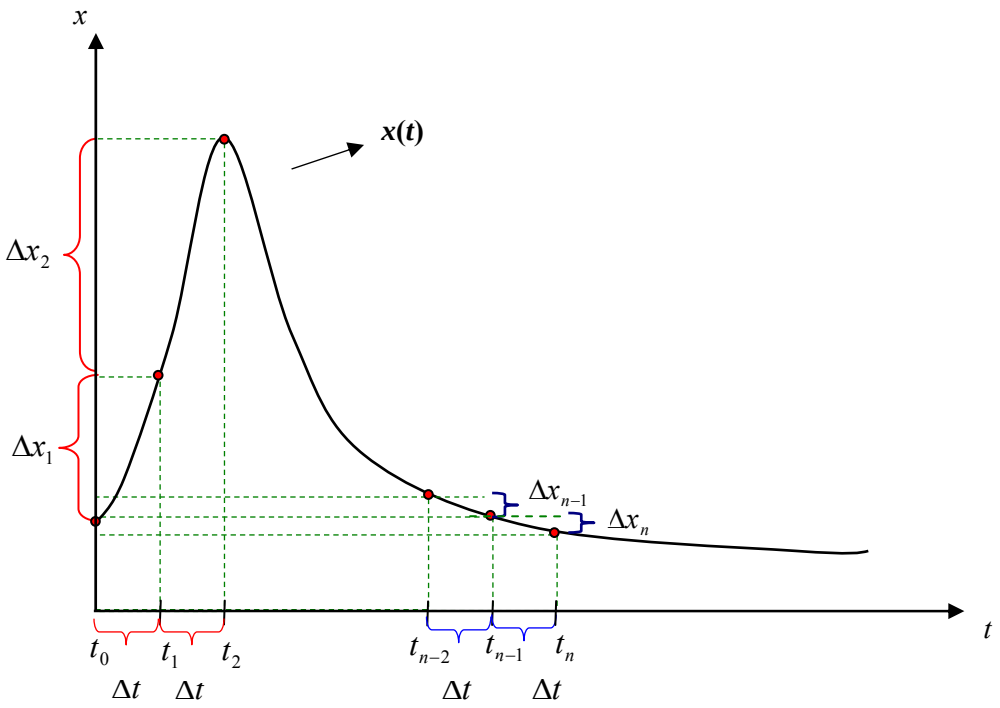
$A \in \boldsymbol{R}^{N \times N}$, $F \in \boldsymbol{R}^N$ – polinom fonksiyon ve $x \in \boldsymbol{R}$ olmak üzere; lineer olmayan

$$y'(x) = Ay(x) + F(y(x))$$

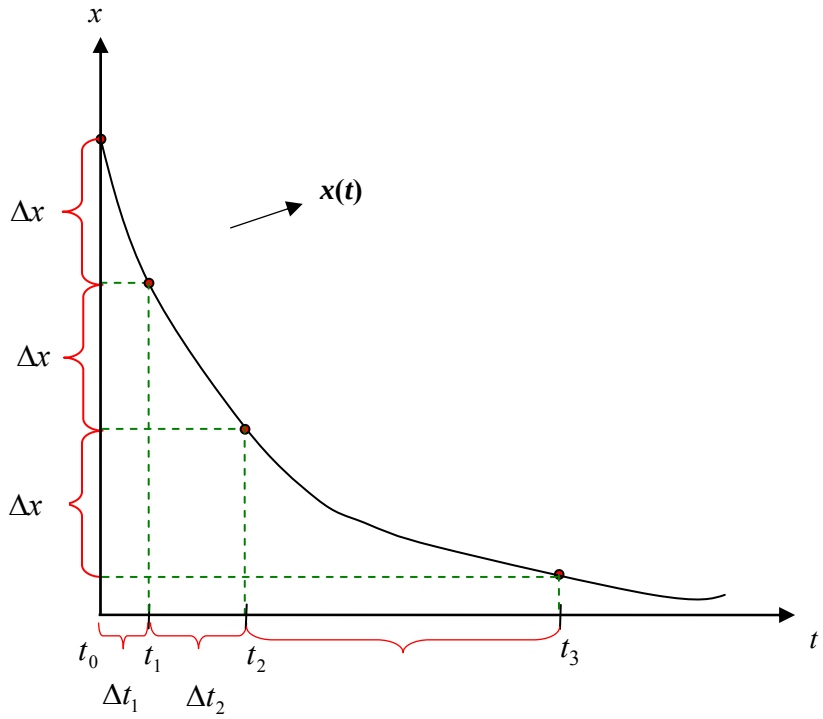
sisteminin yaklaşık çözümü için lineer otonom sistemlerin çözümüne dayanan bir iterasyon metodu elde edilmiştir. Elde edilen bu metotda sabit adım genişlikleri kullanılmaktadır (Lara 2007) .

1.2. Problemin Tanıtımı

Diferensiyel denklem sistemlerinin nümerik integrasyonunda hesaplamaların yaklaşımını etkileyen en önemli kavramlardan birisi adım genişliğidir. Hesaplamalarda; sabit adım genişliği kullanmak pratik değildir (Şekil 1.1). Eğer, adım genişliği büyük seçilirse hesaplanan çözüm, denklemin tam çözümünden uzaklaşabilir. Küçük adım genişlikleri seçilirse de hesaplamaların süresi artmaya başlar. Ayrıca; küçük adım genişlikleri seçilmesi daha fazla aritmetik işlem gerektirdiğinden, hesaplama hataları da artar. Adım genişliği, ele alınan denklem sisteminin tam çözümünün değişiminin hızlı olduğu bölgelerde küçük, değişiminin yavaş olduğu bölgelerde ise daha büyük seçilmelidir (Şekil 1.2).



Şekil 1.1



Şekil 1.2

Bu çalışmada; $A=(a_{ij}) \in \boldsymbol{R}^{N\times N}$ ve $\varphi \in C^m([t_0, T]\times \boldsymbol{R}^N)$ olmak üzere $X'(t)=AX(t)$ ve $X'(t)=AX(t)+\varphi(t,X)$ şeklindeki diferensiyel denklem sistemlerinin $[t_0, T]$ aralığında nümerik çözümleri için adım genişliği stratejileri elde edilmiştir.

1.3. Tezin Yapısı

Tez 4 bölümden oluşmaktadır.

1. bölümde; problemin tanıtımı, literatür özeti ve tezin yapısı ile ilgili bilgi verilmiştir.

2. bölümde; birinci mertebeden Cauchy problemi için Çelik Kızılkan (2004) ın "*Başlangıç Değer Problemlerinin Nümerik İntegrasyonunda Adım Genişliği Tespiti*" isimli tez çalışmasında verilen bilgiler özetlenmiştir.

3. bölümde; diferensiyel denklem sistemleri için Lipschitz şartı, Lipschitz sabiti, lokal hata, hata analizi gibi temel kavramlar ele alınmıştır. $A = (a_{ij}) \in \boldsymbol{R}^{N\times N}$ olmak üzere $X(t_0) = X_0 \in \boldsymbol{R}^N$ başlangıç vektörü ile verilen $X'(t) = AX(t)$ lineer denklem sistemlerinin nümerik integrasyonunda adım genişliğinin seçimi için dört farklı strateji verilmiştir. Verilen stratejilere göre seçilen adım genişlikleri ile problemin nümerik çözümünü hesaplayan dört algoritma verilerek bu adım genişliği stratejileri ve algoritmalar örneklendirilmiştir. Verilen strateji ve algoritmalar, her bir adımda elde edilen adım genişliklerinin büyüklüğü ve oluşan lokal hatalar ile adım sayısı açısından karşılaştırılmıştır.

Daha sonra; $A = (a_{ij}) \in \boldsymbol{R}^{N\times N}$ ve $\varphi \in C^m([t_0, T]\times \boldsymbol{R}^N)$ olmak üzere $X'(t) = AX(t) + \varphi(t, X)$ şeklindeki diferensiyel denklem sistemlerinin hata analizi incelenmiştir. Yapılan hata analizinden yararlanarak bu şekildeki problemlerin nümerik çözümü için üç adım genişliği stratejisi ve bu stratejilere uygun şekilde adım genişlikleri ile nümerik çözüm hesaplayan algoritmalar verilmiştir. Verilen strateji ve algoritmalar örneklendirilmiştir.

2. BİRİNCİ MERTEBEDEN CAUCHY PROBLEMİ İÇİN ADIM GENİŞLİĞİ STRATEJİSİ

Bu bölümde; $D=\{(t,x):|t-t_0|\le T,\ |x-x_0|\le b\}$ bölgesi üzerinde

$$x' = f(t,x),\ x(t_0)=x_0 \tag{2.1}$$

birinci mertebeden Cauchy problemi için Çelik Kızılkan (2004) ın "*Başlangıç Değer Problemlerinin Nümerik İntegrasyonunda Adım Genişliği Tespiti*" isimli tez çalışmasında verilen bilgiler özetlenmiştir.

2.1. Lipschitz Şartı, Lipschitz Sabiti

Lipschitz şartı Cauchy problemlerinin çözümlerinin varlığı ve tekliği ile ilgili bir kavramdır. (t,x_1) , $(t,x_2)\in D=\{(t,x):|t-t_0|\le T,\ |x-x_0|\le b\}$ olmak üzere

$$|f(t,x_1) - f(t,x_2)|\le L\,|x_1 - x_2|$$

olacak şekilde bir $L > 0$ sabiti varsa $f(t,x)$ fonksiyonu $\boldsymbol{R}^2$ deki D bölgesi üzerinde x değişkenine göre *Lipschitz şartını* sağlar denir. Burada L' ye de *Lipschitz sabiti* denir. Eğer $f(t,x)$ fonksiyonu D kümesi üzerinde sürekli ve bu bölgede Lipschitz şartını sağlarsa bu bölgede (2.1) Cauchy probleminin bir tek $x(t)$ çözümü vardır (Gear 1971, Rao 1979, Miranker 1981, Brock ve Malliaris 1989, Shampine ve ark. 1996, Aydın ve ark. 2001, Estep 2002).

2.2. Lokal Hata

(2.1) Cauchy probleminin nümerik metotla hesaplanmasından kaynaklanan lokal hatası; $[t_{i-1},t_i)$ aralığında

$$z' = f(t,z)\,, z(t_{i-1}) = y_{i-1} \tag{2.2}$$

probleminin çözümü $z(t)$ olmak üzere

$$|LE_i| = |y_i - z(t_i)|$$

şeklinde tanımlanır (Shampine ve ark. 1996, Loan 2000, Aydın ve ark. 2001).

2.3. Euler Metodu ve Hata Analizi

(2.1) Cauchy problemi için $t_i \in [t_0 - T, t_0 + T]$ ve $h_i = t_i - t_{i-1}$ olmak üzere Euler metodu,

$$y_{i+1} = y_i + h_{i+1} f_i \ , i = 0, 1, \ldots , n$$

şeklinde tanımlanır (Gear 1971, Hairer ve Wanner 1991, Golub ve Ortega 1992, Shampine ve ark. 1996).

$[t_{i-1}, t_i)$ aralığında (2.2) probleminin çözümü olan $z(t)$ fonksiyonu için ikinci mertebeden Taylor açılımı yazılırsa Euler metodu için;

$$LE_i = -\frac{1}{2} z''(\tau) h_i^{\,2}$$

şeklindedir. Lokal hatanın üst sınırı;

$$|LE_i| \le \frac{h_i^2}{2} \max_{t_{i-1} \le \tau < t_i} |z''(\tau)|, \ \tau \in [t_{i-1}, t_i)$$

şeklinde elde edilir.

2.4. Adım Genişliği Seçimi ve Pratik Adım Genişliği

(2.1) problemi için verilen hata analizi tabanlı adım genişliği stratejisi δ_L hata seviyesinde, Euler metodu kullanılarak

$$h_i < \left(\frac{2\delta_L}{\max\limits_{\tau\in[t_{i-1},t_i)} |z''(\tau)|}\right)^{\frac{1}{2}} \quad , (\, i = 1, 2, \ldots , n) \tag{2.3}$$

şeklinde verilmektedir (Çelik Kızılkan 2004).

h^*, kullanıcının belirlediği istenildiği kadar küçük bir parametre olmak üzere $h_i > h^*$ ise h_i adım genişliği kabul edilir ve işleme devam edilir, $h_i < h^*$ olduğu zaman $h_i = 0$ olarak kabul edilir ve işlem sona erer. Burada h_i pratik adım genişliği ve h^* pratik adım genişliği parametresidir (Çelik Kızılkan 2004).

h^* pratik adım genişliği parametresi kullanıcı tarafından belirlenmezse, kullanılan bilgisayarın formatına uygun pozitif en küçük bilgisayar sayısı olarak seçilir. Ancak bu durumda adım genişlikleri çok küçük bulunabileceğinden çözüm beklenmedik bir şekilde hareket edebilir. Dolayısıyla h^*, kullanıcı tarafından problemin yapısına uygun olarak seçilmelidir.

2.5. Adım Genişliği Kontrolü (K)

k; adım sayısı, $\hat{h}_k$; ele alınan adım genişliği seçiminde hesaplanan adım genişliği ve h^*, pratik adım genişliği parametresi olmak üzere adım genişliği kontrolü aşağıdaki şekilde verilir (Çelik Kızılkan 2004).

K: **1.** $\sum_{i=1}^{k-1} h_i + \hat{h}_k \le T$ ise;

1.1. $\hat{h}_k > h^*$ ise $h_k = \hat{h}_k$ alınır.

1.2. $\hat{h}_k < h^*$ ise h_k=0 alınır ve işlem sona erer.

2. $\sum_{i=1}^{k-1} h_i + \hat{h}_k > T$ ise $\hat{\hat{h}}_k = T - \sum_{i=1}^{k-1} h_i$ alınır.

2.1. $\hat{\hat{h}}_k > h^*$ ise $h_k = \hat{\hat{h}}_k$ alınır.

2.2. $\hat{\hat{h}}_k < h^*$ ise h_k=0 alınır ve işlem sona erer.

2.6. Hata Analizi Tabanlı Adım Genişliği İçin Algoritma

$D=\{(t,x):\ |t-t_0|\le T,\ |x-x_0|\le b\}$ bölgesi üzerinde verilen (2.1) Cauchy problemi için adım genişliği ve yaklaşık çözümünü hesap eden algoritma Euler metodu için; $\max_{t_{i-1}\le\tau<t_i}|\frac{1}{2}z''(\tau)|\le M_{t_i}$ olmak üzere aşağıda verilmiştir (Çelik Kızılkan, 2004).

0. Adım (Giriş Elemanları): T, b, h^* ve δ_L sayıları girilir.

k. Adım: M_{t_k} sayısı hesaplanır.

k.1. $\hat{h}_k<(\frac{2\delta_L}{M_{t_k}})^{\frac{1}{2}}$ olacak şekilde $\hat{h}_k$ sayısı hesaplanır.

k.2. K- kontrolü yapılır

k.3. $t_k=t_{k-1}+h_k$ ve $y_k=y_{k-1}+h_k f(t_{k-1},y_{k-1})$ hesaplanır.

3. DİFERENSİYEL DENKLEM SİSTEMLERİ İÇİN ADIM GENİŞLİĞİ STRATEJİLERİ

Adım genişliği stratejilerine geçmeden önce 2. bölümde (2.1) Cauchy problemi için verilen temel kavramları, sistemler için de tanıtalım.

$$X(t) = (x_j(t)),\ X_0 = (x_{j0});\ x_{j0} = x_j(t_0),$$

$$F(t, X) = (f_j);\ f_j = f_j(t, x_1, x_2, ..., x_N)$$

ve $F \in C^m([t_0, T] \times \boldsymbol{R}^N)$, $X(t)$, X_0, $b = (b_j) \in \boldsymbol{R}^N$ olmak üzere

$$D = \{(t, X) : t \in [t_0, T], | x_j - x_{j0} | \leq b_j\}$$

bölgesi üzerinde

$$X'(t) = F(t, X),\ X(t_0) = X_0 \tag{3.1}$$

Cauchy problemini ele alalım. (m , çalışılan nümerik metoda göre değişebilir.)

Herhangi (t, X_1) , $(t, X_2) \in D$ için,

$$\| F(t, X_1) - F(t, X_2) \| \leq L \| X_1 - X_2 \|$$

olacak şekilde bir L sayısı varsa, $F(t, X(t))$ vektör fonksiyonu *Lipschitz şartını* sağlar denir, L sayısına da *Lipschitz sabiti* denir. Eğer $F(t, X)$ fonksiyonu D bölgesi üzerinde sürekli ve bu bölgede Lipschitz şartını sağlarsa bu bölgede (3.1) Cauchy probleminin bir tek $X(t)$ çözümü vardır (Gear 1971, Rao 1979, Shampine ve ark. 1996).

$Z(t) = \begin{pmatrix} z_1(t) & z_2(t) & \cdots & z_N(t) \end{pmatrix}^T$ vektör fonksiyonu (3.1) Cauchy probleminin $[t_{i-1}, t_i)$ aralığındaki çözümü ve $Y_i = \begin{pmatrix} y_{i1} & y_{i2} & \cdots & y_{iN} \end{pmatrix}^T$ *i.* adımda nümerik metotla elde edilen çözüm vektörü olmak üzere

$$Z'(t) = F(t, Z),\ Z(t_{i-1}) = Y_{i-1}$$

Cauchy problemini oluşturalım. $[t_{i-1}, t_i)$ aralığındaki *lokal hata vektörü*,

$$LE_i = Y_i - Z(t_i) = \begin{pmatrix} y_{i1} - z_1(t_i) \\ y_{i2} - z_2(t_i) \\ \vdots \\ y_{iN} - z_N(t_i) \end{pmatrix} = \begin{pmatrix} LE_{i1} \\ LE_{i2} \\ \vdots \\ LE_{iN} \end{pmatrix} \tag{3.2}$$

şeklinde tanımlıdır. $\| LE_i \|$ değerine de $[t_{i-1}, t_i)$ aralığındaki *lokal hata* denir.

Bütün nümerik metotlar için hata analizi yapılabilmekle beraber bu çalışmada basitlik için Euler metodunu kullanacağız. (3.1) sistemi için *Euler metodu*;

$$Y_i = \begin{pmatrix} y_{i1} & y_{i2} & \cdots & y_{iN} \end{pmatrix}^T, \; F_i = \begin{pmatrix} f_1 & f_2 & \cdots & f_N \end{pmatrix}^T,$$

$$f_j = f_j(t, x_1, x_2, ..., x_N) \; (j = 1, 2, \ldots, N)$$

ve $h_{i+1} = t_{i+1} - t_i$ olmak üzere

$$Y_{i+1} = Y_i + h_{i+1} F_i$$

şeklinde tanımlanır (Loan 2000).

Diferensiyel denklem sistemlerinin nümerik integrasyonunda adım genişliği eğer çok büyük seçilirse nümerik çözüm, tam çözümden uzakta hareket edebilir. Bilimsel hesaplamalar bilgisayarlarda yapıldığından adım genişliği, kayan nokta aritmetiğinin etkileri gözlenebilecek şekilde küçük seçilmesi durumunda da yine nümerik çözüm beklenildiği gibi hareket etmeyebilir. Dolayısı ile adım genişliğinin seçiminde kayan noktalı sayılarla çalışılacağı dikkate alınmalıdır. Kayan noktalı sayılar ve kayan nokta aritmetiğinin hesaplamalara etkisini aşağıda kısaca izah edelim.

3.1. Kayan Noktalı Sayılar ve Hesaplamalarda Kayan Nokta Aritmetiğinin Etkisi

Günümüzde bilimsel hesaplamalar doğal olarak bilgisayar kullanılarak yapılmaktadır. Bilgisayarlar, rasyonel sayıların sonlu bir alt kümesi olan

$$\mathbf{F} = \mathbf{F}(\gamma, p_- , p_+ , \mathrm{k}) = \{0\} \cup \{ \mathbf{z} \mid \mathbf{z} = \pm \gamma^p m_\gamma(\mathbf{z}) \}$$

kümesi üzerinde hesap yapmaktadır. Burada $p_- \in \mathbf{Z}^-$, $p_+ \in \mathbf{Z}^+$ olmak üzere $p_- \le p \le p_+$, $p \in \mathbf{Z}$ ve

$$m_\gamma(\mathbf{z}) = \frac{m_1}{\gamma} + \frac{m_2}{\gamma^2} + ... + \frac{m_k}{\gamma^k}; \quad m_j \in \mathbf{Z},\ 0 \le m_j \le \gamma - 1,\ j = 1, 2, \ldots, k\ (m_1 \neq 0)$$

şeklinde tanımlıdır. **F** kümesine *bilgisayar sayıları kümesi* veya *Format kümesi* denilmektedir. **F** kümesinin elemanlarına *kayan noktalı sayılar* ya da *bilgisayar sayıları* denir (Shampine 1996, Akın ve Bulgak 1998, Aydın ve ark. 2003, Çıbıkdiken 2008).

ε_∞ sayısı formatın en büyük elemanı olmak üzere bir $\mathbf{z} \in [-\varepsilon_\infty, \varepsilon_\infty]$ reel sayısı **F** kümesine, ***z*** nin en yakın olduğu kayan noktalı sayıya dönüşerek yerleşmektedir. ***z*** sayısı formatın elemanı değilse, bu sayıya formatta karşılık gelen sayı $[z]_F$ ile gösterilir.

Bilgisayarda yapılan hesaplamalarda, kayan nokta aritmetiğinin özelliklerine bağlı olarak işlemler gerçekleştirilmektedir. **F** kümesinden de anlaşılacağı gibi reel sayıların, bu sonlu kümelerle ifade edilmesi sırasında doğal olarak hatalar karşımıza çıkmaktadır.

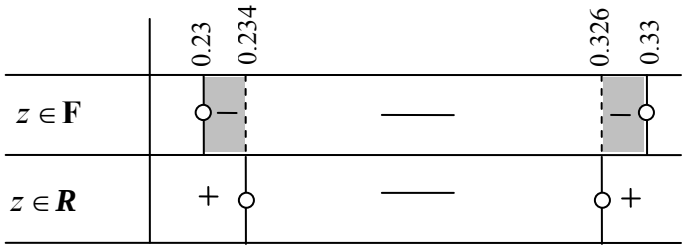
Mesela; $f(x) = x^2 - 0.56x + 0.076284 < 0$ eşitsizliğinin çözüm kümesinin $(0.234, 0.326)$ aralığı olduğu açıktır. **F**(10, -2 , 2 , 2) format kümesinde ise bu aralık $(0.23, 0.33)$ şeklinde bulunmaktadır.

$$[f(0.23)]_F = 0 \quad \text{ve} \quad z \in \mathbf{F},\ z \in (0.23, 0.33) \ \text{ise} \ [f(z)]_F < 0$$

olurken

$$z \in \boldsymbol{R},\ z \in (0.23, 0.234) \cup (0.326, 0.33) \ \text{ise} \ f(z) > 0$$

olmaktadır. Örneğin; x = 0.329 için, $f(0.329)$ = 0.000285 >0 olduğu aşağıdaki şekilde görülmektedir.



Ayrıca kararlılık teorisinde de kayan nokta aritmetiğinin karar vermede etkili olduğu bilinmektedir. Yani kayan nokta aritmetiğine göre yapılan hesaplamalar sonucu verilen sistem, gerçekte kararlı iken kararsız olarak değerlendirilebilmektedir. Bu konuda detaylı bilgi Çıbıkdiken (2008) tarafından verilmiştir.

3.2. $F(t, X) = AX$ Durumunda Adım Genişliği Stratejileri ve Algoritmaları

Bu kısımda; (2.1) Cauchy problemi

$$x' = f(t,x),\ x(t_0) = x_0$$

için (2.3) eşitsizliği ile verilen

$$h_i < \left(\frac{2\delta_L}{\max\limits_{\tau\in[t_{i-1},t_i)} |z''(\tau)|}\right)^{\frac{1}{2}},\ (i = 1, 2, \ldots, n)$$

adım genişliği stratejisi, $A = (a_{ij}) \in \boldsymbol{R}^{N\times N}$, $X = (x_i) \in \boldsymbol{R}^N$ olmak üzere

$$X'(t) = AX(t),\ X(t_0) = X_0 \tag{3.3}$$

Cauchy problemi için genelleştirilecektir.

Şimdi, (3.3) Cauchy probleminin lokal hatasının üst sınırı ile ilgili olan aşağıdaki teoremi verelim.

Teorem 3.1. (3.3) Cauchy probleminin lokal hatası için

$$\| LE_i \| \leq \frac{1}{2} h_i^2 \| A \|^2 \| Z(\tau_{i,j}) \|, \quad \tau_{i,j} \in [t_{i-1}, t_i) \tag{3.4}$$

eşitsizliği doğrudur. Burada $\tau_{i,j}$ değerinde; i, $\tau_{i,j}$ sayısının bulunduğu aralığı ve j ise $Z(t)$ vektörünün j. bileşenini göstermektedir.

İspat. Çelik Kızılkan (2004)' te yapılan *lokal hata analizi*ni (3.3) Cauchy problemine uygulayalım. (3.2) nin LE_{i1} bileşeni için

$$\begin{aligned} LE_{i1} &= y_{i1} - z_1(t_i) \\ &= y_{(i-1)1} + h_i f_{(i-1)1} - (z_1(t_{i-1}) + z_1'(t_{i-1})(t_i - t_{i-1}) + \frac{1}{2!} z_1''(\tau_{i1})(t_i - t_{i-1})^2) \\ &= y_{(i-1)1} + h_i f_{(i-1)1} - (y_{(i-1)1} + h_i f_{(i-1)1} + \frac{1}{2!} z_1''(\tau_{i1}) h_i^2 \end{aligned}$$

$$LE_{i1} = -\frac{1}{2} z_1''(\tau_{i1}) h_i^2 \ , \ \tau_{i1} \in [t_{i-1}, t_i)$$

bulunur. Benzer şekilde diğer bileşenler için de işlem yapılırsa;

$$LE_{ij} = -\frac{1}{2} h_i^2 z_i''(\tau_{ij}) \text{, } i=1,2,\ldots,n \ , \ j=1,2,\ldots,N, \ \tau_{ij} \in [t_{i-1}, t_i)$$

olur. $[t_{i-1}, t_i)$ aralığındaki lokal hata vektörü

$$LE_i = \begin{pmatrix} LE_{i1} \\ LE_{i2} \\ \vdots \\ LE_{iN} \end{pmatrix} = -\frac{1}{2} h_i^2 \begin{pmatrix} z_1''(\tau_{i,1}) \\ z_2''(\tau_{i,2}) \\ \vdots \\ z_N''(\tau_{i,N}) \end{pmatrix} = -\frac{1}{2} h_i^2 Z''(\tau_{i,j}) \text{, } \tau_{i,j} \in [t_{i-1}, t_i) \tag{3.5}$$

şeklinde elde edilir.

(3.5) den norm alınırsa,

$$\| LE_i \| = \frac{1}{2} h_i^2 \| A^2 Z(\tau_{i,j}) \| \leq \frac{1}{2} h_i^2 \| A^2 \| \| Z(\tau_{i,j}) \| \leq \frac{1}{2} h_i^2 \| A \|^2 \| Z(\tau_{i,j}) \|, \quad \tau_{i,j} \in [t_{i-1}, t_i)$$

olup

$$\| LE_i \| \leq \frac{1}{2} h_i^2 \| A \|^2 \| Z(\tau_{i,j}) \|, \quad \tau_{i,j} \in [t_{i-1}, t_i)$$

elde edilir. □

Şimdi (2.3) adım genişliği stratejisinin (3.3) Cauchy problemi için genelleştirilmesi olan stratejiyi ve bu stratejiye uygun adım genişlikleri ile nümerik çözüm hesaplayan algoritmayı verelim.

3.2.1. Strateji ve Algoritma

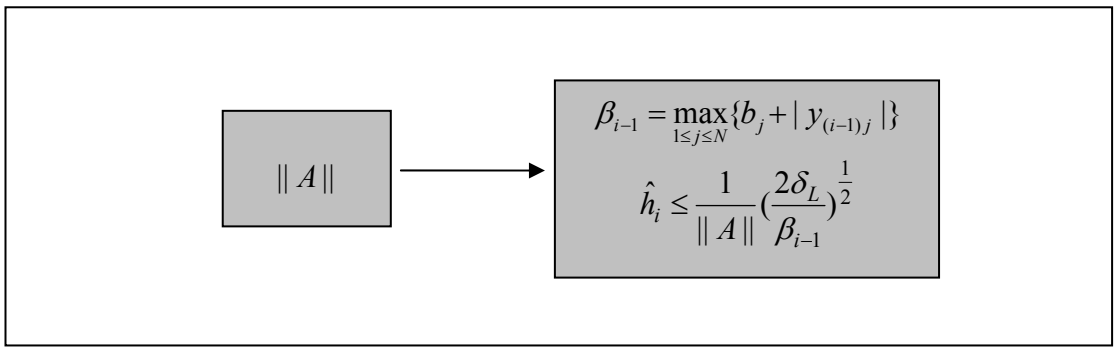
(3.4) eşitsizliğinden, (3.3) lineer Cauchy probleminin nümerik integrasyonu için *i*. adımda

$$\max_{1\le j\le N}\Big(\sup_{t_{i-1}\le\tau_{i,j}<t_i}|z_j(\tau_{i,j})|\Big)\le\beta_{i-1}$$

olmak üzere adım genişliği

$$h_i \le \frac{1}{\|A\|}\left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}} \qquad (3.6)$$

şeklinde hesaplanır. Teoride; (3.6) eşitsizliğinden hesaplanan adım genişliği ile nümerik çözüm yapıldığında her bir adımda oluşan lokal hata, δ_L- hata seviyesinden küçük olur. Strateji 3.2.1 i anlatan şema aşağıda verilmiştir.



Şekil 3.1. *i*. adımda adım genişliğinin Strateji 3.2.1 e göre hesaplanması

Burada;

$$\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})| \le b_j + |z_j(t_{i-1})|, \quad \tau_{i,j} \in [t_{i-1}, t_i)$$

olduğundan β_i sayısı pratikte,

$$\beta_i = \max_{1\le j\le N}\{b_j + |z_j(t_i)|\} = \max_{1\le j\le N}\{b_j + |y_{ij}|\}$$

şeklinde hesaplanacaktır.

(3.3) Cauchy probleminde N=1 olması durumunda aşağıdaki sonuç elde edilir.

Sonuç 3.1. N=1 için (3.6) eşitsizliği, birinci mertebeden lineer ve homojen olmayan Cauchy problemlerinin nümerik integrasyonunda adım genişliği seçimi için verilen (2.3) eşitsizliğine dönüşür.

İspat. N=1 için (3.3) Cauchy problemi $F(t\,,X) = AX = ax = f(t\,,x)$ şeklinde olmaktadır. $z' = az$ ve $\max_{t_{i-1}\le\tau_i<t_i} |z''(\tau_i)| \le M_{t_i}$ olduğundan

$$\max_{t_{i-1}\le\tau_i<t_i} |z''(\tau_i)| \le \sup_{t_{i-1}\le\tau_i<t_i} |z''(\tau_i)| = \sup_{t_{i-1}\le\tau_i<t_i} |a^2 z(\tau_i)|$$

$$= a^2 \sup_{t_{i-1}\le\tau_i<t_i} |z(\tau_i)| \le a^2\beta_{i-1} = M_{t_i}$$

elde edilir. Buradaki M_{t_i} sayısı (2.3) deki ile aynıdır. Ayrıca $\|A\| = |a|$ ve $\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})| = \sup_{t_{i-1}\le\tau_i<t_i} |z(\tau_i)| = \max_{j=1}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})|) \le \beta_{i-1}$ olduğu açıktır. Bu sonuç (3.6) eşitsizliğinde yazılırsa,

$$h_i \le \frac{1}{\|A\|}\left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}} = \frac{1}{a}\left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}} = \left(\frac{2\delta_L}{a^2\beta_{i-1}}\right)^{\frac{1}{2}} = \left(\frac{2\delta_L}{M_{t_i}}\right)^{\frac{1}{2}}$$

bulunur. Bu ise N=1 için (3.6) eşitsizliğinin (2.3) eşitsizliğine denk olduğunu gösterir. □

(3.3) Cauchy probleminin nümerik integrasyonu için (3.6) daki şekilde adım genişliği hesaplayan ve bu adım genişlikleri ile nümerik çözüm hesaplayan algoritma aşağıdaki şekildedir.

0. Adım (Giriş Elemanları): t_o, T, b, h^*, δ_L, X_o, A elemanları girilir.

1. Adım: $\| A \|$ sayısı hesaplanır.

2. Adım: $\beta_{i-1} = \max_{1 \le j \le N} \{ b_j + | y_{(i-1)j} | \}$ ve $\hat{h}_i \le \frac{1}{\| A \|} (\frac{2\delta_L}{\beta_{i-1}})^{\frac{1}{2}}$ olacak şekilde $\hat{h}_i$ sayıları hesaplanır.

3. Adım: K- kontrolü yapılır.

4. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i A) Y_{i-1}$ hesaplanır ve 2. adıma gidilir.

Algoritma 3.2.1. (3.6) daki şekilde adım genişlikleri ile nümerik çözüm hesaplayan algoritma

Not 3.1. Algoritma 3.2.1 de hesaplama işlemini durduran adım K- kontrolünün yapıldığı adımdır. Bu bölümde verilen diğer algoritmalarda da hesaplama işlemini K-kontrol adımı durdurmaktadır. Eğer; K- kontrol algoritması, adım 1.2. veya 2.2. ile algoritmaları sonlandırıyorsa ele alınan Cauchy problemi kötü şartlıdır. Bu durumda, algoritmalar bölgeyi tamamlamadan sona erer ve cevap olarak önceki adımlarda elde edilen sonuçlar verilir.

Örnek 3.1. $D = \{ (t, X) : t \in [0,5], | x_j - x_{j0} | \le 5 \}$ bölgesi üzerinde

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 0{,}5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{3.7}$$

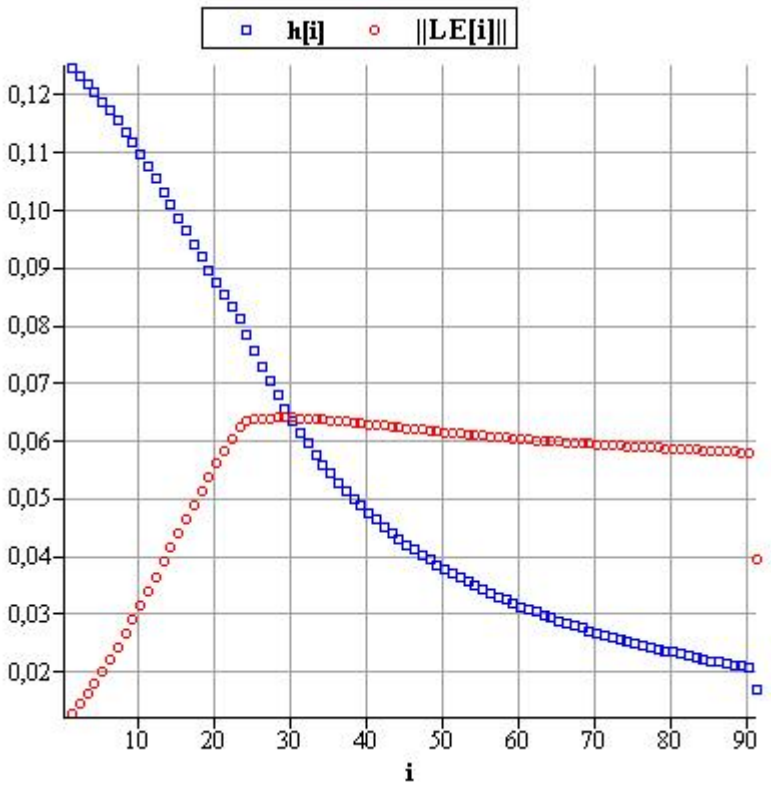
Cauchy probleminin çözümü $h^* = 10^{-12}$ ve $\delta_L = 10^{-1}$ için Algoritma 3.2.1 ile hesaplandığında elde edilen adım genişlikleri ve oluşan lokal hatalar, bazı adımlar için Tablo 3.1. de verilmiştir.

Tablo 3.1. de, Strateji 3.2.1 den her bir adımda lokal hata δ_L hata seviyesinden küçük kalacak şekilde adım genişlikleri elde edildiği görülmektedir.

Tablo 3.1. Algoritma 3.2.1 ile elde edilen h_i ve $\| LE_i \|$ değerleri

i	h_i	$\| LE_i \|$
1	0.1250161491	0.0131475365170381359
2	0.123733735	0.0147710530313330414
3	0.1223512160	0.0165216090215389994
4	0.1208680538	0.0183966633296897484
⋮	⋮	⋮
89	0.02134452474	0.0584853406008727690
90	0.02110750595	0.0584275874221723835
91	0.017261420	0.0398595718171673164

Algoritma 3.2.1 ile her bir adımda elde edilen adım genişlikleri ve oluşan lokal hatalar Grafik 3.1. deki gibidir. Grafik 3.1. de h_i adım genişlikleri h[i] ile ve $\| LE_i \|$ lokal hata değerleri ||LE[i]|| ile gösterilmiştir.



Grafik 3.1. Algoritma 3.2.1 ile her bir adımda elde edilen h_i ve $\| LE_i \|$ değerleri

Hesaplama işleminin son adımında $h_{91} = \hat{\hat{h}}_{91} = T - t_{90}$ şeklinde hesaplandığı Grafik 3.1. de görülmektedir.

Örnek 3.2. $D = \{(t, X) : t \in [0,0.1], | x_j - x_{j0} | \leq 5\}$ bölgesi üzerinde

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 25 & 0 \\ -2 & 100 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \tag{3.8}$$

Cauchy problemi verilsin. Problemin çözümü, $\delta_L = 10^{-1}$ hata seviyesinde $h^* = 10^{-12}$ için Algoritma 3.2.1 ile hesaplandığında elde edilen adım genişlikleri ve oluşan lokal hatalar Tablo 3.2 de verilmiştir.

Tablo 3.2. Algoritma 3.2.1 ile elde edilen h_i ve $\| LE_i \|$ değerleri

i	h_i	$\| LE_i \|$
1	0.1689948036e-2	0.0144747589314876864
2	0.1679839727e-2	0.0166805391180804634
3	0.1669556348e-2	0.0192073945533529321
⋮	⋮	⋮
189	0.1104889178e-3	0.0999993561275817762
190	0.1098854727e-3	**0.100000847583688338**
191	0.109288554e-3	**0.100002304549898030**
⋮	⋮	⋮
563	0.3609754421e-4	**0.100041229473378804**
564	0.332976e-5	0.853375008237391205e-3

Tablo 3.2. den görüldüğü gibi problemin lokal hatası istenilen hata seviyesine çok yakın oluşmaktadır. Hesaplamalarda ortaya çıkan yuvarlama hatalarının birikmesiyle 190. adımdan itibaren lokal hatalar istenilen hata seviyesinden daha büyük gerçekleşmiştir. Son adımda ise adım genişliği $h_{564} = T - t_{563}$ olduğundan doğal olarak $\| LE_{564} \| < \delta_L$ dir.

Uyarı 3.1. Her bir adımda lokal hata δ_L sayısından küçük kalacak şekilde adım genişlikleri (3.6) eşitsizliğinden hesaplandığında, problemin lokal hatası δ_L sayısına çok yakın olabilir. Ancak pratikte; bilgisayarda kayan noktalı sayıların kullanımından kaynaklanan hatalar oluştuğundan, Örnek 3.2 de görüldüğü gibi, hesaplamalarda bazı adımlarda $\| LE_i \| > \delta_L$ gerçekleşebilir.

Bu anlamda; kayan nokta aritmetiğinin sebep olduğu etkileri azaltmak için lokal hatanın (3.4) üst sınırı yerine, (3.4) ten daha büyük olan bir üst sınır yardımıyla adım genişliği hesaplayan aşağıdaki stratejiyi verelim.

3.2.2. Strateji ve Algoritma

Adım genişliği stratejisini vermeden önce, lokal hatanın üst sınırı ile ilgili olarak aşağıdaki teoremi verelim.

Teorem 3.2. (3.3) Cauchy probleminin lokal hatası için $\alpha = \max_{1 \le i,j \le N} | a_{ij} |$ ve $\max_{1 \le j \le N} (\sup_{t_{i-1} \le \tau_{i,j} < t_i} | z_j(\tau_{i,j}) |) \le \beta_{i-1}$ olmak üzere

$$\| LE_i \| \le (\frac{1}{2} \alpha^2 \beta_{i-1}) \sqrt{N^5} h_i^2 \tag{3.9}$$

eşitsizliği doğrudur.

İspat. (3.3) probleminin lokal hatası için (3.4) üst sınırı verilmişti. Her $A = (a_{ij}) \in \boldsymbol{R}^{N \times N}$ ve $Z = (z_j) \in \boldsymbol{R}^{N}$ için

$$\| A \| \le N \max_{1 \le i,j \le N} | a_{ij} | \ , \ \| Z \| \le \sqrt{N} \max_{1 \le j \le N} | z_j |$$

eşitsizlikleri geçerli olduğundan

$$\alpha = \max_{1\le i,j\le N} |\, a_{ij} \,| \quad , \quad \max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_{i,j} < t_i} |\, z_j(\tau_{i,j}) \,|) \le \beta_{i-1}$$

alınırsa;

$$\| LE_i \| \le (\frac{1}{2}\alpha^2 \beta_{i-1})\sqrt{N^5} h_i^2$$

elde edilir. Burada, $\max_{1\le j\le N} |\, z_j \,| \le \beta_{i-1}$ olduğu açıktır. □

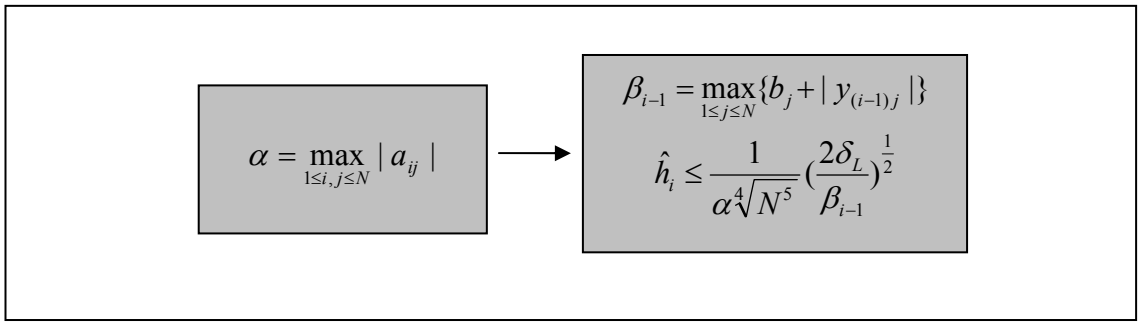
Lokal hatanın (3.9) eşitsizliği göz önüne alındığında; (3.3) lineer Cauchy probleminin nümerik integrasyonu için *i*. adımda;

- δ_L kullanıcının belirlediği hata seviyesi,
- $\alpha = \max_{1\le i,j\le N} |\, a_{ij} \,| \quad , \quad \max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_i < t_i} |\, z_j(\tau_i) \,|) \le \beta_{i-1}$

olmak üzere $\| LE_i \| < \delta_L$ olacak şekilde adım genişliği,

$$h_i \le \frac{1}{\alpha\sqrt[4]{N^5}} (\frac{2\delta_L}{\beta_{i-1}})^{\frac{1}{2}} \qquad (3.10)$$

olacak şekilde hesaplanır. Şekil 3.2. de Strateji 3.2.2 anlatılmıştır.



Şekil 3.2. *i*. adımda adım genişliğinin Strateji 3.2.2 ye göre hesaplanması

Uyarı 3.2. Lokal hatanın (3.10) ile verilen üst sınırı, (3.6) ile verilen üst sınırından daha büyük olduğundan, (3.10) eşitsizliğinden elde edilen adım genişliklerinin (3.6) eşitsizliğinden elde edilen adım genişliklerinden daha küçük olacağı açıktır. Doğal olarak daha küçük adım genişlikleri ile yapılan hesaplamalarda oluşan lokal hata daha küçük olur.

Strateji 3.2.2 ye uygun adım genişliklerini kullanarak (3.3) Cauchy probleminin nümerik çözümünü hesaplayan algoritma aşağıda verilmiştir.

0. Adım (Giriş Elemanları): t_0, T, b, h^*, δ_L, N, X_0, A elemanları girilir.

1. Adım: $\alpha = \max_{1\le i,j\le N} |a_{ij}|$ sayısı hesaplanır.

2. Adım: $\beta_{i-1} = \max_{1\le j\le N}\{b_j + |y_{(i-1)j}|\}$ ve $\hat{h}_i \le \frac{1}{\alpha\sqrt[4]{N^5}}(\frac{2\delta_L}{\beta_{i-1}})^{\frac{1}{2}}$ olacak şekilde $\hat{h}_i$ sayıları hesaplanır.

3. Adım: K- kontrolü yapılır.

4. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i A)Y_{i-1}$ hesaplanır ve 2. adıma gidilir.

Algoritma 3.2.2. (3.10) daki şekilde adım genişlikleri ile nümerik çözüm hesaplayan algoritma

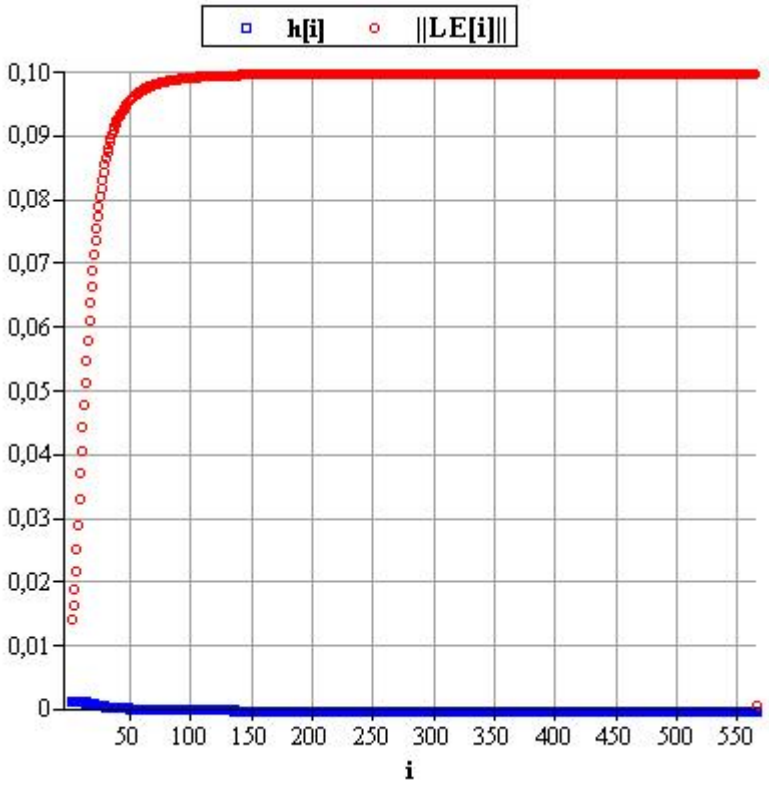
Örnek 3.3. $D = \{(t, X) : t \in [0,0.1], |x_j - x_{j0}| \le 5\}$ bölgesi üzerinde (3.8) ile verilen

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 25 & 0 \\ -2 & 100 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Cauchy probleminin nümerik çözümü için Algoritma 3.2.2 kullanıldığında elde edilen adım genişlikleri ve oluşan lokal hata değerleri Tablo 3.3. ve Grafik 3.2. de görülmektedir. Pratik adım genişliği parametresi $h^* = 10^{-12}$ ve hata seviyesi $\delta_L = 10^{-1}$ alınmıştır.

Tablo 3.3. Algoritma 3.2.2 ile elde edilen h_i ve $\| LE_i \|$ değerleri

i	h_i	$\| LE_i \|$
1	0.7106871836e-3	0.247725475422605522e-2
2	0.7088901783e-3	0.263719975798181513e-2
3	0.7070797425e-3	0.280709854568808245e-2
4	0.7052560422e-3	0.298751831692567124e-2
⋮	⋮	⋮
1455	0.1389815716e-4	0.0176804473977404536
1456	0.103563e-4	0.982982567864236317e-2



Grafik 3.2. Algoritma 3.2.2 ile her bir adımda elde edilen h_i ve $\| LE_i \|$ değerleri

Örnek 3.2. ve Örnek 3.3. deki veriler aynıdır. Tablo 3.2., Tablo 3.3. ve Grafik 3.2. incelendiğinde Algoritma 3.2.1 e göre, Algoritma 3.2.2 deki adım genişliklerinin

sayısının arttığı dolayısıyla her bir adımda oluşan lokal hatanın $\delta_L = 10^{-1}$ sayısından küçük gerçekleştiği görülür.

Şimdi Algoritma 3.2.1 ve Algoritma 3.2.2 nin etkinliklerini aşağıdaki örnek üzerinde inceleyelim.

Örnek 3.4. $D = \{(t, X) : t \in [0,10], | x_j - x_{j0} | \leq 5\}$ bölgesi üzerinde

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 0.5 & -2 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{3.11}$$

probleminin çözümü $h^* = 10^{-12}$ ve hata seviyesi $\delta_L = 10^{-1}$ için Algoritma 3.2.1 ve Algoritma 3.2.2 ile hesaplandığında elde edilen adım genişlikleri Tablo 3.4.a. da ve oluşan lokal hatalar Tablo 3.4.b de özetlenmiştir.

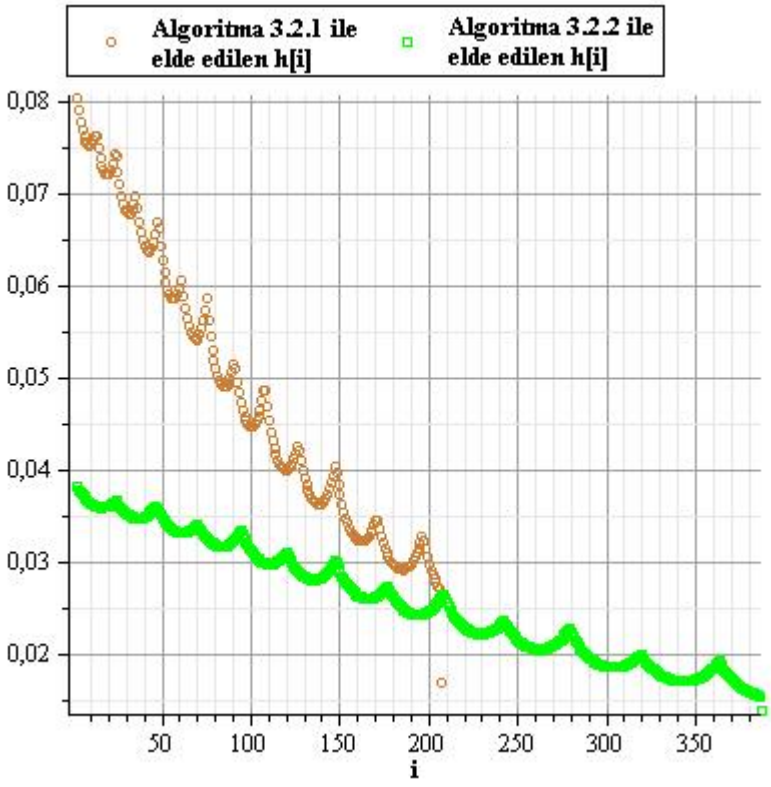
Tablo 3.4.a. Algoritma 3.2.1 ve Algoritma 3.2.2 ile elde edilen adım genişlikleri

Algoritma 3.2.1		**Algoritma 3.2.2**	
i	h_i	***i***	h_i
1	0.8058662244e-1	**1**	0.3838149463e-1
2	0.7926680913e-1	**2**	0.3807822332e-1
3	0.7813720983e-1	**3**	0.3779661276e-1
4	0.7720342995 e-1	**4**	0.3753688838e-1
⋮	⋮	⋮	⋮
205	0.2726122033e-1	**205**	0.2617666791e-1
206	0.17133122e-1	**206**	0.2651910537e-1
		⋮	⋮
		385	0.1571817100e-1
		386	0.14096837e-1

Tablo 3.4.b. Algoritma 3.2.1 ve Algoritma 3.2.2 ile elde edilen $\| LE_i \|$ değerleri

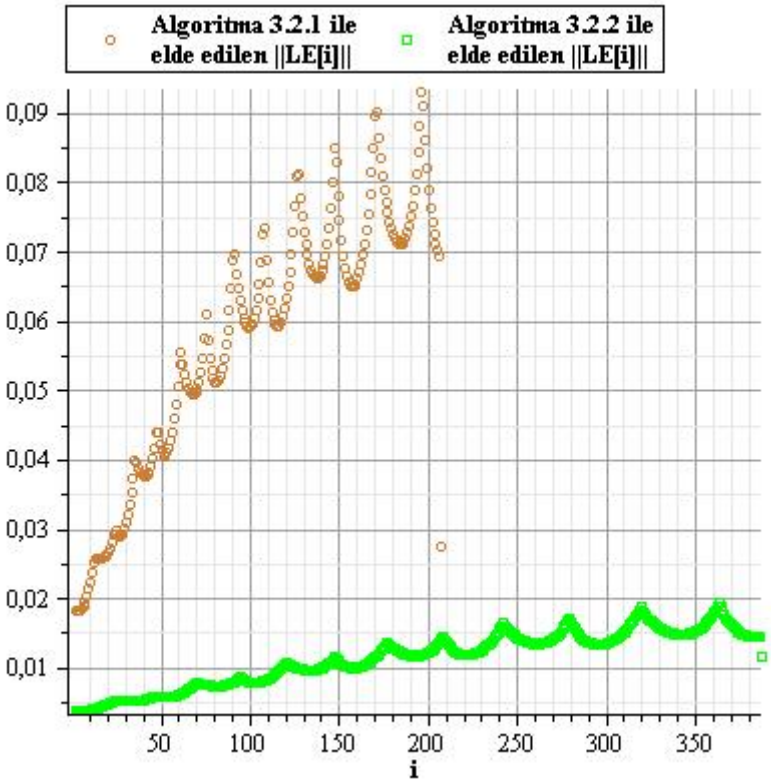
Algoritma 3.2.1		**Algoritma 3.2.2**	
i	$\| LE_i \|$	***i***	$\| LE_i \|$
1	0.185802303853953334**e-1**	**1**	0.420902438666672773**e-2**
2	0.184649418641384296**e-1**	**2**	0.417803168203028860**e-2**
3	0.185398275824510302**e-1**	**3**	0.415731512774776200**e-2**
4	0.188071394118039490**e-1**	**4**	0.414707116645184060**e-2**
⋮	⋮	⋮	⋮
205	0.698194734019560032**e-1**	**205**	0.140837424683594118**e-1**
206	0.279023834806222982**e-1**	**206**	0.145089522051757014**e-1**
		⋮	⋮
		385	0.148043600018341900**e-1**
		386	0.120012728307709385**e-1**

Tablo 3.4.a. ve Tablo 3.4.b. de verilen değerlerin grafikleri aşağıdaki şekildedir.



Grafik 3.3.a. Algoritma 3.2.1 ve Algoritma 3.2.2 den elde edilen adım genişlikleri

Grafik 3.3.a. dan Algoritma 3.2.1 yardımıyla hesaplanan adım genişliklerinin Algoritma 3.2.2 yardımıyla hesaplanan adım genişliklerinden daha büyük olduğu; bunun sonucu olarak da Grafik 3.3.b. den Algoritma 3.2.2 de oluşan lokal hatanın Algoritma 3.2.1 de oluşan lokal hatadan daha küçük olduğu görülmektedir.



Grafik 3.3.b. Algoritma 3.2.1 ve Algoritma 3.2.2 de oluşan lokal hatalar

(3.10) eşitsizliğindeki gibi adım genişliği seçildiğinde oluşan lokal hatanın istenilen hata seviyesinin çok altında kaldığı Örnek 3.4. ten rahatlıkla görülmektedir.

3.2.3. Strateji ve Algoritma

Bu kısımda, lokal hatanın δ_L sayısına daha yakın gerçekleşmesinin istenmesi durumunda Strateji 3.2.2 den elde edilen aşağıdaki modifiye adım genişliği stratejisi ve bu stratejiye uygun olarak nümerik çözüm hesap eden algoritma verilecektir.

i- inci adımda $\hat{h}_i$, (3.10) eşitsizliğinde önerilen adım genişliği ve γ>1 olacak şekilde bir reel sayı olmak üzere

$$h_i^k = \gamma^{k-1}\hat{h}_i \ , \ \ k = 2,3,...,p$$

adım genişliği dizisini oluşturalım.

$$Y_i^k = (I + h_i^k A)Y_{i-1} \ , \ Z_i^k = e^{Ah_i^k} Y_{i-1}$$

olmak üzere $\| LE_i^k \| = \| Y_i^k - Z_i^k \|$ hatalarını hesaplayalım. Burada *p* sayısı,

$$\| LE_i^{p-1} \| < \delta_L \ , \ \| LE_i^p \| > \delta_L$$

şartını sağlayan bir doğal sayıdır ve $\| LE_i^1 \| < \delta_L$ olduğu açıktır. Böylece, *i*. adımda

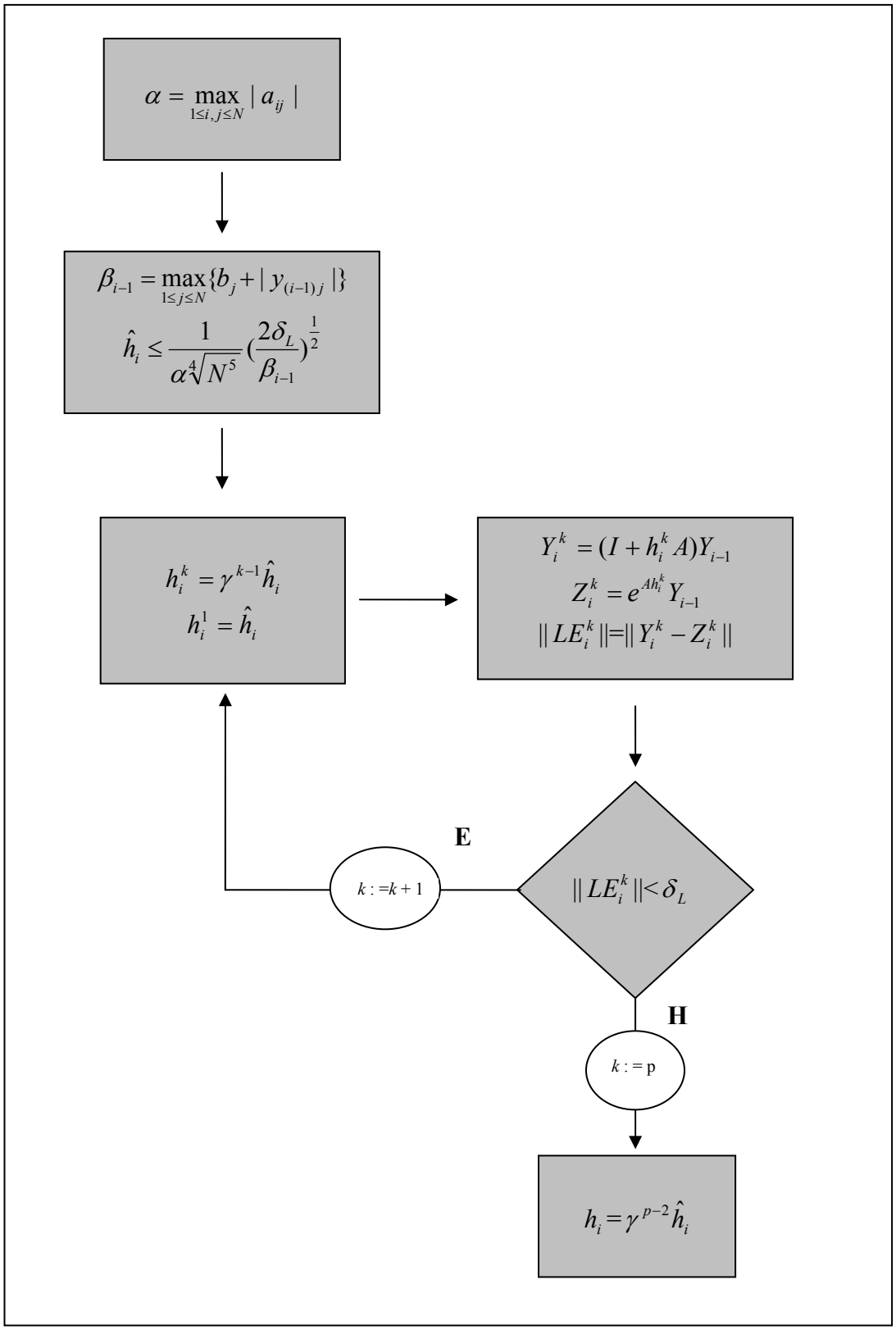
- δ_L kullanıcının belirlediği hata seviyesi,
- $\alpha = \max\limits_{1 \le i,j \le N} | a_{ij} |$, $\max\limits_{1 \le j \le N} (\sup\limits_{t_{i-1} \le \tau_{i,j} < t_i} | z_j(\tau_{i,j}) |) \le \beta_{i-1}$

olmak üzere adım genişliği;

$$h_i = \gamma^{p-2}\hat{h}_i \ ; \ \ \hat{h}_i \le \frac{1}{\alpha \sqrt[4]{N^5}} \left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}} \qquad (3.12)$$

şeklinde hesaplanır. Burada γ sayısı güvenlik çarpanıdır (safety factor). γ>2 olması durumu, hesaplama işlemlerinde çok pratik olmadığından; biz bu çalışmada $\gamma \in \boldsymbol{R}$ sayısını $1 < \gamma < 2$ olacak şekilde kullanacağız.

Strateji 3.2.3 ü anlatan şema aşağıda verilmiştir.



Şekil 3.3. *i*. adımda adım genişliğinin Strateji 3.2.3 e göre hesaplanması

(3.3) Cauchy probleminin nümerik integrasyonu için her bir adımda (3.12) deki şekilde adım genişliği, bu adım genişliklerini kullanarak nümerik çözümleri ve oluşan lokal hataları hesaplayan Algoritma 3.2.3 ü aşağıda verelim.

0. Adım (Giriş Elemanları): t_o, T, b, h^*, δ_L, γ, N, X_o, A elemanları girilir.

1. Adım: $\alpha = \max\limits_{1\le i,j\le N} | a_{ij} |$ sayısı hesaplanır.

2. Adım: $\beta_{i-1} = \max\limits_{1\le j\le N}\{b_j + | y_{(i-1)j} |\}$ ve $\hat{h}_i \le \dfrac{1}{\alpha\sqrt[4]{N^5}}\left(\dfrac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}}$ olacak şekilde $\hat{h}_i$ sayıları hesaplanır.

3. Adım: $k \ge 1$ $(k \in \boldsymbol{N})$ için $h_i^k = \gamma^{k-1}\hat{h}_i$, $Y_i^k = (I + \hat{h}_i^k A)Y_{i-1}$, $Z_i^k = e^{A\hat{h}_i^k}Y_{i-1}$ ve $\| LE_i^k \| = \| Y_i^k - Z_i^k \|$ sayıları hesaplanır.

4. Adım: Eğer $\| LE_i^k \| < \delta_L$ ise, $k:=k+1$ için $h_i^k = \gamma^{k-1}\hat{h}_i$ hesaplanır ve 3. adıma gidilir.

5. Adım: $\hat{h}_i^k = \gamma^{k-2}\hat{h}_i$ hesaplanır.

6. Adım: K- kontrolü yapılır.

7. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i A)Y_{i-1}$ hesaplanır ve 2. adıma gidilir.

Algoritma 3.2.3. (3.12) deki şekilde adım genişlikleri ile nümerik çözüm hesaplayan algoritma

Örnek 3.5. $D = \{(t, X) : t \in [0\,, 10]\,, | x_j - x_{j0} | \le 5\}$ bölgesi üzerinde

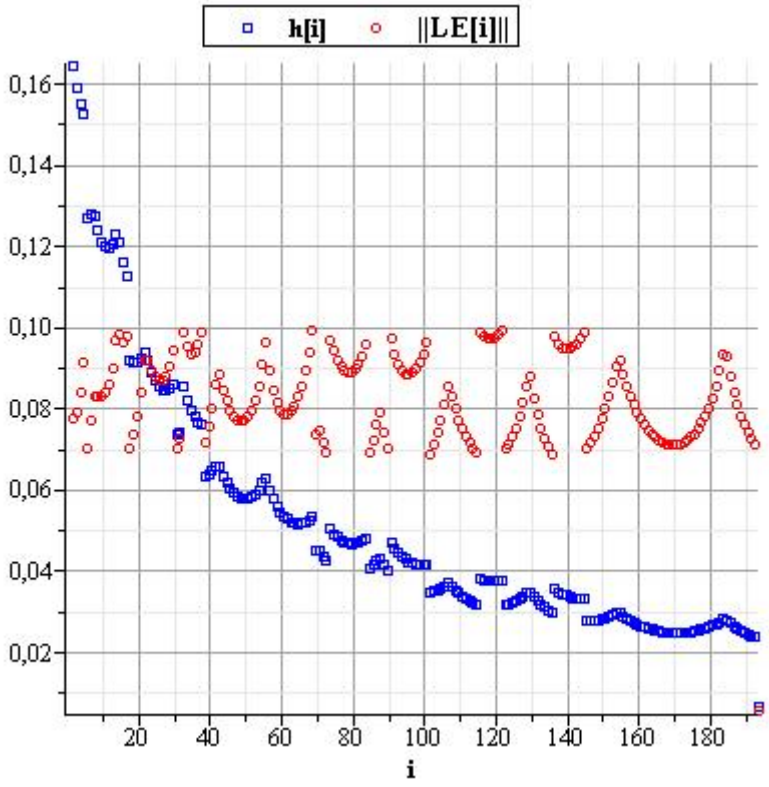
$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 0.5 & -2 \\ 2 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Cauchy probleminin çözümünün $h^* = 10^{-12}$ ve $\delta_L = 10^{-1}$ için Algoritma 3.2.1 ve Algoritma 3.2.2 ile hesaplanması durumunda elde edilen adım genişlikleri ve lokal hatalar Örnek 3.4. de verilmişti. Aynı problem için γ=1.2 alınarak Algoritma 3.2.3

ile elde edilen adım genişlikleri ve oluşan lokal hataların bazı adımları Tablo 3.5. de ve Grafik 3.4. de görülmektedir.

Tablo 3.5. Algoritma 3.2.3 ile elde edilen h_i ve $\| LE_i \|$ değerleri (γ=1.2)

i	h_i	$\| LE_i \|$
1	0.1650334016	0.780906450710349420e-1
2	0.1596360449	0.797996964932324304e-1
3	0.1557193302	0.844532729650658288e-1
4	0.1534190859	0.919635474074751018e-1
⋮	⋮	⋮
192	0.2426248719e-1	0.718666525611627444e-1
193	0.7099897e-2	0.620405331381500193e-2



Grafik 3.4. Algoritma 3.2.3 ile elde edilen h_i ve $\| LE_i \|$ değerleri (γ=1.2)

Aynı problemin çözümü; Algoritma 3.2.1 ile 206 adımda, Algoritma 3.2.2 ile 386 adımda ve Algoritma 3.2.3 ile ise 193 adımda hesaplanmıştır.

Strateji 3.2.3 te, (3.10) eşitsizliği ile elde edilen adım genişlikleri lokal hata ile kontrol edilerek, istenilen hata seviyesinde önceki stratejilerde elde edilen adım genişliklerinden daha büyük adım genişlikleri hesaplanabilmektedir.

3.2.4. Strateji ve Algoritma

Bu kısımda; Strateji 3.2.2 ve Strateji 3.2.3 ü kullanarak Strateji 3.2.4 ü vereceğiz. Strateji 3.2.4 ile Strateji 3.2.3 deki adım genişliklerinden daha büyük adım genişliği hesaplanmaktadır. Ayrıca lokal hata istenilen hata seviyesine daha yakın oluşmaktadır. Nümerik integrasyonun *i*. adımında Strateji 3.2.4 e göre adım genişliği aşağıdaki şekilde seçilir.

- **I. AŞAMA**

$\hat{h}_i$; Strateji 3.2.2 de önerilen (3.10) eşitsizliğinden hesaplanır.

- **II. AŞAMA**

γ >1 olacak şekilde bir reel sayı olsun. Strateji 3.2.3 de yapıldığı gibi,

$$h_i^k = \gamma^{k-1}\hat{h}_i \ , \ \ k = 2,3,...,p$$

adım genişliği dizisini ele alalım. *p* sayısı,

$$\| LE_i^{p-1} \| < \delta_L \text{ ve } \| LE_i^{p} \| > \delta_L$$

şartını sağlayan bir doğal sayı, $Y_i^k = (I + h_i^k A)Y_{i-1}$, $Z_i^k = e^{Ah_i^k} Y_{i-1}$ olmak üzere $\| LE_i^k \| = \| Y_i^k - Z_i^k \|$ hatalarını hesaplayalım.

- **III. AŞAMA**

$h_i^{p-1} = \gamma^{p-2}\hat{h}_i$ adım genişliği ile hesaplama yapıldığında $\| LE_i^{p-1} \| < \delta_L$ ve $h_i^p = \gamma^{p-1}\hat{h}_i$ adım genişliği ile hesaplama yapıldığında $\| LE_i^p \| > \delta_L$ olduğuna göre, (γ^{p-1}, γ^p) aralığındaki bir c reel sayısı için $c\hat{h}_i$ adım genişliği ile hesaplama yapıldığında oluşan lokal hata δ_L seviyesine daha yakın gerçekleşir. Böyle bir c reel sayısını bulabilmek için,

$$e^{p,0} = \gamma^{p-1} \ , \ e^{p,j} = \frac{1}{2}(\gamma^{p-2} + e^{p,j-1}) \ \ , \ j=1,2,\ldots,r$$

katsayı dizisini ve

$$h_i^{p,j} = e^{p,j}\hat{h}_i \, , \ j=1,2,\ldots,r$$

adım genişliklerini oluşturalım. r sayısı,

$$\| LE_i^{p,r-1} \| > \delta_L \text{ ve } \| LE_i^{p,r} \| < \delta_L$$

şartını sağlayan bir doğal sayısı olmak üzere

$$Y_i^{k,j} = (I + h_i^{k,j} A)Y_{i-1} \ , \ Z_i^{k,j} = e^{Ah_i^{k,j}} Y_{i-1} \ , \ \| LE_i^{k,j} \| = \| Y_i^{k,j} - Z_i^{k,j} \|$$

değerlerini hesaplayalım.

- **IV. AŞAMA**

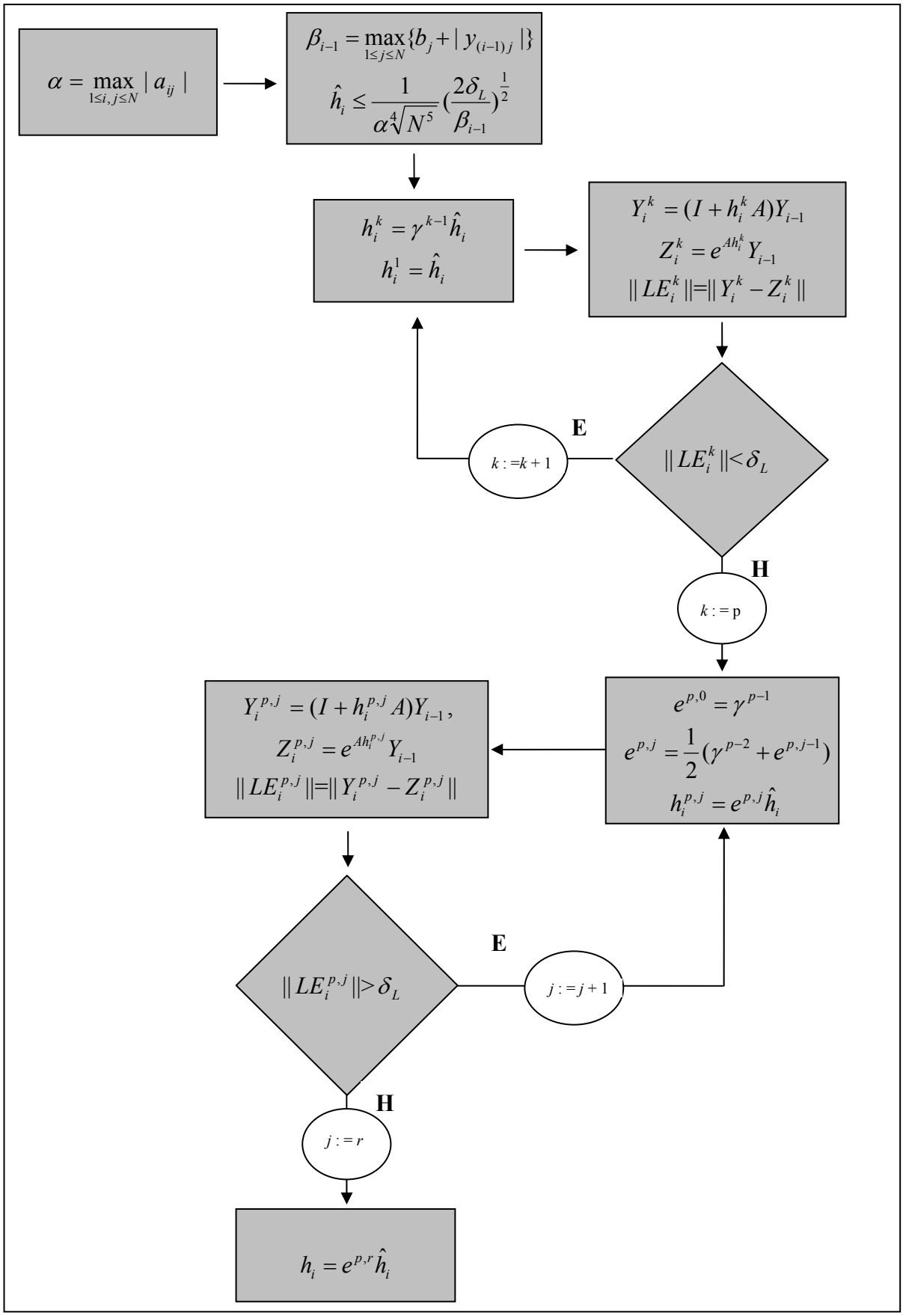
Böylece i. adımda $\| LE_i^{p,r} \| < \delta_L$ olmasını sağlayan adım genişliği;

- δ_L kullanıcının belirlediği hata seviyesi
- $\alpha = \max_{1 \le i,j \le N} | a_{ij} |$, $\max_{1 \le j \le N} (\sup_{t_{i-1} \le \tau_i < t_i} | z_j(\tau_i) |) \le \beta_{i-1}$

olmak üzere;

$$h_i = e^{p,r}\hat{h}_i ; \ \ e^{p,0} = \gamma^{p-1} , \ e^{p,j} = \frac{1}{2}(\gamma^{p-2} + e^{p,j-1}) , \ \hat{h}_i \le \frac{1}{\alpha \sqrt[4]{N^5}} \left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}} \qquad (3.13)$$

ile verilir. Şekil 3.4. Strateji 3.2.4 ü gösteren şemadır.



Şekil 3.4. *i* inci adımda adım genişliğinin Strateji 3.2.4 e göre hesaplanması

Uyarı 3.3. Strateji 3.2.4 ten hesaplanacak adım genişlikleri ile ele alınan Cauchy probleminin nümerik integrasyonunda adım sayısı azalmakla birlikte, adım genişliğinin kontrolü için yapılan işlem sayısı artacağından çözümün hesaplama süresi de artar. Problemin çözümünde hangi durum daha avantajlı ise (problemin yapısına göre) o strateji tercih edilmelidir.

Her bir adımda (3.13) deki şekilde adım genişliği hesap eden ve bu adım genişliklerini kullanarak (3.3) Cauchy problemi için nümerik çözüm ile oluşan lokal hataları hesaplayan Algoritma 3.2.4 ü verelim.

0. Adım (Giriş Elemanları): t_o, T, b, h^*, δ_L , γ , N, X_o, A elemanları girilir.

1. Adım: $\alpha = \max\limits_{1\le i,j\le N} | a_{ij} |$ sayısı hesaplanır.

2. Adım: $\beta_{i-1} = \max\limits_{1\le j\le N}\{b_j + | y_{(i-1)j} |\}$ ve $\hat{h}_i \le \dfrac{1}{\alpha\sqrt[4]{N^5}}(\dfrac{2\delta_L}{\beta_{i-1}})^{\frac{1}{2}}$ olacak şekilde $\hat{h}_i$ sayıları hesaplanır.

3. Adım: $k \ge 1$ $(k \in \boldsymbol{N})$ için $h_i^k = \gamma^{k-1}\hat{h}_i$, $Y_i^k = (I + h_i^k A)Y_{i-1}$, $Z_i^k = e^{Ah_i^k}Y_{i-1}$ ve $\| LE_i^k \| = \| Y_i^k - Z_i^k \|$ sayıları hesaplanır.

4. Adım: Eğer $\| LE_i^k \| < \delta_L$ ise, $k := k+1$ alınır ve 3. adıma gidilir.

5. Adım: $j \ge 1$ $(j \in \boldsymbol{N})$ için $e^{k,0} = \gamma^{k-1}$ olmak üzere $e^{k,j} = \dfrac{1}{2}(\gamma^{k-2} + e^{k,j-1})$ $h_i^{k,j} = e^{k,j}\hat{h}_i$, $Y_i^{k,j} = (I + h_i^{k,j}A)Y_{i-1}$, $Z_i^{k,j} = e^{Ah_i^{k,j}}Y_{i-1}$ ve $\| LE_i^{k,j} \| = \| Y_i^{k,j} - Z_i^{k,j} \|$ sayıları hesaplanır.

6. Adım: Eğer $\| LE_i^{k,j} \| > \delta_L$ ise 5. adıma gidilir.

7. Adım: $\hat{h}_i^{k,j} = e^{k,j}\hat{h}_i$ hesaplanır.

8. Adım: K- kontrolü yapılır.

9. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i A)Y_{i-1}$ hesaplanır ve 2. adıma gidilir.

Algoritma 3.2.4. (3.13) deki şekilde adım genişlikleri ile nümerik çözüm hesaplayan algoritma

Örnek 3.6. Örnek 3.4. ve Örnek 3.5. de verilen

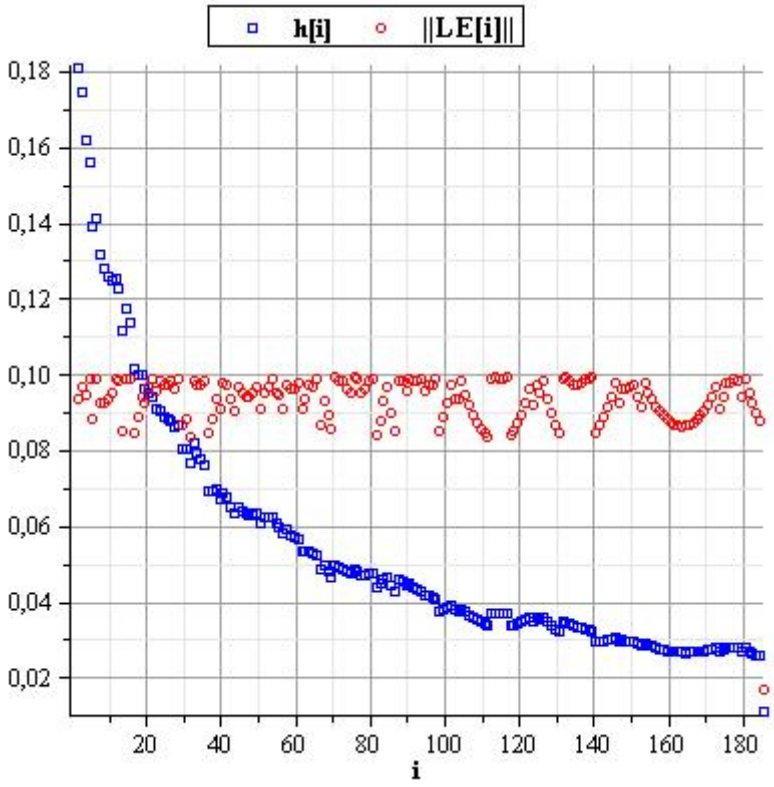
$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 0.5 & -2 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Cauchy problemini aynı verilerle ele alalım. Yani; $D = \{(t, X) : t \in [0,10], |x_j - x_{j0}| \leq 5\}$, $h^* = 10^{-12}$, $\delta_L = 10^{-1}$ ve $\gamma = 1.2$ olsun. Problemin nümerik çözümü için Algoritma 3.2.4 kullanıldığında elde edilen adım genişlikleri ve oluşan lokal hataların bazı adımları Tablo 3.6. da ve Grafik 3.5. de verilmiştir.

Tablo 3.6. Algoritma 3.2.4 ile elde edilen h_i ve $\| LE_i \|$ değerleri ($\gamma = 1.2$)

i	h_i	$\| LE_i \|$
1	0.1815367417	0.945212279352336400e-1
2	0.1750374597	0.974368109258717391e-1
3	0.1627245036	0.951661420000421837e-1
4	0.1566951816	0.996991714287730896e-1
⋮	⋮	⋮
184	0.2633035147e-1	0.885998860658100962e-1
185	0.11706133e-1	0.176706060296323956e-1

Tablo 3.4.a.- 3.4.b.- 3.5.- 3.6. ve Grafik 3.3.a.- 3.3.b- 3.4.- 3.5. incelendiğinde; nümerik integrasyon işleminde lokal hatanın istenilen hata seviyesine yakın olacak şekilde gerçekleşmesi için en büyük adım genişliklerinin Algoritma 3.2.4 ile elde edildiği görülmektedir.



Grafik 3.5. Algoritma 3.2.4 ile elde edilen h_i ve $\| LE_i \|$ değerleri (γ=1.2)

3.2.5. Stratejilerin *m* inci Mertebeden Cauchy Problemlerine Uygulanması

$[t_0, T]$ aralığında $x(t_0)$, $x'(t_0)$, . . . , $x^{(m-1)}(t_0)$ başlangıç şartlarıyla verilen

$$x^{(m)} - a_{m-1}\, x^{(m-1)} - \ldots - a_1\, x' - a_0\, x = 0\,,\ a_i \in \boldsymbol{R}\ (i = 0, 1, \ldots, m-1) \qquad (3.14)$$

Cauchy problemini ele alalım. $x = x_1$, $x' = x_2$, . . . , $x^{(m-1)} = x_m$ alınırsa (3.14) Cauchy problemi,

$$\begin{pmatrix} x_1' \\ x_2' \\ \vdots \\ x_m' \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_0 & a_1 & \cdots & a_{m-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \begin{pmatrix} x_1(t_0) \\ x_2(t_0) \\ \vdots \\ x_m(t_0) \end{pmatrix} = \begin{pmatrix} x_{10} \\ x_{20} \\ \vdots \\ x_{m0} \end{pmatrix}$$

şeklinde yazılabilir. Bu Cauchy problemi, *C* matrisi (3.14) denkleminin Companion matrisi olmak üzere kısaca,

$$X' = CX \ , \ X(t_0) = X_0 \tag{3.15}$$

şeklinde yazılabilir. Bu takdirde (3.3) Cauchy problemi için verilen adım genişliği stratejilerinin ve algoritmaların tümü (3.14) m. mertebeden Cauchy problemi için de rahatlıkla kullanılabilir.

Örnek 3.7. $D = \{(t,x) : t \in [0,5], | x - x_0 | \leq 5\}$ bölgesi üzerinde

$$x'' - x' + 2x = 0 \ , \ x(0) = 1 \ , \ x'(0) = 2 \tag{3.16}$$

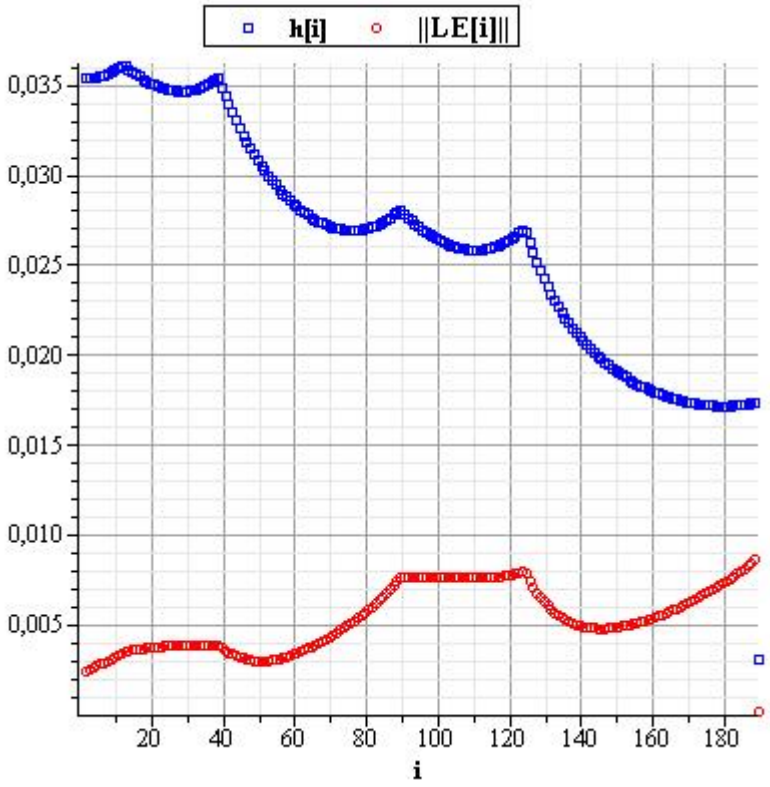
Cauchy problemi,

$$X'(t) = \begin{pmatrix} 0 & 1 \\ -2 & 1 \end{pmatrix} X(t) \ , \ X(0) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

şeklinde yazılabilir. (3.16) probleminin çözümünü mesela, Algortima 3.2.2 ile hesaplayalım. $h^* = 10^{-12}$ ve $\delta_L = 10^{-1}$ olsun. Elde edilen adım genişlikleri ve lokal hata değerleri Tablo 3.7. ve Grafik 3.6. da verilmiştir.

Tablo 3.7. Algoritma 3.2.2 ile elde edilen h_i ve $\| LE_i \|$ değerleri

i	h_i	$\| LE_i \|$
1	0.3553435919e-1	0.255520075611192014e-2
2	0.3553435919e-1	0.264643518554205822e-2
3	0.3554718578e-1	0.273944928167041153e-2
4	0.3557334725e-1	0.283423858472207566e-2
⋮	⋮	⋮
188	0.1745794177e-1	0.873206199579816372e-2
189	0.3211990e-2	0.298731866088589842e-3



Grafik 3.6. Algoritma 3.2.2 ile elde edilen h_i ve $\| LE_i \|$ değerleri

3.2.6. Değerlendirme

3.2. kısımda (3.3) Cauchy probleminin nümerik integrasyonunda adım genişliğinin seçimi için problemin lokal hatası istenilen hata seviyesinden küçük olacak şekilde dört strateji ve bu stratejilere uygun olarak elde edilen adım genişlikleri ile nümerik çözüm hesap eden dört algoritma verilmiştir. Bu algoritmaların özelliklerini aşağıdaki tabloda özetleyelim.

ALGORİTMA 3.2.1	• Birinci mertebeden Cauchy problemleri için Çelik Kızılkan (2004) ın verdiği *hata analizi tabanlı adım genişliği stratejisi*nin (3.3) lineer sistemine genelleştirilmesidir. • Her bir adımda lokal hata istenilen hata seviyesinden küçük kalacak şekilde adım genişlikleri hesaplanmaktadır. • Bazı problemlerde, kayan nokta aritmetiğinden kaynaklanan hesaplama hataları nedeniyle, lokal hata istenilen hata seviyesini geçebilmektedir. Kullanıcının hata seviyesini esnek bıraktığı durumlarda, Algoritma 3.2.1 tercih edilebilir.
ALGORİTMA 3.2.2	• Algoritma 3.2.1 den elde edilmiştir. • Her bir adımda lokal hata, istenilen hata seviyesinden çok küçük gerçekleşmektedir. • Hesaplama işleminde adım sayısı, Algoritma 3.2.1 deki adım sayısından daha fazladır.
ALGORİTMA 3.2.3	• Algoritma 3.2.2 den elde edilmiştir. • Her bir adımda lokal hata, istenilen hata seviyesinden küçük kalacak şekilde adım genişlikleri hesaplanmaktadır. • Her bir adımda lokal hata, istenilen hata seviyesine yakın gerçekleşmektedir. • Hesaplama işleminde adım sayısı, Algoritma 3.2.1 ve 3.2.2 deki adım sayısından daha azdır.
ALGORİTMA 3.2.4	• Algoritma 3.2.2 ve 3.2.3 den elde edilmiştir. • Her bir adımda lokal hata, istenilen hata seviyesinden küçük kalacak şekilde adım genişlikleri hesaplanmaktadır. • Her bir adımda lokal hata, istenilen hata seviyesine daha yakın gerçekleşmektedir. • Hesaplama işleminde adım sayısı, Algoritma 3.2.1, 3.2.2 ve 3.2.3 deki adım sayılarından daha azdır.

Bu kısımda Örnek 3.4. – 3.6. da ele alınan

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 0.5 & -2 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Cauchy problemi için elde edilen sonuçları Tablo 3.7. de gösterelim. Tablo 3.7. de kullanılan semboller $\underline{h} = \min_i \{h_i\}$, $\overline{h} = \max_i \{h_i\}$ ve $\Delta = \max_i \{| \delta_L - \| LE_i \| |\}$ şeklindedir. Elde edilen adım genişlikleri ve oluşan lokal hatalar sırasıyla Grafik 3.6. ve Grafik 3.7. de gösterilmiştir. CPU, Maple 12 yazılımı ve Intel Core Duo T2400 @ 1.83GHz işlemci hızında hesaplanmıştır.

Tablo 3.7. Örnek 3.4. – 3.6. dan elde edilen sonuçlar

	Algoritma 3.2.1	**Algoritma 3.2.2**	**Algoritma 3.2.3**	**Algoritma 3.2.4**
i	206	386	193	185
$\underline{h}$	0.02726122033	0.01571817100	0.02426248719	0.02633035147
$\overline{h}$	0.08058662244	0.03838149463	0.1650334016	0.1815367417
Δ	0.08153505814	0.09585292883	0.02938907637	0.01586486817
CPU	1.07 s	2.37 s	6.34 s	11.96 s

3.2.m (m = 1 , 2 , 3 , 4), algoritma numarası olmak üzere Algoritma 3.2.m ile elde edilen adım sayısını $i^{(m)}$, elde edilen en büyük adım genişliğini $\overline{h}^{(m)}$ ve işlem zamanı $\text{CPU}^{(m)}$ ile gösterelim. Tablo 3.7. de,

$$\text{CPU}^{(1)} < \text{CPU}^{(2)} < \text{CPU}^{(3)} < \text{CPU}^{(4)}$$

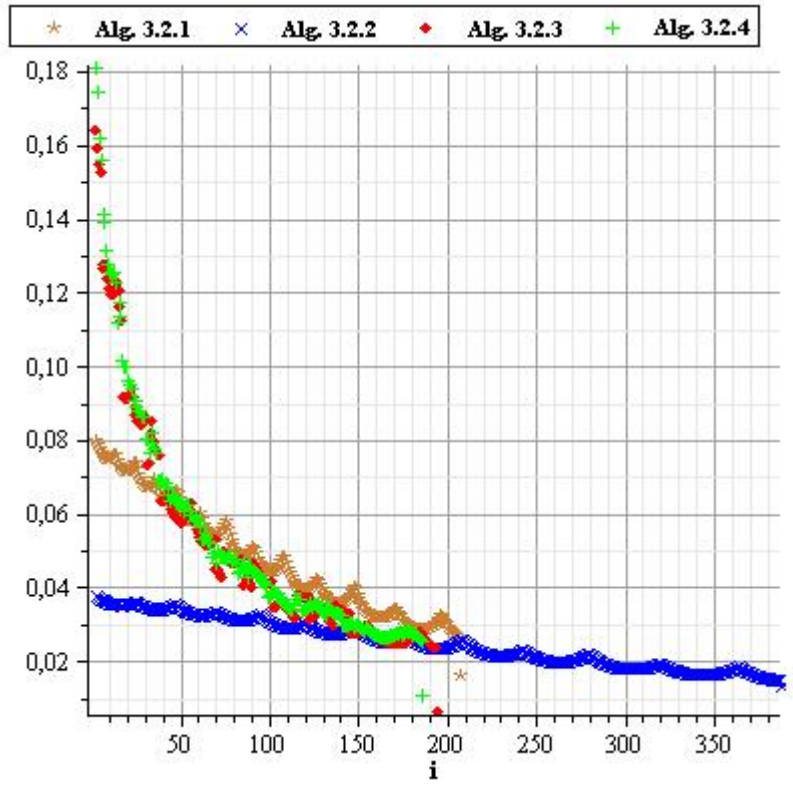
iken,

$$\overline{h}^{(2)} < \overline{h}^{(1)} < \overline{h}^{(3)} < \overline{h}^{(4)}$$

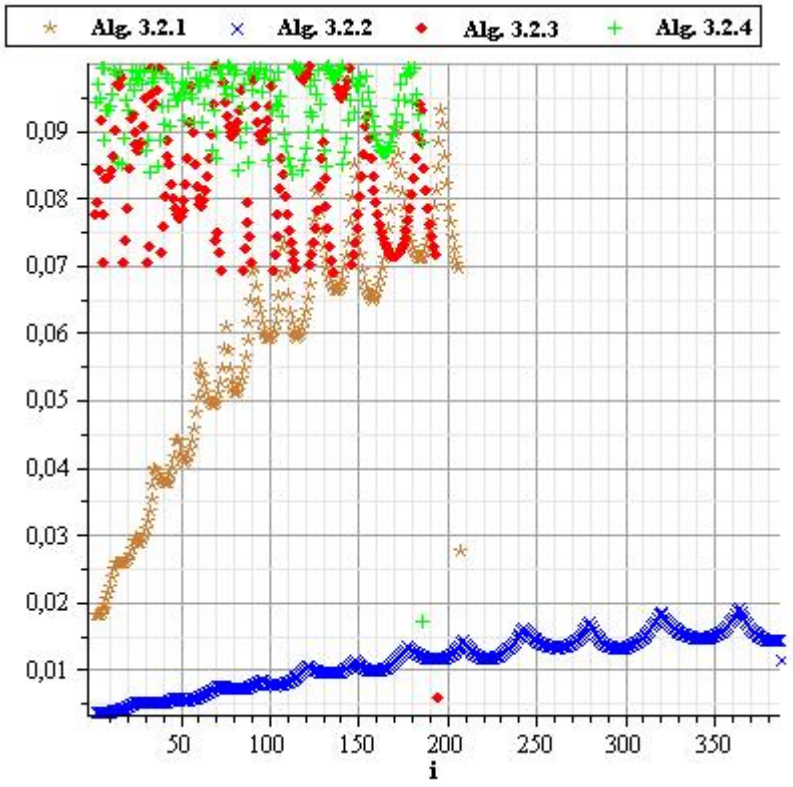
ve

$$i^{(2)} > i^{(1)} > i^{(3)} > i^{(4)}$$

olduğu görülmektedir. Bu durum algoritmaların yapısına uygundur. Buna göre; kullanıcının önceliklerine göre her bir algoritma tercih edilebilir.



Grafik 3.6. Örnek 3.4. – 3.6. da elde edilen $h[i]$ değerleri



Grafik 3.7. Örnek 3.4. – 3.6. da oluşan $\| LE[i] \|$ değerleri

Ayrıca; Strateji 3.2.1 de $||A||$ normu kullanılmakta olduğundan kullanılan normlara göre ve normlar arasındaki eşitsizlikler kullanılarak da adım genişlikleri elde edilebilir. Bu durumda doğal olarak farklı normlar için farklı sayıda adım genişlikleri elde edilebilir. Biz bu çalışmada normlar arasındaki farklılıklardan kaynaklanan durumları ayrı strateji veya ayrı algoritma olarak değerlendirmeyeceğiz. Mevcut strateji ve algoritmalar farklı normlar için de kolaylıkla uygulanabilir.

3.2.7. Ek 1: Algoritmalar İçin Yazılan Prosedürler

Prosedürler Maple 12 programında yazılmıştır. Algoritmalarda kullanılan bazı sembollerin prosedürlerdeki karşılığı aşağıda verilmiştir.

δ_L	:	**dl**
h^*	:	**hstar**
β_k	:	**c[k]**
$\hat{h}_k$	:	**ph[k]**
$t_{k-1} + h_k$	:	**hSum**
Matrisin boyutu	:	**dim**
$\|\|A\|\|$	:	**a** (Algoritma 3.2.1 de)
$\max_{1 \le i,j \le N} \{\| a_{ij} \|\}$	:	**a** (Algoritma 3.2.2, 3.2.3. ve 3.2.4 te)

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar)
global a,c,ph,h, k,hSum,Y,t,Z,LE;
local sub1, sub2, v1, v2;
sub1:=proc()
c[k-1]:=max(b+abs(Y[k-1]));
ph[k]:=evalf((1/a)*sqrt(2*dl/c[k-1]));
hSum:=evalf(t[k-1]+ph[k]);
if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0;
end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
t[k]:=t[k-1]+h[k];
Y[k]:=Y[k-1]+h[k]*MatrixVectorMultiply(A,Y[k-1]);
Z[k]:=MatrixVectorMultiply(MatrixExponential(A*h[k]),Y[k-1]);
LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
v1:={seq([n,h[n]],n=1..k)};
v2:={seq([n,LE[n]],n=1..k)};  print(k,t[k],h[k],LE[k]);
if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if:
end proc:
sub2:=proc()  multiple(plot,[v1],[v2]); end proc:
Y[0]:=X0; Z[0]:=X0; t[0]:=t0; a:=Norm(A,Euclidean);k:=1; sub1();
end proc:
```

Prosedür 3.1. Algoritma 3.2.1 için prosedür

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar)
global a,c,ph,h,dim,k,hSum,Y,t,Z,LE;
local sub1, sub2, v1, v2;
sub1:=proc()
c[k-1]:=max(b+abs(Y[k-1]));
ph[k]:=evalf((1/(a*dim^(5/4)))*sqrt(2*dl/c[k-1]));
hSum:=evalf(t[k-1]+ph[k]);
if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0;
end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
t[k]:=t[k-1]+h[k];
Y[k]:=Y[k-1]+h[k]*MatrixVectorMultiply(A,Y[k-1]);
Z[k]:=MatrixVectorMultiply(MatrixExponential(A*h[k]),Y[k-1]);
LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};
print(k,t[k],h[k],LE[k]);
if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if:
end proc:
sub2:=proc()  multiple(plot,[v1],[v2]); end proc:
dim:=RowDimension(A); Y[0]:=X0; Z[0]:=X0; t[0]:=t0; a:=max(abs(A));
 k:=1; sub1();
end proc:
```

Prosedür 3.2. Algoritma 3.2.2 için prosedür

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,gamma,hstar)
global a,c,ph,h,h1,h2,dim,k,hSum,Y,Y1,t,Z,Z1,LE,LE1,i;
local sub1, sub2, sub3, sub4, sub5,v1, v2, v3, v4;
sub1:=proc()
c[k-1]:=max(b+abs(Y[k-1]));
ph[k]:=evalf((1/(a*dim^(5/4)))*sqrt(2*dl/c[k-1])); h1[k]:=ph[k]; sub2();
end proc:
sub2:=proc()
Y1[k]:=Y[k-1]+h1[k]*MatrixVectorMultiply(A,Y[k-1]);
Z1[k]:=MatrixVectorMultiply(MatrixExponential(A*h1[k]),Y[k-1]);
LE1[k]:=VectorNorm(VectorAdd(Y1[k],-Z1[k]),Euclidean);
if LE1[k]<dl then i:=i+1; h1[k]:=((gamma)^(i-1))*ph[k]; sub2();
else h2[k]:=((gamma)^(i-2))*ph[k]; sub3(); end if:
end proc:
sub3:=proc() hSum:=evalf(t[k-1]+h2[k]);
if (hSum<=T) then
if (h2[k]>hstar) then h[k]:=h2[k]; sub4(); else h[k]:=0; sub5(); end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; sub4(); else h[k]:=0; sub5(); end if:
end proc:
sub4:=proc()
t[k]:=t[k-1]+h[k]; Y[k]:=Y[k-1]+h[k]*MatrixVectorMultiply(A,Y[k-1]);
Z[k]:=MatrixVectorMultiply(MatrixExponential(A*h[k]),Y[k-1]);
LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
if t[k]=T then sub5(); end if:
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};
print(k,t[k],h[k],LE[k]); k:=k+1;i:=1; sub1();
end proc:
sub5:=proc()  multiple(plot,[v1],[v2]); end proc:
dim:=RowDimension(A); Y[0]:=X0; Z[0]:=X0; t[0]:=t0; a:=max(abs(A));
k:=1; i:=1; sub1();
end proc:
```

Prosedür 3.3. Algoritma 3.2.3 için prosedür

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar,gamma)
global a,c,ph,h,h1,h2,dim,k,hSum,Y,Y1,Y2,t,Z,Z1,Z2,LE,LE1,LE2,i,e,j;
local sub1, sub2, sub3, sub4,sub5,sub6, v1, v2, v3, v4;
sub1:=proc()   c[k-1]:=max(b+abs(Y[k-1]));
ph[k]:=evalf((1/(a*dim^(5/4)))*sqrt(2*dl/c[k-1]));
h1[k]:=ph[k];  sub2();  end proc:
sub2:=proc()   Y1[k]:=Y[k-1]+h1[k]*MatrixVectorMultiply(A,Y[k-1]);
Z1[k]:=MatrixVectorMultiply(MatrixExponential(A*h1[k]),Y[k-1]);
LE1[k]:=VectorNorm(VectorAdd(Y1[k],-Z1[k]),Euclidean);
if LE1[k]<dl then  i:=i+1; h1[k]:=(gamma)^(i-1)*ph[k]; sub2();
else sub3(); end if:
end proc:
sub3:=proc()
e[0]:=(gamma)^(i-1); e[j]:=((gamma)^(i-2)+e[j-1])/2;  h2[k]:=e[j]*ph[k];
Y2[k]:=Y[k-1]+h2[k]*MatrixVectorMultiply(A,Y[k-1]);
Z2[k]:=MatrixVectorMultiply(MatrixExponential(A*h2[k]),Y[k-1]);
LE2[k]:=VectorNorm(VectorAdd(Y2[k],-Z2[k]),Euclidean);
if LE2[k]>0.1 then j:=j+1; sub3(); else h[k]:=h2[k];  sub4();  end if:
end proc:
sub4:=proc()  hSum:=evalf(t[k-1]+h2[k]);
if (hSum<=T) then   if (h2[k]>hstar) then h[k]:=h2[k]; sub5();
else h[k]:=0; sub6();end if:  end if:
if hSum>T then   if (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; sub5();
else h[k]:=0; sub6(); end if:  end if:
end proc:
sub5:=proc()   t[k]:=t[k-1]+h[k];
Y[k]:=Y[k-1]+h[k]*MatrixVectorMultiply(A,Y[k-1]);
Z[k]:=MatrixVectorMultiply(MatrixExponential(A*h[k]),Y[k-1]);
LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
if t[k]=T then sub6(); end if:
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};  print(k,t[k],h[k],LE[k]);
k:=k+1;i:=1;j:=1;  sub1();  end proc:
sub6:=proc()  multiple(plot,[v1],[v2]); end proc:
dim:=RowDimension(A); Y[0]:=X0; Z[0]:=X0; t[0]:=t0; a:=max(abs(A));
k:=1; i:=1; j:=1;sub1();
end proc:
```

Prosedür 3.4. Algoritma 3.2.4 için prosedür

3.3. $F(t, X) = AX + \varphi(t, X)$ Durumunda Adım Genişliği Stratejileri ve Algoritmaları

$A = (a_{ij}) \in \boldsymbol{R}^{N\times N}$, $X \in \boldsymbol{R}^{N}$ ve $\varphi \in C^{m}([t_0, T]\times \boldsymbol{R}^{N})$ olmak üzere

$$X'(t) = AX(t) + \varphi(t, X),\ X(t_0) = X_0 \tag{3.17}$$

Cauchy problemini

$$D = \{(t, X) : t \in [t_0, T], |x_j - x_{j0}| \le b_j\}$$

bölgesi üzerinde ele alalım. Bu kısımda; (3.3) lineer Cauchy problemi için 3.2. kısımda verilen stratejilerden yola çıkarak (3.17) Cauchy problemi için adım genişliği stratejileri elde edeceğiz.

(3.17) Cauchy probleminin lokal hatası ile ilgili olarak aşağıdaki teoremi verelim.

Teorem 3.3. (3.17) probleminin lokal hatası için;

$$\| LE_i \| \le \frac{1}{2} h_i^2 \{ \| A \|^2 \| Z(\tau_{i,j}) \| + \| A \| \| \varphi(\tau_{i,j}, Z(\tau_{i,j})) \| + \| \varphi'(\tau_{i,j}, Z(\tau_{i,j})) \| \} \tag{3.18}$$

eşitsizliği doğrudur. Burada $\tau_{i,j}$ değerinde; i, $\tau_{i,j}$ sayısının bulunduğu aralığı ve j ise $Z(t)$ vektörünün j. bileşenini göstermektedir.

İspat. (3.17) Cauchy problemine *lokal hata analizi* uygulanırsa,

$$\begin{aligned} LE_i &= Y_i - Z(t_i) \\ &= Y_{i-1} + h_i[AY_{i-1} + \varphi_{i-1}] - Z(t_{i-1}) - h_i[AZ(t_{i-1}) + \varphi(t_{i-1}, Z(t_{i-1}))] - \frac{1}{2} h_i^2 Z''(\tau_{i,j}) \\ &= Y_{i-1} + h_i[AY_{i-1} + \varphi_{i-1}] - Y_{i-1} - h_i[AY_{i-1} + \varphi_{i-1}] - \frac{1}{2} h_i^2 Z''(\tau_{i,j}) = -\frac{1}{2} h_i^2 Z''(\tau_{i,j}) \\ &= -\frac{1}{2} h_i^2 [A^2 Z(\tau_{i,j}) + A\varphi(\tau_{i,j}, Z(\tau_{i,j})) + \varphi'(\tau_{i,j}, Z(\tau_{i,j}))] \end{aligned}$$

$$LE_i = -\frac{1}{2} h_i^2 [A^2 Z(\tau_{i,j}) + A\varphi(\tau_{i,j}, Z(\tau_{i,j})) + \varphi'(\tau_{i,j}, Z(\tau_{i,j}))],\ \tau_{i,j} \in [t_{i-1}, t_i) \tag{3.19}$$

elde edilir. Lokal hatanın üst sınırını bulmak için (3.19) dan norm alınırsa;

$$\| LE_i \| = \frac{1}{2} h_i^2 \| A^2 Z(\tau_{i,j}) + A\varphi(\tau_{i,j}, Z(\tau_{i,j})) + \varphi'(\tau_{i,j}, Z(\tau_{i,j})) \|, \tau_{i,j} \in [t_{i-1}, t_i)$$

$$\leq \frac{1}{2} h_i^2 \{ \| A^2 Z(\tau_{i,j}) \| + \| A\varphi(\tau_{i,j}, Z(\tau_{i,j})) \| + \| \varphi'(\tau_{i,j}, Z(\tau_{i,j})) \| \}$$

$$\| LE_i \| \leq \frac{1}{2} h_i^2 \{ \| A \|^2 \| Z(\tau_{i,j}) \| + \| A \| \| \varphi(\tau_{i,j}, Z(\tau_{i,j})) \| + \| \varphi'(\tau_{i,j}, Z(\tau_{i,j})) \| \}$$

eşitsizliği elde edilir. □

3.3.1. Strateji ve Algoritma

Teorik olarak; lokal hata δ_L - hata seviyesinden küçük olacak şekilde nümerik integrasyonun *i*. adımında adım genişliği;

$$\max_{1 \leq j \leq N} (\sup_{t_{i-1} \leq \tau_{i,j} < t_i} | z_j(\tau_{i,j}) |) \leq \beta_{i-1}, \quad \max_{1 \leq j \leq N} (\sup_{t_{i-1} \leq \tau_{i,j} < t_i} | \varphi_j(\tau_{i,j}, z(\tau_{i,j})) |) \leq \gamma_{i-1},$$

$$\max_{1 \leq j \leq N} (\sup_{t_{i-1} \leq \tau_{i,j} < t_i} | \frac{d\varphi_j}{dt}(\tau_{i,j}, z(\tau_{i,j})) |) \leq \zeta_{i-1}$$

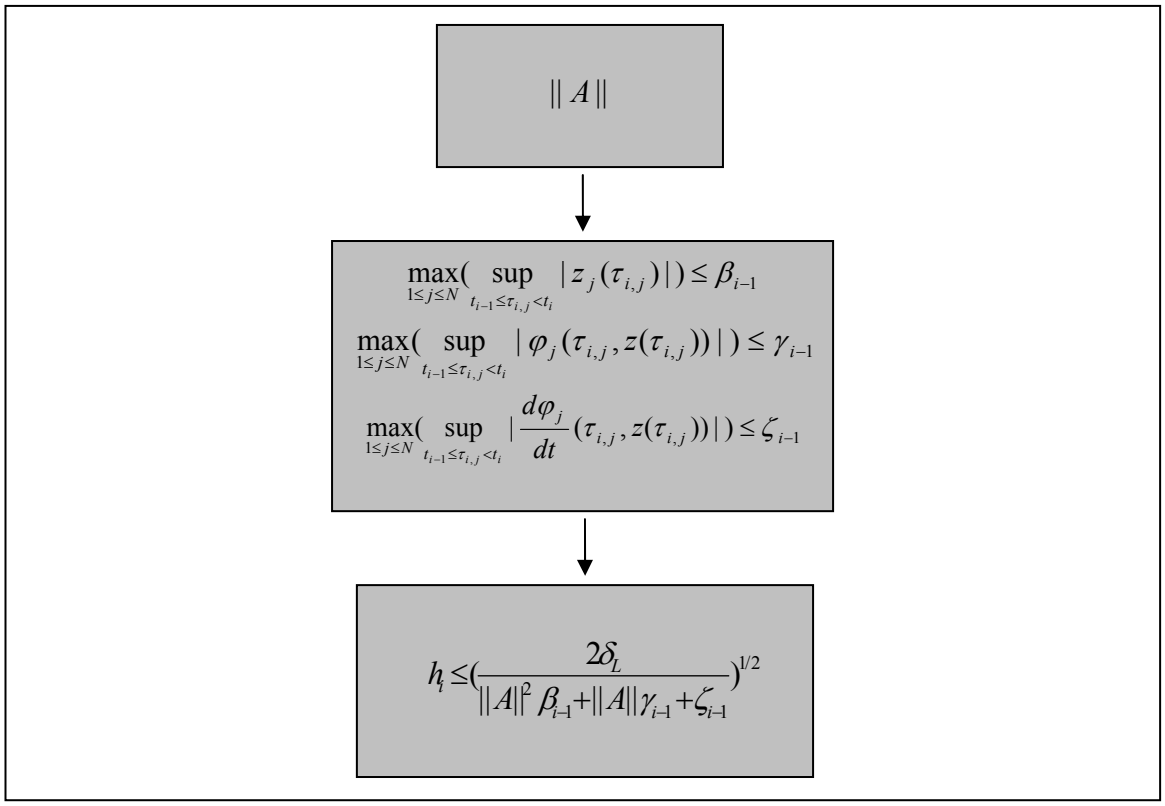
olmak üzere (3.18) eşitsizliğinden,

$$h_i \leq \left(\frac{2\delta_L}{\| A \|^2 \beta_{i-1} + \| A \| \gamma_{i-1} + \zeta_{i-1}} \right)^{1/2} \qquad (3.20)$$

ile hesaplanır.

Sonuç 3.2. (3.17) Cauchy probleminde $\varphi(t, X) = 0$ olması durumunda $\gamma_{i-1} = \zeta_{i-1} = 0$ olduğundan (3.20) eşitsizliği ile verilen adım genişliği stratejisinin (3.6) eşitsizliğine dönüştüğü açıktır.

Şekil 3.5. de Strateji 3.3.1 verilmiştir.



Şekil 3.5. *i*. adımda adım genişliğinin Strateji 3.3.1 e göre hesaplanması

(3.20) eşitsizliğine uygun adım genişlikleri ile nümerik çözüm hesaplayan algoritmayı verelim.

0. Adım (Giriş Elemanları): t_0 , T , b , h^* , δ_L , X_0 , $\varphi(t, X)$, A

1. Adım: $\| A \|$ sayısı hesaplanır ve $\frac{d\varphi(t,X)}{dt}$ türevi bulunur.

2. Adım: $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})|) \le \beta_{i-1}$, $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\varphi_j(\tau_{i,j}, z(\tau_{i,j}))|) \le \gamma_{i-1}$,

$\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\frac{d\varphi_j}{dt}(\tau_{i,j}, z(\tau_{i,j}))|) \le \zeta_{i-1}$ sayıları hesaplanır.

3. Adım: $\hat{h}_i \le (\frac{2\delta_L}{\| A \|^2 \beta_{i-1} + \| A \| \gamma_{i-1} + \zeta_{i-1}})^{1/2}$ adım genişliği hesaplanır.

4. Adım: K- adım genişliği kontrolü yapılır.

5. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i A) Y_{i-1} + h_i \varphi(t_{i-1}, Y_{i-1})$ hesaplanır.

Algoritma 3.3.1. (3.20) deki adım genişlikleri ile nümerik çözüm hesaplayan algoritma

Örnek 3.8. $D=\{(t,x_i): t\in[0,1], |x_i - x_{i0}|\le 5\}$ bölgesi üzerinde

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} tx_1^2 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{3.21}$$

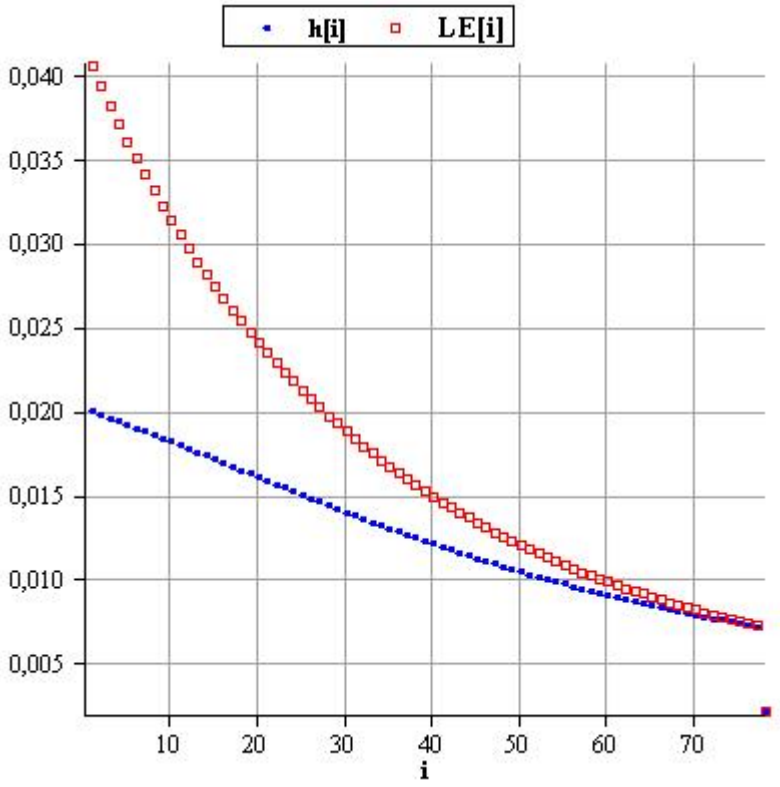
Cauchy probleminin nümerik çözümü $h^* = 10^{-12}$, $\delta_L = 10^{-1}$ alarak Algoritma 3.3.1 yardımıyla hesaplandığında elde edilen adım genişlikleri ile oluşan lokal hatalar Tablo 3.8. de ve Grafik 3.8. de görülmektedir.

Tablo 3.8. Algoritma 3.3.1 ile elde edilen h_i ve $\| LE_i \|$ değerleri

i	h_i	$\| LE_i \|$
1	0.2016194596e-1	0.407470943905615878e-1
2	0.1997175123e-1	0.395539052221946102e-1
3	0.1977882650e-1	0.384095493282268636e-1
4	0.1958332270e-1	0.373109372557251101e-1
⋮	⋮	⋮
76	0.7452259853e-2	0.753009559807405263e-2
77	0.7358505219e-2	0.740710603606112815e-2
78	0.23086070e-2	0.229389393527329672e-2

Tablo 3.8. ve Grafik 3.8. de (3.20) deki şekilde adım genişliği ile nümerik çözüm hesaplandığında oluşan lokal hatanın istenen hata seviyesinden küçük kaldığı görülmektedir.

Uyarı 3.4. 3.2.1. kısmında; kayan noktalı sayıların kullanımının, lokal hatası *gerçekte* δ_L hata seviyesine çok yakın oluşan problemlerde $\| LE_i \| > \delta_L$ olacak şekilde gerçekleşmesine neden olabildiği belirtilmişti. (3.17) Cauchy probleminin nümerik integrasyonu için (3.20) eşitsizliğinden adım genişlikleri seçildiğinde de benzer durum karşımıza çıkmaktadır.



Grafik 3.8. Algoritma 3.3.1 ile elde edilen h_i ve $\| LE_i \|$

Şimdi aşağıdaki örneği inceleyelim.

Örnek 3.9. $h^* = 10^{-12}, \delta_L = 10^{-1}$ değerleri ile $D = \{(t, x_i) : t \in [0,\ 0.05],\ |x_i - x_{i0}| \le 5\}$ bölgesi üzerinde

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 100 & 0 \\ 25 & 100 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2\cos t - 3x_1^2 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{3.22}$$

Cauchy probleminin nümerik çözümü Algoritma 3.3.1 yardımıyla hesaplandığında elde edilen adım genişlikleri ve oluşan lokal hatalar Tablo 3.9. da özetlenmiştir.

(3.22) Cauchy problemi için (3.20) deki adım genişlikleri kullanılarak çözüm hesaplandığında 45. adımdan itibaren oluşan lokal hatanın $\delta_L = 10^{-1}$ hata seviyesini geçmeye başladığı Tablo 3.9. da görülmektedir. Son adımda ise adım genişliği $h_{649} = T - t_{648}$ olduğundan $\| LE_{649} \| < \delta_L$ dir.

Tablo 3.9. Algoritma 3.3.1 ile elde edilen h_i ve $\| LE_i \|$ değerleri

i	h_i	$\| LE_i \|$
1	0.1084446335e-2	0.1048588864e-1
2	0.1063240699e-2	0.1112144855e-1
3	0.1040893214e-2	0.1169841882e-1
⋮	⋮	⋮
44	0.2735741110e-3	0. 9900823672e-1
45	0.2661246677e-3	**0.1020425554**
⋮	⋮	⋮
648	0.1113722222e-4	**0.2014700172**
649	0.719034e-5	**0.8423000715e-1**

Kayan nokta aritmetiğinin sebep olduğu etkilerden tamamen kurtulmak mümkün olmasa da; lineer denklem sistemlerinin nümerik integrasyonu için adım genişliği seçerken lokal hatanın üst sınırını *yeterince* büyüterek bu etkilerden kaçınmıştık. (3.17) Cauchy probleminin nümerik çözümü için de benzer şekilde yeni bir strateji verebiliriz. Lokal hatanın (3.18) üst sınırından daha büyük olan bir üst sınır yardımıyla adım genişlikleri hesaplanırsa; oluşan lokal hatalar daha küçük olacağından, istenilen hata seviyesini geçmez.

3.3.2. Strateji ve Algoritma

Lokal hatanın üst sınırı ile ilgili olan aşağıdaki teoremi verelim.

Teorem 3.4. (3.17) sisteminin lokal hatası için

$$\alpha = \max_{1\le i,j\le N} |\, a_{ij} \,|,\ \max_{1\le j\le N}\Big(\sup_{t_{i-1}\le \tau_i < t_i} |\, z_j(\tau_i) \,|\Big) \le \beta_{i-1},\ \max_{1\le j\le N}\Big(\sup_{t_{i-1}\le \tau_i < t_i} |\, \varphi_j(\tau_i, z(\tau_i)) \,|\Big) \le \gamma_{i-1},$$

$$\max_{1\le j\le N}\Big(\sup_{t_{i-1}\le \tau_i < t_i} |\, \frac{d\varphi_j}{dt}(\tau_i, z(\tau_i)) \,|\Big) \le \zeta_{i-1}$$

olmak üzere

$$\| LE_i \| \le \frac{1}{2}\sqrt{N}\ h_i^2 [N^2\alpha^2\beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}] \qquad (3.23)$$

eşitsizliği doğrudur.

İspat. (3.18) eşitsizliğinde

$$\alpha = \max_{1\le i,j\le N} |\, a_{ij} \,|,\ \max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_i < t_i} |\, z_j(\tau_i) \,|) \le \beta_{i-1},\ \max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_i < t_i} |\, \varphi_j(\tau_i, z(\tau_i)) \,|) \le \gamma_{i-1},$$

$$\max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_i < t_i} |\frac{d\varphi_j}{dt}(\tau_i, z(\tau_i)) \,|) \le \zeta_{i-1}$$

alınırsa,

$$\| A \| \le N\alpha\ ,\ \| Z \| \le \sqrt{N}\beta_{i-1},\ \| \varphi \| \le \sqrt{N}\gamma_{i-1},\ \| \frac{d\varphi}{dt} \| \le \sqrt{N}\zeta_{i-1}$$

olur. Buna göre;

$$\| LE_i \| \le \frac{1}{2}\sqrt{N}\ h_i^2[N^2\alpha^2\beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}]$$

elde edilir. □

Buna göre; (3.17) Cauchy probleminin nümerik integrasyonunun *i*. adımında lokal hata δ_L - hata seviyesinden küçük olacak şekilde adım genişliği;

- $\alpha = \max_{1\le i,j\le N} |\, a_{ij} \,|$, $\max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_{i,j} < t_i} |\, z_j(\tau_{i,j}) \,|) \le \beta_{i-1}$,
- $\max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_i < t_i} |\, \varphi_j(\tau_i, z(\tau_i)) \,|) \le \gamma_{i-1}$, $\max_{1\le j\le N}(\sup_{t_{i-1}\le \tau_i < t_i} |\frac{d\varphi_j}{dt}(\tau_i, z(\tau_i)) \,|) \le \zeta_{i-1}$

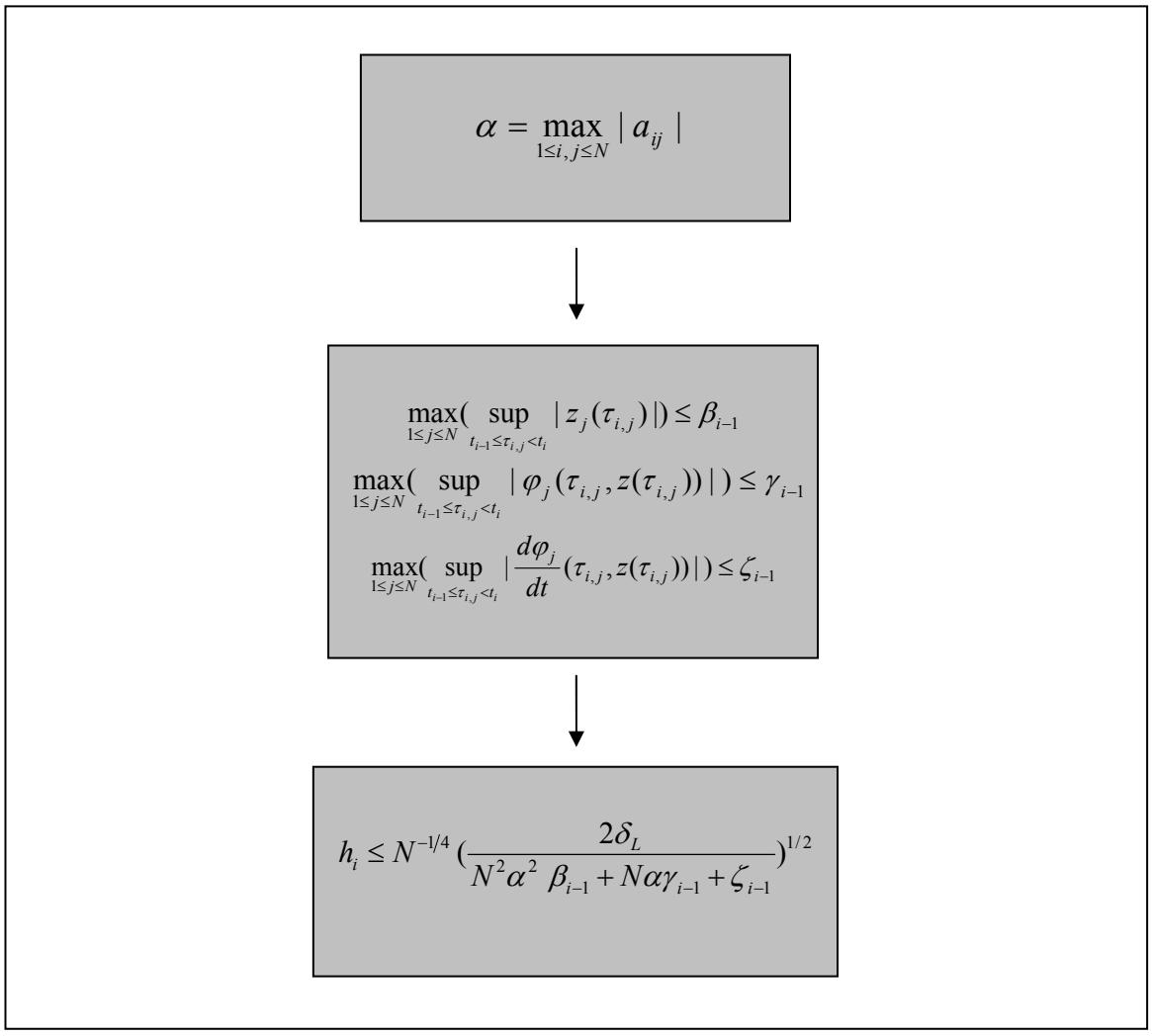
olmak üzere

$$h_i \le N^{-1/4}\left(\frac{2\delta_L}{N^2\alpha^2\ \beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}}\right)^{1/2} \qquad (3.24)$$

eşitsizliğinden hesaplanır.

Uyarı 3.5. (3.2.4) eşitsizliğindeki adım genişlikleri (3.20) eşitsizliğindeki adım genişliklerinden daha küçük olduğundan; aynı bölgede Algoritma 3.3.2 ile hesaplama yapıldığında adım sayısı, Algoritma 3.3.1 deki adım sayısından daha fazladır.

i. adımda adım genişliğinin Strateji 3.3.2 ye göre hesaplanması aşağıdaki şemada gösterilmiştir.



Şekil 3.6. *i*. adımda adım genişliğinin Strateji 3.3.2 ye göre hesaplanması

(3.24) eşitsizliğine uygun adım genişlikleri ile nümerik çözüm hesaplayan algoritmayı verelim.

0. Adım (Giriş Elemanları): t_0 , T , b , h^* , δ_L , X_0 , $\varphi(t,X)$, A

1. Adım: $\alpha = \max_{1\le i,j\le N} |a_{ij}|$ sayısı hesaplanır ve $\frac{d\varphi(t,X)}{dt}$ türevi bulunur.

2. Adım: $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})|) \le \beta_{i-1}$, $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\varphi_j(\tau_{i,j}, z(\tau_{i,j}))|) \le \gamma_{i-1}$,

$\max_{1\le j\le N}(\sup_{t_{i-1}<\tau_i<t_i} |\frac{d\varphi_j}{dt}(\tau_i, z(\tau_i))|) \le \zeta_{i-1}$ sayıları hesaplanır.

3. Adım: $\hat{h}_i \le N^{-1/4}\left(\frac{2\delta_L}{N^2\alpha^2\ \beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}}\right)^{1/2}$ adım genişliği hesaplanır.

4. Adım: K- adım genişliği kontrolü yapılır.

5. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i\ A)Y_{i-1} + h_i\ \varphi(t_{i-1}, Y_{i-1})$ hesaplanır.

Algoritma 3.3.2. (3.24) deki adım genişlikleri ile nümerik çözüm hesaplayan algoritma

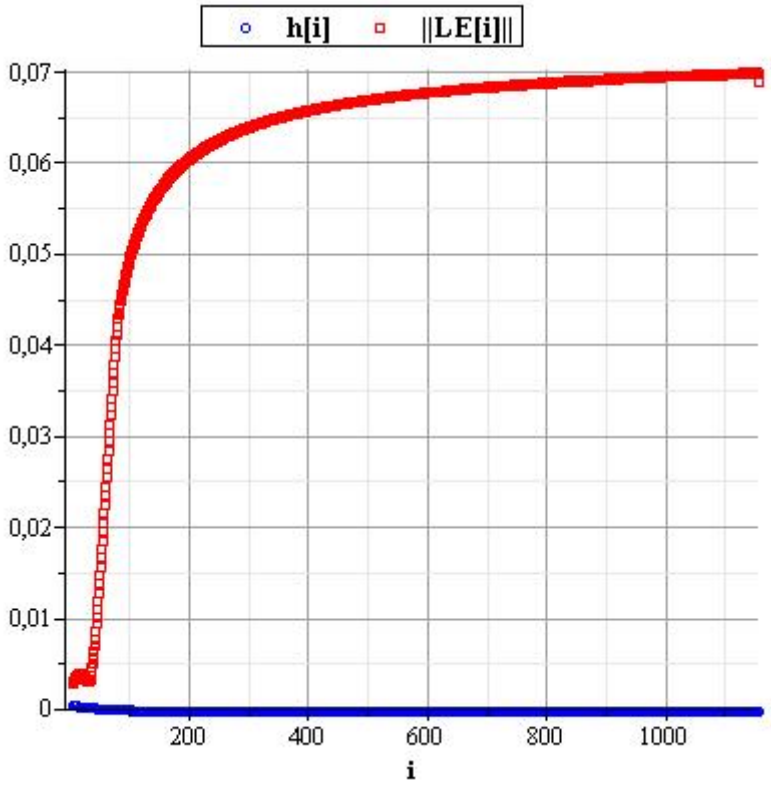
Örnek 3.10. $D=\{(t,x_i): t\in[0\ ,\ 0.05],\ |x_i - x_{i0}|\le 5\}$ bölgesi üzerinde (3.22) ile verilen

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} 100 & 0 \\ 25 & 100 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2\cos t - 3x_1^2 \end{pmatrix},\ \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Cauchy probleminin nümerik çözümünü $h^* = 10^{-12}$ ve $\delta_L = 10^{-1}$ alarak Algoritma 3.3.2 yardımıyla yeniden hesaplayalım. Elde edilen adım genişlikleri ile oluşan lokal hatalar Tablo 3.10. ve Grafik 3.9. da görülmektedir.

Tablo 3.10. Algoritma 3.3.2 ile elde edilen h_i ve $\|LE_i\|$ değerleri

i	h_i	$\|LE_i\|$
1	0.6026096827e-3	0.3186671007e-2
2	0.5968708670e-3	0.3309187798e-2
3	0.5909294394e-3	0.3428175150e-2
4	0.5847906204e-3	0.3542497276e-2
⋮	⋮	⋮
1151	0.6173581407e-5	0. 7007000505e-1
1152	0.6167964996e-5	0.7008000505e-1
1153	0.612070e-5	0.6922000499e-1



Grafik 3.9. Algoritma 3.3.2 ile elde edilen h_i ve $\| LE_i \|$ değerleri

Tablo 3.9. , Tablo 3.10 ve Grafik 3.9. incelendiğinde; (3.22) Cauchy probleminin nümerik çözümünde oluşan lokal hatanın istenilen hata seviyesini; (3.20) eşitsizliğinden elde edilen adım genişlikleri kullanıldığında geçerken (3.24) eşitsizliğinden elde edilen adım genişlikleri kullanıldığında geçmediği görülmektedir.

Not 3.2. Eğer, (3.17) Cauchy probleminin nümerik çözümünde lokal hatanın δ_L hata seviyesine daha yakın gerçekleşmesi istenirse, kısım 3.2. de verilen Strateji 3.2.3 ve Strateji 3.2.4 e benzer olarak adım genişliği stratejileri verilebilir. Bu çalışmada, 3.2. kısımdaki Strateji 3.2.3 ve 3.2.4 e benzer olan stratejiler verilmeyecektir.

Şimdi, (3.3) lineer Cauchy problemi için verilen Strateji 3.2.2 den yararlanarak (3.17) Cauchy problemi için *yeni* bir adım genişliği stratejisi verelim.

3.3.3. Strateji ve Algoritma

Kısım 3.2.2. de (3.3) ile verilen

$$X'(t) = AX(t)\,,\ X(t_0) = X_0$$

lineer Cauchy probleminin nümerik integrasyonu için, δ_L kullanıcının belirlediği hata seviyesi ve $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})|) \le \beta_{i-1}$ olmak üzere, adım genişliğinin

$$\hat{h}_i \le \frac{1}{\alpha\sqrt[4]{N^5}}\left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}}$$

eşitsizliğinden seçilmesi önerilmişti. Eğer;

$$\widehat{h}_i = \frac{1}{\alpha\sqrt[4]{N^5}}\left(\frac{2\delta_L}{\beta_{i-1}}\right)^{\frac{1}{2}}$$

adım genişlikleri ile (3.17) Cauchy probleminin nümerik çözümü hesaplanırsa integrasyonun *i* inci adımında lokal hata için,

$$\| LE_i \| \le \frac{1}{2}\sqrt{N}\ \widehat{h}_i^2[N^2\alpha^2\beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}] = \xi_i\delta_L$$

olacak şekilde bir ξ_i reel sayısı vardır. Buradan;

$$\delta_L = \frac{1}{2}\left(\frac{\widehat{h}_i}{\sqrt{\xi_i}}\right)^2\sqrt{N}\,[N^2\alpha^2\beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}]$$

olduğundan adım genişliği

$$h_i = \frac{\widehat{h}_i}{\sqrt{\xi_i}}$$

alınırsa, (3.17) Cauchy probleminin çözümü δ_L hata seviyesinde hesaplanmış olur. O halde, (3.17) Cauchy probleminin nümerik integrasyonunun *i*. adımında lokal hata δ_L - hata seviyesinden küçük olacak şekilde adım genişliği;

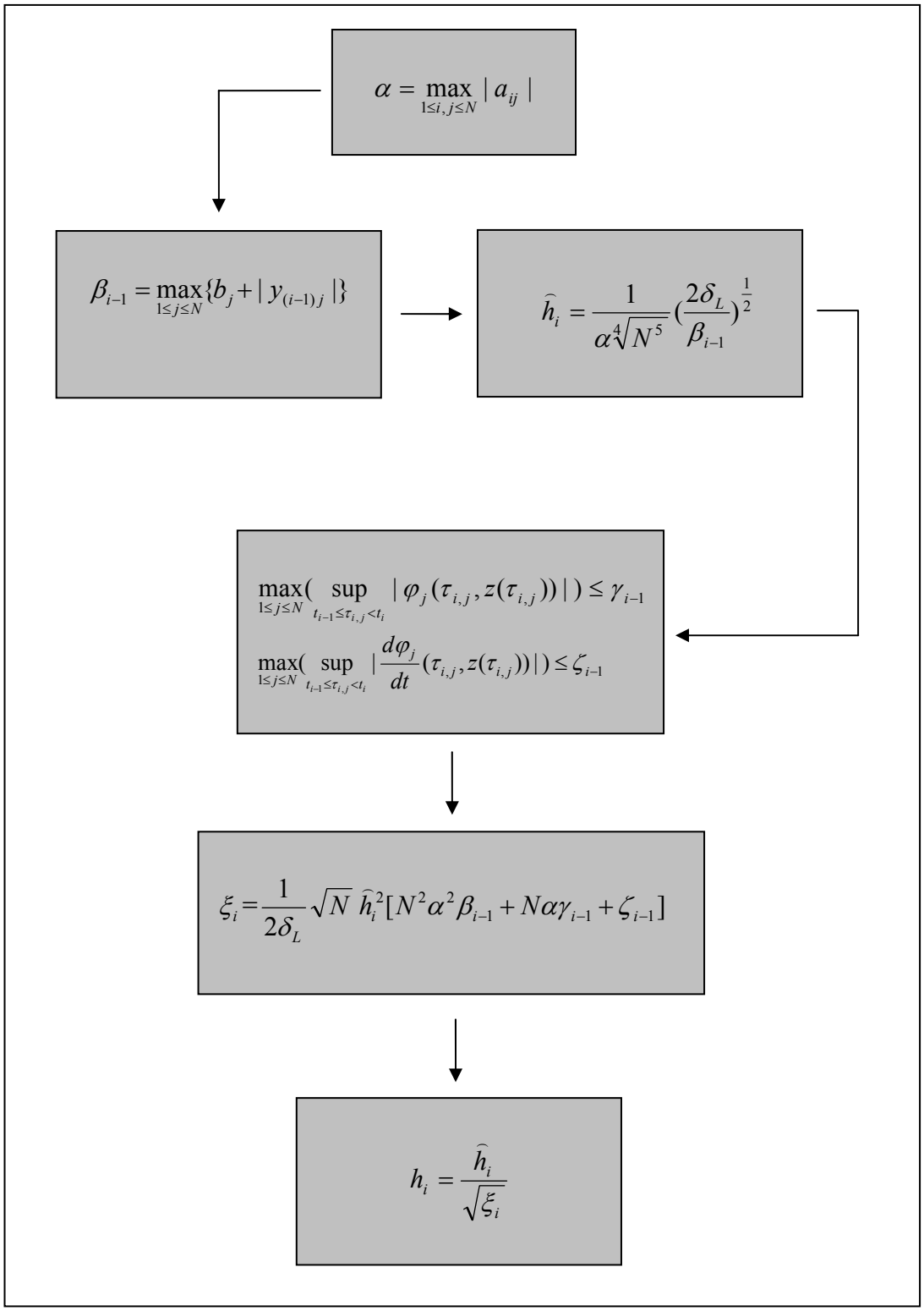
- $\alpha = \max_{1\le i,j\le N} |a_{ij}|$, $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})|) \le \beta_{i-1}$,
- $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\varphi_j(\tau_{i,j}, z(\tau_{i,j}))|) \le \gamma_{i-1}$,
- $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\frac{d\varphi_j}{dt}(\tau_{i,j}, z(\tau_{i,j}))|) \le \zeta_{i-1}$,
- $\xi_i = \frac{1}{2\delta_L}\sqrt{N}\ \hat{h}_i^2[N^2\alpha^2\beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}]$

olmak üzere

$$\hat{h}_i = \frac{1}{\alpha\sqrt[4]{N^5}}(\frac{2\delta_L}{\beta_{i-1}})^{\frac{1}{2}} \quad ; \quad h_i = \frac{\hat{h}_i}{\sqrt{\xi_i}} \tag{3.25}$$

şeklinde hesaplanır.

Strateji 3.3.3 ün şeması aşağıda verilmiştir.



Şekil 3.7 *i*. adımda adım genişliğinin Strateji 3.3.3 e göre hesaplanması

(3.17) Cauchy probleminin nümerik çözümünü (3.25) eşitliğindeki adım genişlikleri ile hesaplayan algoritmayı verelim.

0. Adım (Giriş Elemanları): t_0 , T , b , h^* , δ_L , X_0 , $\varphi(t,X)$, A

1. Adım: $\alpha = \max_{1\le i,j\le N} |a_{ij}|$ sayısı hesaplanır ve $\frac{d\varphi(t,X)}{dt}$ türevi bulunur.

2. Adım: $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |z_j(\tau_{i,j})|) \le \beta_{i-1}$ sayısı hesaplanır.

3. Adım: $\widehat{h}_i = \frac{1}{\alpha\sqrt[4]{N^5}}(\frac{2\delta_L}{\beta_{i-1}})^{\frac{1}{2}}$ hesaplanır.

4. Adım: $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\varphi_j(\tau_{i,j},z(\tau_{i,j}))|) \le \gamma_{i-1}$, $\max_{1\le j\le N}(\sup_{t_{i-1}\le\tau_{i,j}<t_i} |\frac{d\varphi_j}{dt}(\tau_{i,j},z(\tau_{i,j}))|) \le \zeta_{i-1}$

sayıları hesaplanır.

5. Adım: $\xi_i = \frac{1}{2\delta_L}\sqrt{N}\,\widehat{h}_i^2[N^2\alpha^2\beta_{i-1} + N\alpha\gamma_{i-1} + \zeta_{i-1}]$ katsayısı hesaplanır.

6. Adım: $\hat{h}_i = \frac{\widehat{h}_i}{\sqrt{\xi_i}}$ adım genişliği hesaplanır.

7. Adım: K- adım genişliği kontrolü yapılır.

8. Adım: $t_i = t_{i-1} + h_i$ ve $Y_i = (I + h_i\,A)Y_{i-1} + h_i\,\varphi(t_{i-1},Y_{i-1})$ hesaplanır.

Algoritma 3.3.3. (3.25) daki adım genişlikleri ile nümerik çözüm hesaplayan algoritma

Not 3.3. Algoritma 3.3.3 te hesaplanan γ_{i-1} ve ζ_{i-1} sayıları, Algoritma 3.3.2 de hesaplanan γ_{i-1} ve ζ_{i-1} sayılarına ya eşittir ya da bu sayılardan daha küçüktür. Bu sayıların eşit olması durumunda Algoritma 3.3.2 de hesaplanan adım genişlikleri ile Algoritma 3.3.3 de hesaplanan adım genişlikleri eşittir. Küçük olması durumunda ise Algoritma 3.3.3 ten elde edilen adım genişlikleri daha büyük olur. Dolayısıyla hesaplama işlemi daha az adımda tamamlanır.

Örnek 3.11. $D=\{(t,X):t\in[0,1],|\,x_j-x_{j0}\,|\le 5\}$ bölgesi üzerinde

$$\begin{pmatrix}x_1'\\x_2'\end{pmatrix}=\begin{pmatrix}-1 & 1\\0 & 2\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}+\begin{pmatrix}t\ x_2^2\\0\end{pmatrix},\ \begin{pmatrix}x_1(0)\\x_2(0)\end{pmatrix}=\begin{pmatrix}1\\1\end{pmatrix} \tag{3.26}$$

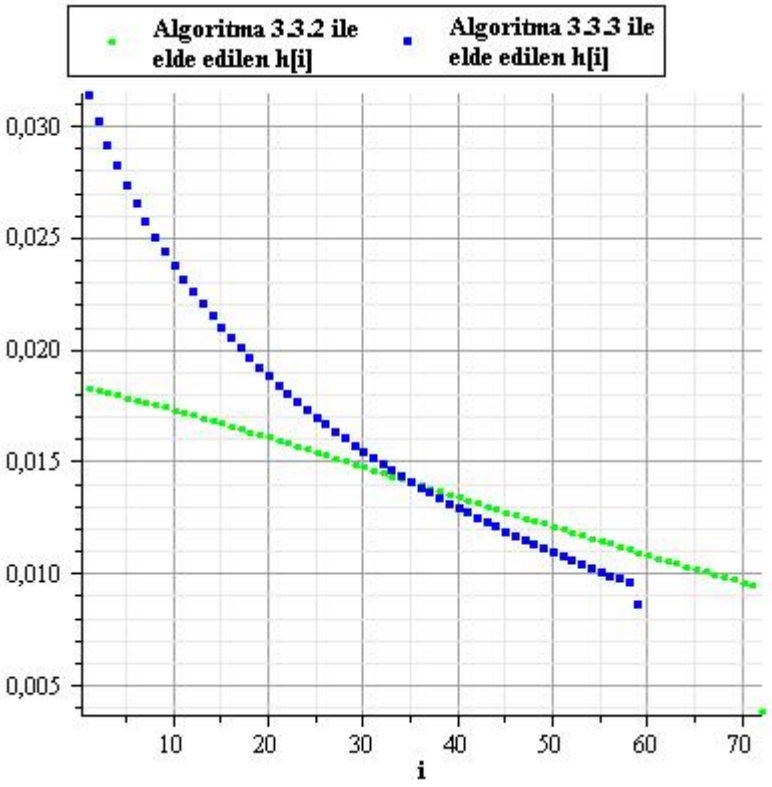
probleminin çözümü $h^*=10^{-12}$ ve hata seviyesi $\delta_L=10^{-1}$ için Algoritma 3.3.2 ve Algoritma 3.3.3 ile hesaplandığında elde edilen adım genişlikleri Tablo 3.11.a. da ve oluşan lokal hatalar Tablo 3.11.b de özetlenmiştir. Ayrıca, Tablo 3.11.a. ve Tablo 3.11.b. de verilen değerlerin grafikleri Grafik 3.10.a ve Grafik 3.10.b. de verilmiştir.

Tablo 3.11.a. Algoritma 3.3.2 ve Algoritma 3.3.3 ile elde edilen adım genişlikleri

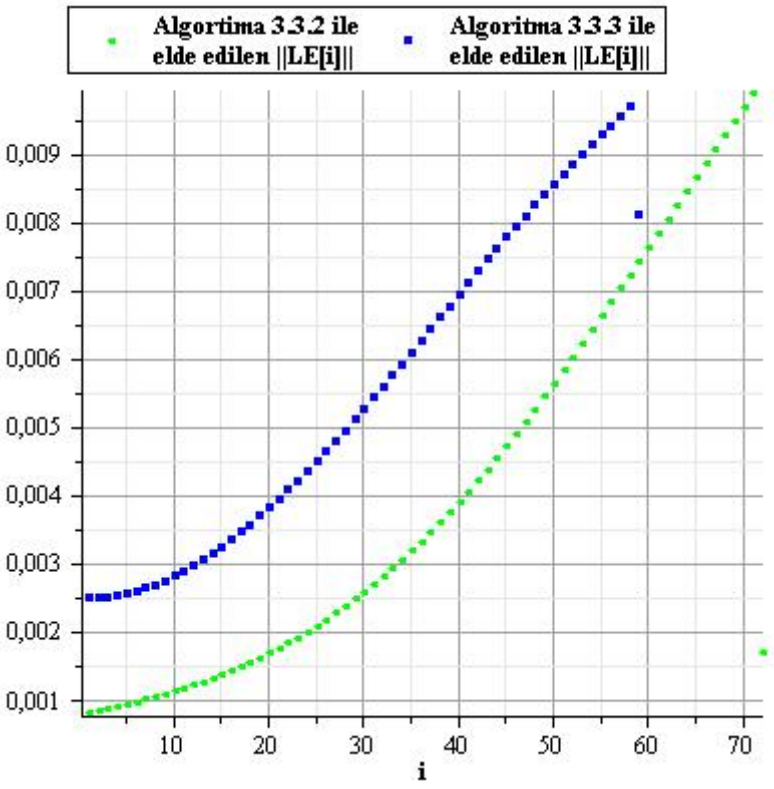
Algoritma 3.3.2		**Algoritma 3.3.3**	
i	h_i	***i***	h_i
1	0.1834986417e-1	**1**	0.3144182114e-1
2	0.1825099242e-1	**2**	0.3028033082e-1
3	0.1815016296e-1	**3**	0.2922982207e-1
4	0.1804739919e-1	**4**	0.2827079649e-1
⋮	⋮	⋮	⋮
58	0.1116133861e-1	**58**	0.9676055086e-1
59	0.1103262530e-1	**59**	0.86677998e-2
⋮	⋮		
71	0.9593767015e-2		
72	0.39269291e-2		

Tablo 3.11.b. Algoritma 3.3.2 ve Algoritma 3.3.3 ile elde edilen $\|\,LE_i\,\|$ değerleri

Algoritma 3.3.2		**Algoritma 3.3.3**	
i	$\|\,LE_i\,\|$	***i***	$\|\,LE_i\,\|$
1	0.854145975463057933e-3	**1**	0.253434857681399595e-2
2	0.882367478862261914e-3	**2**	0.252988998091363896e-2
3	0.911950139731815117e-3	**3**	0.253757504469621920e-2
4	0.942975142824566929e-3	**4**	0.255583033530652674e-2
⋮	⋮	⋮	⋮
58	0.726753766053366002e-2	**58**	0.971581200149233132e-2
59	0.746855669929858118e-2	**59**	0.813582921141354533e-2
⋮	⋮		
71	0.992722626996024940e-2		
72	0.172257226914432869e-2		



Grafik 3.10.a. Algoritma 3.3.2 ve Algoritma 3.3.3 den elde edilen adım genişlikleri



Grafik 3.10.b. Algoritma 3.3.2 ve Algoritma 3.3.3 de oluşan lokal hatalar

Tablo 3.11.a. , Tablo 3.11.b. , Grafik 3.10.a. ve Grafik 3.10.b. de Algoritma 3.3.3 e göre yapılan hesaplamada iterasyon sayısının daha az olduğu görülmektedir.

3.3.4. Değerlendirme

3.3. kısımda, (3.17) Cauchy probleminin nümerik integrasyonunda adım genişliği seçimi için, üç Strateji ve bu stratejilere uygun hesaplama yapan algoritmalar verilmiştir. Aşağıdaki tablo, verilen algoritmaları özetlemektedir.

ALGORİTMA 3.3.1	• (3.3) lineer sistemi için verilen Algoritma 3.2.1 in (3.17) Cauchy problemine genelleştirilmesidir. • Bazı problemlerde, kayan nokta aritmetiğinden kaynaklanan hesaplama hataları nedeniyle, lokal hata istenilen hata seviyesini geçebilmektedir. Kullanıcının hata seviyesini esnek bıraktığı durumlarda Algoritma 3.3.1 kullanılabilir.
ALGORİTMA 3.3.2	• Algoritma 3.3.1 den elde edilmiştir. • Her bir adımda lokal hata, istenilen hata seviyesinden küçük gerçekleşmektedir. • Hesaplama işleminde adım sayısı, Algoritma 3.3.1 deki adım sayısından daha fazladır.
ALGORİTMA 3.3.3	• (3.3) lineer sistemi için verilen adım genişliği stratejisi dikkate alınarak elde edilmiştir. • Her bir adımda lokal hata, istenilen hata seviyesinden küçük gerçekleşmektedir. • Hesaplama işleminde adım sayısı, Algoritma 3.3.2 deki adım sayısından daha az ya da eşit sayıdadır.

3.3.5. Ek 2: Örnekler İçin Yazılan Prosedürler

Prosedürler Maple 12 programında yazılmıştır. Algoritmalarda kullanılan bazı sembollerin prosedürlerdeki karşılığı aşağıda verilmiştir.

δ_L	:	**dl**
h^*	:	**hstar**
β_k	:	**c[k]**
γ_k	:	**gama[k]**
ζ_k	:	**zeta [k]**
ξ_k	:	**xi[k]**
$\hat{h}_k$	:	**ph[k]**
$\widehat{h}_k$	:	**phh[k]**
$t_{k-1} + h_k$	:	**hSum**
$\|\|A\|\|$	:	**a** (Prosedür 3.5. ve 3.6. da)
$\max_{1\le i,j\le N} \{\| a_{ij} \|\}$	:	**a** (Prosedür 3.7.-3.8. ve 3.9. da)
$AX + \varphi(t, X)$	:	**F[k]**

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar)
global a, c, gama, zeta, ph, h, F, sbt, k, hSum, Y, t, Z, LE;
local sub1, sub2, v1, v2;
sub1:=proc()   c[k-1]:=max(b+abs(Y[k-1])); gama[k-1]:=T*(c[k-1])^2;
zeta[k-1]:=((c[k-1])^2)*(1-2*T+2*(T^2)*c[k-1]);
 ph[k]:=evalf(sqrt(2*dl/((a^2)*c[k-1]+a*gama[k-1]+zeta[k-1])));
hSum:=evalf(t[k-1]+ph[k]);
 if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0; end if:
 elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
 t[k]:=t[k-1]+h[k];     F[k-1]:=Vector([-Y[k-1][1]+t[k-1]*(Y[k-1][1])^2, 2*Y[k-1][2]]);
 Y[k]:=Vector([Y[k-1][1]+h[k]*F[k-1][1], Y[k-1][2]+h[k]*F[k-1][2]]);
 sbt[k]:=Vector([((1/Y[k-1][1])+t[k-1]-1)*exp(t[k-1]), Y[k-1][2]*exp(-2*t[k-1])]);
 Z[k]:=Vector([1/(1-t[k]+sbt[k][1]*exp(-t[k])), sbt[k][2]*exp(2*t[k])]);
 LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
 v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};    print(k,h[k],LE[k]);
 if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if: end proc:
 sub2:=proc()  multiple(plot,[v1],[v2]); end proc
 Y[0]:=X0;  Z[0]:=X0; t[0]:=t0; a:=Norm(A,Euclidean); k:=1; sub1();
 end proc:
```

Prosedür 3.5 Örnek 3.8. için prosedür

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar)
global a, c1,c2,c,gama, zeta, ph, h, F1,F2, F,sbt1,sbt2, k, hSum, Y1,Y2,Y, t, Z1,Z2,Z, LE;
local sub1, sub2, v1, v2;
sub1:=proc()  c1[k-1]:=5+abs(Y1[k-1]); c2[k-1]:=5+abs(Y2[k-1]); c[k-1]:=max(c1[k-1],c2[k-1]);
gama[k-1]:=2*cos(t[k-1])+3*(c1[k-1]^3);     zeta[k-1]:=2+90*(c1[k-1]^3);
ph[k]:=evalf(sqrt((2*dl)/((a^2)*c[k-1]+a*gama[k-1]+zeta[k-1])));   hSum:=evalf(t[k-1]+ph[k]);
if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0; end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
t[k]:=t[k-1]+h[k];
F1[k-1]:=100*Y1[k-1]; F2[k-1]:=25*Y1[k-1]+100*Y2[k-1]+2*cos(t[k-1])-3*(Y1[k-1]^3);
 F[k-1]:=Vector([F1[k-1],F2[k-1]]);
Y1[k]:=Y1[k-1]+h[k]*F1[k-1]; Y2[k]:=Y2[k-1]+h[k]*F2[k-1]; Y[k]:=Vector([Y1[k],Y2[k]]);
sbt1[k]:=Y1[k-1]*exp(-100*t[k-1]);
sbt2[k]:=(Y2[k-1]-25*t[k-1]*Y1[k-1]+(200/10001)*cos(t[k-1])-
                                    (2/10001)*sin(t[k-1])+(3/200)*(Y1[k-1]^3))*exp(-100*t[k-1]);
Z1[k]:=sbt1[k]*exp(100*t[k]);
Z2[k]:=sbt2[k]*exp(100*t[k])+25*t[k]*Z1[k]-(200/10001)*cos(t[k])+
                                                    (2/10001)*sin(t[k])-(3/200)*(Z1[k]^3);
Z[k]:=Vector([Z1[k],Z2[k]]);  LE[k]:=evalf(VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean));
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};   print(k,h[k],LE[k]);
if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if:
end proc:
sub2:=proc()  multiple(plot,[v2]); end proc:
Y[0]:=X0;Y1[0]:=1; Y2[0]:=1;  Z1[0]:=1;Z2[0]:=1; t[0]:=t0; a:=Norm(A,Euclidean); k:=1; sub1();
end proc:
```

Prosedür 3.6. Örnek 3.9. için prosedür

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar,N)
global a, c1,c2,c,gama, zeta, ph, h, F1,F2, F,sbt1,sbt2, k, hSum, Y1,Y2,Y, t, Z1,Z2,Z, LE;
local sub1, sub2, v1, v2;
sub1:=proc()
c1[k-1]:=b[1]+abs(Y1[k-1]); c2[k-1]:=b[2]+abs(Y2[k-1]); c[k-1]:=max(c1[k-1],c2[k-1]);
gama[k-1]:=2*cos(t[k-1])+3*(c1[k-1]^3);    zeta[k-1]:=2+90*(c1[k-1]^3);
ph[k]:=evalf(N^(-1/4)*(sqrt((2*dl)/((N^2)*(a^2)*c[k-1]+N*a*gama[k-1]+zeta[k-1]))));
hSum:=evalf(t[k-1]+ph[k]);
if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0; end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
t[k]:=t[k-1]+h[k];
 F1[k-1]:=100*Y1[k-1];  F2[k-1]:=25*Y1[k-1]+100*Y2[k-1]+2*cos(t[k-1])-3*(Y1[k-1]^3);
F[k-1]:=Vector([F1[k-1],F2[k-1]]);
Y1[k]:=Y1[k-1]+h[k]*F1[k-1];   Y2[k]:=Y2[k-1]+h[k]*F2[k-1];   Y[k]:=Vector([Y1[k],Y2[k]]);
sbt1[k]:=Y1[k-1]*exp(-100*t[k-1]);
sbt2[k]:=(Y2[k-1]-25*t[k-1]*Y1[k-1]+(200/10001)*cos(t[k-1])-
                                (2/10001)*sin(t[k-1])+(3/200)*(Y1[k-1]^3))*exp(-100*t[k-1]);
Z1[k]:=sbt1[k]*exp(100*t[k]);
Z2[k]:=sbt2[k]*exp(100*t[k])+25*t[k]*Z1[k]-(200/10001)*cos(t[k])+
                                                    (2/10001)*sin(t[k])-(3/200)*(Z1[k]^3);
Z[k]:=Vector([Z1[k],Z2[k]]);
LE[k]:=evalf(VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean));
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};    print(k,h[k],LE[k]);
if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if:
end proc:
sub2:=proc()  multiple(plot,[v2]); end proc:
Y[0]:=X0;Y1[0]:=1; Y2[0]:=1;  Z1[0]:=1;Z2[0]:=1; t[0]:=t0; a:=max(abs(A)); k:=1; sub1();
end proc:
```

Prosedür 3.7. Örnek 3.10 için prosedür

```
sssls:=proc(A::Matrix,b::Vector,dl,t0,T,hstar)
global a,c,c1,c2,gama, zeta, ph, h, F1,F2, k, hSum, Y1,Y2,Y, t, Z1,Z2,Z, LE,sbt1,sbt2;
local sub1, sub2, v1, v2;
sub1:=proc()
c1[k-1]:=5+abs(Y1[k-1]);   c2[k-1]:=5+abs(Y2[k-1]);   c[k-1]:=max(c1[k-1],c2[k-1]);
gama[k-1]:=T*(c2[k-1]^2);    zeta[k-1]:=(c2[k-1]^2)*(1+4*T);
ph[k]:=evalf((2^(-1/4))*(sqrt((2*dl)/(4*(a^2)*c[k-1]+2*a*gama[k-1]+zeta[k-1]))));
hSum:=evalf(t[k-1]+ph[k]);
if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0; end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
t[k]:=t[k-1]+h[k];
F1[k-1]:=-Y1[k-1]+Y2[k-1]+t[k-1]*(Y2[k-1])^2;   F2[k-1]:=2*Y2[k-1];
Y1[k]:=Y1[k-1]+h[k]*F1[k-1];    Y2[k]:=Y2[k-1]+h[k]*F2[k-1];
Y[k]:=Vector([Y1[k],Y2[k]]);
sbt2[k]:=Y2[k-1]*exp(-2*t[k-1]);
sbt1[k]:=(Y1[k-1]-((1/5)*t[k-1]-(1/25))*(Y2[k-1]^2)-(1/3)*Y2[k-1])*exp(t[k-1]);
Z2[k]:=sbt2[k]*exp(2*t[k]);
Z1[k]:=((1/5)*t[k]-(1/25))*(Z2[k]^2)+(1/3)*Z2[k]+sbt1[k]*exp(-t[k]);
Z[k]:=Vector([Z1[k],Z2[k]]);
LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)}; print(k,h[k],LE[k]);
if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if:
end proc:
sub2:=proc()  multiple(plot, [v2]); end proc:
Y1[0]:=1; Y2[0]:=1;  Z1[0]:=1;Z2[0]:=1; t[0]:=t0; a:=max(abs(A)); k:=1; sub1();
end proc:
```

Prosedür 3.8. Örnek 3.11 için Algoritma 3.3.2 ile yazılan prosedür

```
sssls:=proc(A::Matrix,X0::Vector,b::Vector,dl,t0,T,hstar)
global a,c,c1,c2,ph,phh,xi,gama, zeta,h,k,hSum,Y1,Y2,Y,F1,F2,t,Z1,Z2,Z,LE,sbt1,sbt2;
local sub1, sub2, v1, v2;
sub1:=proc()
c1[k-1]:=5+abs(Y1[k-1]);   c2[k-1]:=5+abs(Y2[k-1]);    c[k-1]:=max(c1[k-1],c2[k-1]);
phh[k]:=evalf((1/(a*(2^(5/4))))*sqrt(2*dl/c[k-1]));
gama[k-1]:=(t[k-1]+phh[k])*(c2[k-1]^2);    zeta[k-1]:=(c2[k-1]^2)*(1+4*(t[k-1]+phh[k]));
xi[k]:=(0.5*sqrt(2)*(phh[k]^2)*(4*(a^2)*c[k-1]+2*a*gama[k-1]+zeta[k-1]))/dl;
ph[k]:=evalf(phh[k]/sqrt(xi[k]));  hSum:=evalf(t[k-1]+ph[k]);
if (hSum<=T) then if (ph[k]>hstar) then h[k]:=ph[k]; else h[k]:=0; end if:
elif (T-t[k-1]>hstar) then h[k]:=T-t[k-1]; else h[k]:=0; end if:
t[k]:=t[k-1]+h[k];
F1[k-1]:=-Y1[k-1]+Y2[k-1]+t[k-1]*(Y2[k-1])^2;   F2[k-1]:=2*Y2[k-1];
Y1[k]:=Y1[k-1]+h[k]*F1[k-1];Y2[k]:=Y2[k-1]+h[k]*F2[k-1]; Y[k]:=Vector([Y1[k],Y2[k]]);
sbt2[k]:=Y2[k-1]*exp(-2*t[k-1]);
sbt1[k]:=(Y1[k-1]-((1/5)*t[k-1]-(1/25))*(Y2[k-1]^2)-(1/3)*Y2[k-1])*exp(t[k-1]);
Z2[k]:=sbt2[k]*exp(2*t[k]);  Z1[k]:=((1/5)*t[k]-
                                    (1/25))*(Z2[k]^2)+(1/3)*Z2[k]+sbt1[k]*exp(-t[k]);
Z[k]:=Vector([Z1[k],Z2[k]]); LE[k]:=VectorNorm(VectorAdd(Y[k],-Z[k]),Euclidean);
v1:={seq([n,h[n]],n=1..k)}; v2:={seq([n,LE[n]],n=1..k)};  print(k,t[k],h[k],LE[k]);
if t[k]=T or h[k]=0 then sub2(); else k:=k+1; sub1(); end if:
end proc:
sub2:=proc()  multiple(plot,[v2]); end proc:
Y[0]:=X0;Y1[0]:=1; Y2[0]:=1;  Z1[0]:=1;Z2[0]:=1; t[0]:=t0;a:=max(abs(A)); k:=1; sub1();
end proc:
```

Prosedür 3.9. Örnek 3.11 için Algoritma 3.3.3 ile yazılan prosedür

4. KAYNAKLAR

A. García, P. Martín, A. B. González (2004), *High order Runge- Kutta- Nyström codes fort he integration of oscillatory prroblems,* Applied Numerical Mathematics, 48: 13-25.

A. O. Çıbıkdiken (2008), *Kayan Nokta Aritmetiğinin Periyodik Lineer Fark Denklem Sisteminin Fundamental Matrisinin Hesaplanmasına Etkisi*, Doktora Tezi, S. Ü. Fen Bilimleri Enstitüsü, Konya.

C. W. Gear (1971), *Numerical Initial Value Problems In Ordinary Differential Equations*, Prentice- Hall, New Jersey.

C. F. V. Loan (2000), *Introduction to scientific Computing,* Prentice- Hall, London.

C. Arévalo, G. Söderlind, J. D. López (2007), *Constant coefficient linear multistep methods with step density control*, Journal of Computational and Applied Mathematics, 205: 891- 900, 2007.

D. Estep (2002), *Practical Analysis in One Variable*, Springer-Verlag, New York.

E. Hairer, G. Wanner (1991), *Solving Ordinary Differential Equations II, Stiff and Differential- Algebraic Problems,* Springer- Verlag, New York.

G. H. Golub, J. M. Ortega (1992), *Scientific Computing and DifferentialEquations*, Academic Pres Limited, London.

G. Çelik Kızılkan (2004), *Başlangıç Değer Problemlerinin Nümerik İntegrasyonunda Adım Genişliği Tespiti*, Yüksek Lisans Tezi, S. Ü. Fen Bilimleri Enstitüsü, Konya.

G. Çelik Kızılkan, K. Aydın (2005), *Hata analizi tabanlı adım genişliği stratejisi*, Selçuk Üniversitesi Fen Edebiyat Fakültesi Fen Dergisi, 25:79-86.

G. Çelik Kızılkan, K. Aydın (2006), *A new variable step size algorithm for Cauchy problem,* Applied Mathematics and Computation, 183:878-884.

G. Dahlquist, A. Björck (2008), *Numerical Methods in Scientific Computing,* Volume I, SIAM Press, Philadelphia.

H. Ramos, J. Vigo-Aguiar (2005), *Variable stepsize Störmer- Cowell methods,* Mathematical and Computer Modelling, 42: 837- 846.

H. Ramos, J. Vigo-Aguiar (2007), *Variable stepsize Chebyshev- type methods fort he integration of second order I.V.P.' s,* Journal of Computational and Applied Mathematics, 204:102-113.

J. C. Butcher, H. Podhaisky (2006), *On error estimation in general linear methods for stiff ODEs*, Applied Numerical Mathematics, 56:345- 357.

K. Aydın, A. Bulgak, H. Bulgak, N. A. Chumakova (2001), *Global Error Estimation in Numerical Integration of Ordinary Differential Equations,* Report No.4/2001, Konya.

K. Aydın, , A. Bulgak, H. Bulgak (2003), *Bilgisayarla Matematik Analiz*, Selçuk Üniversitesi Yayınları, Konya.

L. F. Shampine, R. C. Allen, S. Pruess (1996), *Fundamentals of Numerical Computing*, John Wiley&Sons, INC, New York.

L. F. Shampine (2005), *Error estimation and control for ODEs*, Journal Of Scientific Computing, Volume 25, Numbers 1-2 ,3-16.

L. Lara (2007), *One- step recursive method for solving systems of ordinary differential equations,* Journal of Computational and Applied Mathematics, 216: 210-216.

M. R. M. Rao (1979), *Ordinary Differential Equations Theory and Applications,* Indian Institue of Tecnology, Kanpur.

Ö. Akın, H. Bulgak (1998), *Lineer Fark Denklemleri ve Kararlılık Teorisi*, S Selçuk Üniversitesi Yayınları, Konya.

R. Holsapple, R. Iyer, D. Doman (2007), *Variable step- size selection methods for implicit integration schemes for ODEs,* International Journal of Numerical Analysis and Modelling, Volume 4, Number 2, 210- 240.

W. L. Miranker (1981), *Numerical Methods for Stiff Equations and Singular Perturbation Problems*, D. Rediel Publishing Company, Holland.

W. A. Brock, A. G. Malliaris (1989), *Diferential Equations,Stability and Chaos in Dynamic Economics*, Elsevier Science Publishers, Amsterdam.