İNCE CİDARLI KOMPOZİT KİRİŞLERİN STATİK VE DİNAMİK ANALİZİ

M. Gökhan GÜNAY

YÜKSEK LİSANS TEZİ

MAKİNE MÜHENDİSLİĞİ
ANA BİLİM DALI

Prof. Dr. Taner TIMARCI

EDİRNE

2013

TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

T.Ü. Fen Bilimleri Enstitüsü onayı.

Prof. Dr. Mustafa ÖZCAN
Fen Bilimleri Enstitüsü Müdürü

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

Prof. Dr. Taner TIMARCI
Anabilim Dalı Başkanı

Bu tez tarafımca okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans/ Doktora tezi olarak kabul edilmiştir.

Prof. Dr. Taner TIMARCI
Tez Danışmanı

Bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Mekanik Anabilim Dalında bir Yüksek lisans tezi olarak oy birliği ile kabul edilmiştir.
Jüri Üyeleri

Prof. Dr. Taner TIMARCI

Yrd.Doç.Dr. Nusret MEYDANLIK

Yrd.Doç.Dr. Oğuzhan ERDEM

Tarih: ...../……/……

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
MAKİNA MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

DOĞRULUK BEYANI

İlgili tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin kaynak gösterilerek ilgili tezde yer aldığını beyan ederim.

Yüksek Lisans Tezi
İnce Cidarlı Kompozit Kirişlerin Statik ve Dinamik Analizi
T.Ü. Fen Bilimleri Enstitüsü
Makine Mühendisliği Ana Bilim Dalı
M. Gökhan GÜNAY

Özet

Bu çalışmada açık veya kapalı herhangi bir kesite sahip ince cidarlı kompozit kirişlerin çeşitli yükleme koşulları altında statik ve dinamik analizini yapabilen bir sonlu elemanlar yazılımı geliştirilmiştir. Hesaplamalarda kullanılan analitik model kompozit malzemelerde oluşabilecek uzama-kayma, uzama-eğilme, eğilme-burulma bağlılıklarını ve çarpılma davranışını dikkate almaktadır. Geliştirilen özel bir algoritma ile tek hücreli herhangi bir kesite sahip kirişin mekanik özellikleri hesaplanmaktadır. Analitik modelin çözümü yer değiştirme temelli sonlu elemanlar yöntemi kullanılarak yapılmıştır. Geliştirilen yazılım ile çeşitli kesitlere sahip ince cidarlı kompozit kirişler incelenmiş, bulunan sonuçlar daha önce yapılan çalışmalarla ve çeşitli sonlu elemanlar yazılımları sonuçları ile karşılaştırılmıştır.

Yıl…… :2013

Sayfa Sayısı……. :147

Anahtar Kelimeler….. :İnce cidarlı kompozit kirişler, Elyaf takviyeli tabakalı kompozit malzemeler, Eğilme-burulma davranışı, Kirişlerin titreşimi, Sonlu elemanlar yöntemi

Master Thesis
Static and Dynamic Analysis of Thin Walled Composite Beams
Trakya University Institute of Natural Sciences
Mechanical Engineering Department
M. Gökhan Günay

Abstract

At this study a finite element software is developed for static and dynamic analyses of arbitrary open or closed cross-section thin walled composite beams under different loading conditions. The analytical model takes into account extension-shear, extension-bending, bending-twist couplings which can occur in composite materials and warping behavior. With a algorithm developed the mechanical properties of any beam which has single cell arbitrary cross-section can be calculated. Analytical is model solved with displacement based finite element method. Thin walled composite beams having several cross-sections are investigated with the software developed, the results obtained are compared with previous studies and the results of other finite element software.

Year……..:2013

Number of Pages……..:147

Keywords……...:Thin walled composite beams, Fiber-reinforced laminated composite materials, Flexural-torsional behavior, Vibration of beams, Finite element method

TEŞEKKÜRLER

Bu eserin ortaya çıkmasında emeğini ve desteğini benden hiç esirgemeyen sayın hocam Prof. Dr. Taner TIMARCI`ya, değerli hocalarıma, sevgili aileme ve arkadaşlarıma şükranlarımı sunarım.

M. Gökhan GÜNAY

İÇİNDEKİLER

Tez Bildirimi…… ii
Doğruluk Beyanı……..iii
Özet……... iv
Abstract…… v
Teşekkürler…….. vi
Semboller ve Kısaltmalar……ix
Şekiller Listesi……..xii
Tablolar Listesi…… xiv
1. GİRİŞ……... **1**
1.1. Amaç ve Kapsam……...1
1.2. Daha Önce Yapılan Çalışmalar…… 2
2. SÜREKLİ ELYAF TAKVİYELİ KOMPOZİT MALZEMELER…… **4**
2.1. Kompozit Malzemeler Hakkında Genel Bilgi……... 4
2.2. Sürekli Elyaf Takviyeli Kompozit Malzemeler…… 5
2.3. Laminanın Mikromekanik Özellikleri……... 6
2.4. Laminatın Makromekanik Özellikleri……... 7
2.5. Laminatın Makromekanik Davranışı……15
3. İNCE CİDARLI KOMPOZİT KİRİŞLER…….. **24**
3.1. İnce Cidarlı Kompozit Kirişlerin Kinematiği…… 24
3.2. Yer Değiştirme Alanı……28
3.3. Genleme Alanı……... 31
3.4. Varyasyonel Formülasyon……33
3.5. Bünye Denklemleri……35
3.6. Hareket Denklemleri……..37
3.7. Sonlu Elemanlar Formülasyonu…… 38
4. GELİŞTİRİLEN SONLU ELEMANLAR YAZILIMI……... **42**
4.1. Yazılım Hakkında Genel Bilgi…….. 42
4.2. Yazılımın Akış Şeması ve Çalışma Prensibi…… 43
5. ANALİZLER…….. **50**
5.1. Doğrulama ve Karşılaştırmalar……..50
5.2. Çeşitli Kesitlere Sahip Kirişlerin Analizleri ve Karşılaştırmalar…….. 57
5.3. Doğal Frekansların Elyaf Açılarına Bağlı Olarak Değişimi ……..... 59

6. SONUÇ VE YORUM..... **62**
KAYNAKLAR..... **64**
Ek-A Kesit Boyutları..... **67**
Ek-B Yazılımın Akış Şeması..... **80**
Ek-C Yazılımın Matlab Kodları..... **143**
Ek-D Örnek Kesit ve DXF Dosyası..... **146**

SEMBOLLER VE KISALTMALAR

V_f	Elyafın hacimsel oranı
V_m	Matrisin hacimsel oranı
E_f	Elyafın elastiklik modülü
E_m	Matrisin elastiklik modülü
G_f	Elyafın kayma modülü
G_m	Matrisin kayma modülü
E_1	Laminanın 1 yönündeki elastiklik modülü
E_2	Laminanın 2 yönündeki elastiklik modülü
v_{12}	Laminanın 1-2 düzlemindeki poisyon oranı
G_{12}	Laminanın 1-2 düzlemindeki kayma modülü
σ	Gerilme
γ	Kayma gerilmesi
ε	Genleme
τ	Kayma genlemesi
C_{ij}	Sertlik matrisi
S_{ij}	Uygunluk matrisi
$\mathbf{Q_{ij}}$	İndirgenmiş sertlik matrisi
$\mathbf{Q_{ij}}$	Döndürülmüş indirgenmiş sertlik matrisi
T	Dönüşüm matrisi
R	Reuter matrisi
N_z	Eksenel kuvvet
M_x	X eksenindeki eğilme momenti
M_y	Y eksenindeki eğilme momenti
M_ω	Çarpılma momenti (Bimoment)

M_t	Burulma momenti
$( )'$	z eksenine göre türev
L	Kiriş uzunluğu
A_{ij}	A matrisi
B_{ij}	B matrisi
D_{ij}	D matrisi
(x,y,z)	Genel koordinat sistemi notasyonu
(n,s,z)	Lokal koordinat sistemi notasyonu
e_n	Lokal koordinat sistemi n doğrultusundaki birim vektör
e_t	Lokal koordinat sistemi s doğrultusundaki birim vektör
$P\ x,y,z$	Kutup noktası
$m\ x,y,z$	Orta yüzeydeki herhangi bir nokta
$M\ x,y,z$	Kiriş kesitindeki herhangi bir nokta
p, □	Herhangi bir noktanın konum vektörü
q, Q	Herhangi bir noktanın konum vektörünün s bileşeni
r, R	Herhangi bir noktanın konum vektörünün n bileşeni
U	Kutup noktasının x yönündeki yer değiştirmesi
V	Kutup noktasının y yönündeki yer değiştirmesi
W	Kutup noktasının z yönündeki yer değiştirmesi
ϕ	Kutup noktasının z ekseni etrafındaki dönmesi yer değiştirmesi
u, v, w	Orta yüzeydeki herhangi bir noktanın x,y,z yönündeki yer değiştirmesi
u_M, v_M, w_M	Kiriş kesitindeki herhangi bir noktanın x,y,z yönündeki yer değiştirmesi
γ_{sz}	Kiriş kesitinde sz yüzeyinde oluşan kayma gerilmesi
$F\ s$	St. Venant kayma akımı akışı
$h(s)$	Cidar kalınlığı
$\omega\ s$	Çarpılma fonksiyonu

$\boldsymbol{\varepsilon_z^0}$	Eksenel genleme
$\boldsymbol{\kappa_x}$	X yönündeki bieksenel eğrilik
$\boldsymbol{\kappa_y}$	Y yönündeki bieksenel eğrilik
$\boldsymbol{\kappa_\omega}$	Çarpılma eğriliği
$\boldsymbol{\kappa_{sz}}$	Burulma eğriliği
$\boldsymbol{\mathcal{W}}$	Genleme enerjisi
$\boldsymbol{\mathcal{T}}$	Kinetik enerji
$\boldsymbol{\rho}$	Kompozit malzemenin yoğunluğu
$\boldsymbol{E_{ij}}$	Sertlik matrisi
$\boldsymbol{K_{ij}}$	Eleman sertlik matrisi
$\boldsymbol{M_{ij}}$	Eleman kütle matrisi
Δ	Yer değiştirme vektörü
$\boldsymbol{f}$	Kuvvet vektörü
$\boldsymbol{\varphi}$	Lagrange interpolasyon fonksiyonu
$\boldsymbol{\psi}$	Hermetik-kübik interpolasyon fonksiyonu
$\boldsymbol{Wn}$	Sistemin doğal frekansı
CPP	Composite Pre Post

ŞEKİLLER LİSTESİ

Şekil 2.1 Matris ve Takviye Elemanı

Şekil 2.2 Lamina

Şekil 2.3 Laminat

Şekil 2.4 Laminanın Mikromekanik Özellikleri

Şekil 2.5 Birim Eleman Üzerindeki Gerilmeler

Şekil 2.6 Döndürülmüş Koordinatlar

Şekil 2.7 Döndürülmüş Lamina

Şekil 2.8 Laminada Yer Değiştirmeler

Şekil 2.9 Laminada Gerilme ve Genlemeler

Şekil 2.10 Laminada Kuvvetler

Şekil 2.11 Laminada Momentler

Şekil 2.12 Katman Kalınlıkları

Şekil 2.13 Simetrik Laminat

Şekil 2.14 Simetrik Açılı Laminat

Şekil 2.15 Antisimetrik Laminat

Şekil 2.16 Dik Katmanlı Laminat

Şekil 3.1 Kiriş

Şekil 3.2 İnce Cidarlı Kiriş

Şekil 3.3 İnce Cidarlı Kiriş Tipleri

Şekil 3.4 Genel ve Lokal Koordinat Sistemleri

Şekil 3.5 İnce Cidarlı Kiriş Kesiti

Şekil 3.6 Çarpılma (Warping)

Şekil 3.7 Yer Değiştirmeler

Şekil 3.8 Herhangi Bir m Noktasının Tanımlanması

Şekil 3.9 Eğrilikler

Şekil 3.10 Lagrange İnterpolasyon Fonksiyonu

Şekil 3.11 Hermetik-Kübik İnterpolasyon Fonksiyonu

Şekil 4.1 Akış Şeması

Şekil 4.2 Kullanıcı Arayüzü (Ana)

Şekil 4.3 Kullanıcı Arayüzü (Ayarlar)

Şekil 4.4 Kesitten Dataların Toplanması

Şekil 4.5 Doğru Parçası

Şekil 4.6 Dataların İşlenmesi

Şekil 4.7 Statik Deformasyon Çıktı Örneği

Şekil 4.8 Serbest Titreşim Çıktı Örneği

Şekil 5.1 Eksenel Yük Altındaki Kutu Kiriş

Şekil 5.2 Moment Etkisindeki Kutu Kiriş

Şekil 5.3 Eksantrik Yayılı Yük Altındaki Kutu Kiriş

Şekil 5.4 Bir Ucu Ankastre Sabitlenmiş Kutu Kiriş

Şekil 5.5 Katman Dizilimi [0/45]3 Olan Kutu Kirişin Doğal Frekansları ve Mod Şekilleri

Şekil 5.6 Çeşitli Yükler Altındaki I Kiriş

Şekil 5.7 Yayılı Yükler Altındaki Rüzgar Türbin Kanadı

Şekil 5.8 Kompozit Bisiklet Çerçevesi Alt Kirişi

Şekil 5.9 Kutu Kiriş

Şekil 5.10 Kutu Kiriş İçin Elyaf Açısına Bağlı Olarak Wn-v1`in Değişimi

Şekil 5.11 Kutu Kiriş İçin Elyaf Açısına Bağlı Olarak Wn- ϕ1`in Değişimi

Şekil 5.12 Özel Kiriş

Şekil 5.13 Özel Kiriş İçin Elyaf Açısına Bağlı Olarak Wn-v1`in Değişimi

Şekil 5.14 Özel Kiriş İçin Elyaf Açısına Bağlı Olarak Wn- ϕ1`in Değişimi

TABLOLAR LİSTESİ

Tablo 1 Malzeme Özellikleri

Tablo 2 Eksenel Yerdeğiştirme

Tablo 3 Düşey Yerdeğiştirme (100Nm Moment)

Tablo 4 Düşey Yerdeğiştirme (6,5kN/m Yayılı Yük)

Tablo 5 Doğal Frekanslar (Serbest Titreşim)

Tablo 6 I-Kiriş Yerdeğiştirmeler

Tablo 7 Türbin Kanadı Statik Analiz

Tablo 8 Türbin Kanadı Doğal Frekanslar (Serbest Titreşim)

Tablo 9 Bisiklet Kirişi Statik Analiz

Tablo 10 Bisiklet Kirişi Doğal Frekanslar (Serbest Titreşim)

BÖLÜM 1

GİRİŞ

1.1. Amaç ve Kapsam

Kompozit malzemeler yüksek dayanım ve düşük ağırlık özellikleriyle ön plana çıkmıştır. Son 30 yıldır birçok uygulamada kompozit malzemeler kullanılmış ve kullanım alanları giderek artmaktadır. Günümüzde çok çeşitli özellik ve yapıda kompozit malzemeler mevcuttur. İnce cidarlı kompozit elemanlar havacılık, enerji, inşaat, deniz ve otomotiv sanayi olmak üzere birçok alanda taşıyıcı veya fonksiyonel yapı elemanı olarak kullanılmaktadır ve bu elemanların büyük bir kısmı kiriş formundadır.

Daha önce yapılan çalışmalarda dikdörtgen, daire, I ve U gibi basit bir kesite sahip kompozit kirişler incelenmiş ve bu kesitlere ait çeşitli çözümler, çözüm yöntemleri verilmiştir. Günümüzdeki uygulamalarda ise giderek daha karmaşık kesitlere sahip kirişlere ihtiyaç duyulmaktadır. Herhangi bir kesite sahip ince cidarlı kompozit kirişler*, ince cidarlı kompozit kirişler teorisi* ile modellenebilir fakat çözümlerin elde edilmesi zor ve zaman alıcıdır. Kiriş kesiti karmaşıklaştıkça sonuçlara ulaşmak daha da zorlaşmaktadır. Benzer şekilde bir uygulamada kiriş kesitinin, yükleme koşullarının değişmesi veya optimize edilmesi istendiğinde ise modelleme ve çözümlemelerin tekrar tekrar yapılması gerekmektedir. Bu ve benzeri sebeplerden dolayı bu çalışmada günümüz uygulamalarında verimli bir şekilde kullanılabilecek açık veya kapalı herhangi bir kesite sahip ince cidarlı kompozit kirişlerin çeşitli yükleme koşulları altında statik ve dinamik analizini yapabilecek bir yazılım geliştirmek hedeflenmiştir.

Bu çalışma yapılırken ve yazılım geliştirilirken daha önce yapılmış çalışmalar ve mevcut ticari yazılımlar incelenmiştir. Hesaplamalarda kullanılan denklemler daha önce yapılan çalışmalardan alınmış veya esinlenilmiştir.

İnce cidarlı kompozit kirişler genel olarak katmanlı sürekli elyaf takviyeli kompozit malzemeler kullanılarak üretilirler. Bölüm-2`de sürekli elyaf takviyeli kompozit malzemelerin mekanik özellikleri ve katmanlı kompozit malzemelerin mekanik davranışı hakkında bilgi verilmiştir. Bölüm-3`de ince cidarlı kompozit kirişler teorisi açıklanmış ve sonlu elemanlar çözüm yöntemi verilmiştir. Bölüm-4`de açık veya kapalı herhangi bir kesite sahip ince cidarlı kompozit kirişlerin çeşitli yükleme koşulları altında statik ve dinamik analizini yapabilmek için geliştirilen yazılım hakkında bilgi verilmişitr. Bölüm-5`de yapılan çalışmanın doğruluğunu sınamak için bulunan sonuçlar, bu konuda yapılmış diğer çalışmalarla, Ansys ve Patran sonuçlarıyla karşılaştırılmıştır. Bölüm-6`da çalışmanın sonuçları özetlenmiş ve yazılım ile ilgili ileride yapılabilecek çalışmalar açıklanmıştır. EK-A`da analizlerde kullanılan kullanılan kiriş kesitleri, EK-B`de geliştirilen yazılımın akış şeması, EK-C`de geliştirilen yazılımın Matlab kodları, EK-D`de ise yazılıma kiriş kesitini tanıtmada kullanılan örnek bir DXF dosyasının ilgili içeriği verilmiştir.

1.2. Daha Önce Yapılan Çalışmalar

İnce cidarlı kapalı kesitli izotropik elemanların teorisi ilk kez Vlasov [1] ve Gjelsvik [2] tarafından geliştirilmiştir. Fiber takviyeli kutu kirişler için değişik kabuller altında çeşitli çalışmalar yapılmıştır. Chandra ve ark. [3] eğilme burulma ve çekme yükleri altındaki simetrik ve antisimetrik kutu kirişler için yapısal bağlılıkları incelemiştir. Buldukları teorik sonuçları sonlu elemanlar analizi sonuçlarıyla ve kendi yaptıkları deneylerle kıyaslamışlardır. Song ve Librescu [4] herhangi bir kesite sahip ince veya kalın cidarlı kompozit kirişlerin dinamik analizi üzerine yoğunlaşmıştır. Yaptıkları bu çalışmada dönen ve spin yapan (rotating and spinning) ince cidarlı kutu kirişlere geniş bir yer verilmiştir. Ayrıca aynı çalışmada herhangi bir kesite sahip çok hücreli kirişler için bir çözüm yöntemi de verilmiştir. Puspita ve ark. [5] ortotropik kompozit kirişler için basitleştirilmiş bir analitik hesaplama yöntemi yayınlamış ve helikopter rotoru gibi çeşitli şekillerdeki kompozit kirişleri incelemek için bir sonlu elemanlar programı hazırlamıştır. Yayınlanan bu analitik yöntem basit kesitlere sahip kirişler için sağlıklı sonuçlar vermiştir. Jeon ve ark. [6] kirişlerde büyük yerdeğiştirmeler teorisini kullanarak kompozit kutu kirişlerde statik ve dinamik analiz için bir analitik model geliştirmiştir. Yaptıkları bu çalışmada büyük yer değiştirmeler altındaki sonuçlar bulunsa da hesaplamalarda oluşan küçük genlemeler de dikkate

alınmıştır. Kollar ve Pluzsik [7] ince cidarlı açık veya kapalı kesite sahip kompozit kirişler için bir kiriş teorisi yayınlamıştır. Bu çalışmada çarpılma ve yanal kaymalar ihmal edilmiştir. Salim ve Davalos [8] Gjelsvic [2]`in modelini geliştirerek açık veya kapalı kesite sahip kompozit kirişler için oluşabilecek bütün elastik bağlılıkları içeren lineer bir model hazırlamışlardır. Cortinez ve Piovan [9] Hellinger-Ressner prensibini kullanarak açık veya kapalı kesite sahip ince cidarlı kompozit kirişlerin kararlılık analizini yapmak için bir teorik model geliştirmiştir. Lee ve Lee [10] ince cidarlı açık kesite sahip I kirişlerin eğilme burulma analizini yapabilen analitik bir model geliştirmiştir. Geliştirilen bu modelde eğilme-burulma bağlılığı dikkate alınmıştır. Vo ve Lee [11] Lee ve Lee`nin [10] modelini geliştirerek kapalı kesite sahip ince cidarlı kompozit kutu kirişlerin eğilme burulma davranışını incelemiştir. Vo ve Lee [12] Lee ve Lee`nin [10] modelinden yola çıkarak kapalı kesite sahip ince cidarlı kompozit kutu kirişlerin serbest titreşim davranışını incelemiştir.

Bu çalışmada açık veya kapalı herhangi bir kesite sahip ince cidarlı kompozit kirişlerin çeşitli yükleme koşulları altında statik ve dinamik analizini yapabilen bir sonlu elemanlar yazılımı geliştirilmiştir. Kullanılan analitik model Vo ve Lee`nin [11,12] çalışmalarından esinlenerek ve klasik laminasyon teorisi esas alınarak hazırlanmıştır. Hesaplamalarda kompozit malzemelerde oluşabilecek tüm elastik bağlılıklar dikkate alınmıştır. Geliştirilen özel bir algoritma ile kiriş kesiti okunmakta ve okunan kesite göre kirişin mekanik özellikleri hesaplanmaktadır. Analitik modelin çözümü yer değiştirme temelli sonlu elemanlar yöntemi kullanılarak yapılmıştır. Geliştirilen yazılım ile çeşitli kesitlere sahip ince cidarlı kompozit kirişler incelenmiş, bulunan sonuçlar daha önce yapılan çalışmalarla ve çeşitli sonlu elemanlar yazılımları sonuçları ile karşılaştırılmıştır.

BÖLÜM 2

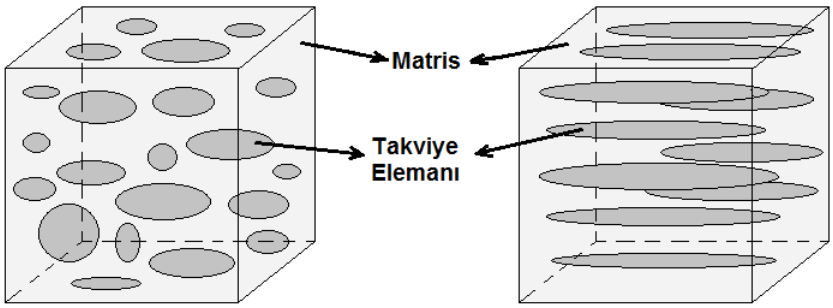
SÜREKLİ ELYAF TAKVİYELİ KOMPOZİT MALZEMELER

Bu bölümde kompozit malzemeler hakkında genel bilgiler ve sürekli elyaf takviyeli kompozit malzemelerin mekanik özellikleri hakkında teorik bilgi verilmiştir**.**

2.1. Kompozit Malzemler Hakkında Genel Bilgi

İki veya daha fazla sayıdaki malzemenin, makro seviyede birleştirilmesiyle oluşturulan yeni malzemeye *kompozit malzeme* adı verilir. Kompozit malzemeler bileşenlerinin iyi özelliklerini taşıyacak ve yeni özelliklere sahip olacak şekilde kullanım amacına yönelik olarak tasarlanabilirler.

Kompozit malzemeyi oluşturan malzemeler, matris ve takviye elemanı olmak üzere iki gruba ayrılır;



Şekil 2.1 Matris ve Takviye Elemanı

Matris:

Kompozit malzemede kullanılan takviye elamanlarını bir arada tutarak malzemenin bütünlüğünü sağlayan yapıya *matris* adı verilir. Matris takviye elemanına göre çok fazla yük taşımaz fakat malzemenin genel şeklini verir. Matris malzemesi olarak reçineler, metaller ve seramikler gibi çeşitli malzemeler kullanılabilir.

Takviye Elemanı:

Kompozit malzemede yükün büyük kısmını taşıyan yapılara *takviye elemanı* adı verilir. Takviye elemanlarının şekilleri, dağılımları ve yönelimleri kompozit malzemenin mekanik özelliklerini büyük ölçüde etkiler. Takviye elemanı olarak cam, karbon, boron, alüminyum oksit, silisyum karbür ve organik moleküller (aramidler) gibi çeşitli yüksek mukavemetli malzemeler kullanılabilir.

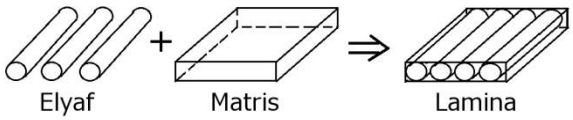
2.2. Sürekli Elyaf Takviyeli Kompozit Malzemeler

İnce cidarlı kompozit kirişler genellikle sürekli elyaf takviyeli kompozit malzeme kullanılarak imal edilirler. Bu yüzden bu çalışmada sürekli elyaf takviyeli kompozit malzemelerin mekanik özellikleri incelenmiş ve kiriş denklemleri bu özelliklere göre çıkarılmıştır.

Sürekli elyaf takviyeli kompozit malzemelerde takviye elemanı olarak elyaf kullanılır, elyaflar malzeme boyunca kesintisiz bir şekilde uzanır. Sürekli elyaf takviyeli kompozit malzemeler lamina veya laminattan oluşur.

Lamina:

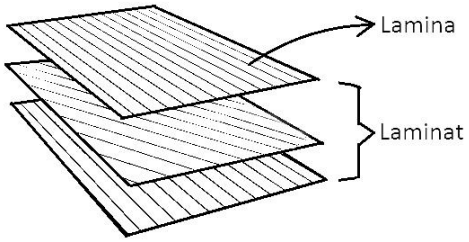
Şekil 2.2 deki gibi tek yönde uzanan elyafların matris malzemesi ile bir araya getirilmesi ile oluşturulan yapıya *lamina* adı verilir. Lamina makroskopik olarak homojen kabul edilir; fakat mikroskopik olarak heterojendir.



Şekil 2.2 Lamina

Laminat:

Şekil 2.3 deki gibi iki veya daha fazla laminanın bir birine katmanlar halinde yapıştırılması ile oluşturulan yapıya *laminat* adı verilir.



Şekil 2.3 Laminat

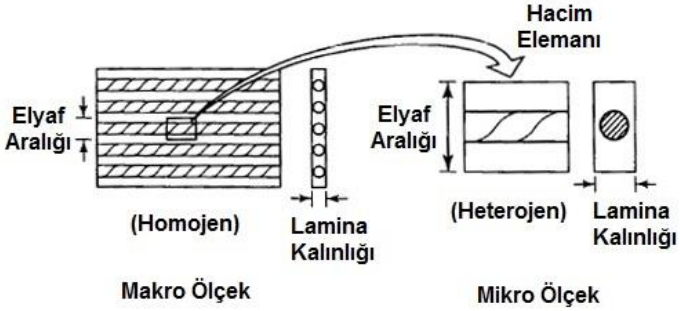
Kompozit malzemeler birden fazla malzemenin bir araya gelmesiyle oluşur, bu yüzden çoğu kompozit malzeme heterojen ve anizotropitir. Kompozit malzemeler sahip oldukları bu karmaşık yapıdan dolayı, *mikromekanik* ve *makromekanik* olmak üzere iki farklı açıdan incelenir [13];

Mikromekanik çalışmada, kompozit malzemeyi oluşturan malzemelerin bir biri ile etkileşimi araştırılır. Bu etkileşimlere göre kompozit malzemenin mekanik özellikleri, kendini oluşturan malzemelerin mekanik özellikleri cinsinden tanımlanır.

Makromekanik çalışmada ise, mikro mekanik çalışmada tanımlanan kompozit malzemenin mekanik özellikleri kullanılarak, malzemenin makro boyutta kuvvet ve momentlere nasıl bir tepki vereceği araştırılır.

2.3. Laminanın Mikromekanik Özellikleri

Laminanın mekanik özelliklerini belirlemek için Şekil 2.4 deki gibi laminanın mikroskopik davranışını temsil eden bir hacim elemanı seçilir ve hesaplamalar bu hacim elemanı üzerinden yapılır [13].

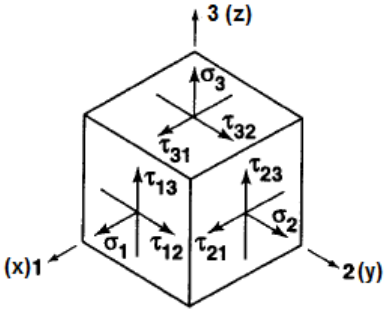


Şekil 2.4 Laminanın Mikromekanik Özellikleri

2.4. Laminatın Makromekanik Özellikleri

Bir malzemedeki gerilme genleme ilişkisi Genelleştirilmiş Hooke yasası ile aşağıdaki gibi tanımlanır [13]:

$$\sigma_i = C_{ij}.\varepsilon_j \qquad i,j = 1,\dots,6 \tag{2.1}$$



Şekil 2.5 Birim Eleman Üzerindeki Gerilmeler

Burada $\boldsymbol{\sigma_i}$ gerilme bileşenlerini, $\boldsymbol{C_{ij}}$ sertlik matrisini , $\boldsymbol{\varepsilon_j}$ ise genleme bileşenlerini temsil eder. Şekil 3.4 da birim eleman üzerindeki gerilmeler gösterilmiştir. **x,y,z** eksenler ve **u,v,w** yerdeğiştirmeler olmak üzere, gerilmeler aşağıdaki gibi tanımlanır:

$$\varepsilon_1 = \frac{\partial u}{\partial x}, \varepsilon_2 = \frac{\partial v}{\partial y}, \varepsilon_3 = \frac{\partial w}{\partial z}$$

$$\gamma_{12} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, \gamma_{23} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \gamma_{13} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \tag{2.2}$$

Denklem 2.1 deki $\boldsymbol{C_{ij}}$ sertlik matrisi 36 adet bağımsız elemana sahiptir. Elastik bir malzemede birim eleman için genleme enerjisi;

$$dW = \sigma_i.d\varepsilon_j \tag{2.3}$$

şeklinde tanımlanır. Denklem 2.1 kullanılırsa bu ifade;

$$dW = C_{ij}.\varepsilon_j.d\varepsilon_j \tag{2.4}$$

şekline dönüşür. Bu ifade üzerinden integral alınırsa hacim için genleme enerjisi;

$$W = \frac{1}{2} C_{ij}.\varepsilon_i.\varepsilon_j$$

(2.5)

şeklinde tanımlanır. Bu ifade;

$$\frac{\partial W}{\partial \varepsilon_i} = C_{ij}.\varepsilon_j$$

(2.6)

şeklinde tekrar düzenlenerek Genelleştirilmiş Hooke yasası elde edilebilir. Bu ifadede $\boldsymbol{C_{ij}}$ yalnız bırakılırsa ifade;

$$\frac{\partial^2 W}{\partial \varepsilon_i.\partial \varepsilon_j} = C_{ij}$$

(2.7)

şekline dönüşür. Aynı yolla $\boldsymbol{C_{ji}}$ aşağıdaki şekilde tanımlanabilir;

$$\frac{\partial^2 W}{\partial \varepsilon_j.\partial \varepsilon_i} = C_{ji}$$

(2.8)

(2.7) ve (2.8) nolu ifadelerde diferansiyellerin sırası önemli değildir. Buna göre;

$$C_{ij} = C_{ji}$$

(2.9)

eşitliği ortaya çıkar. Bu durumda $\boldsymbol{C}$ sertlik matrisi simetrik olur ve bir birinden bağımsız eleman sayısı 21 e düşer.

Benzer şekilde $\boldsymbol{S}$ uygunluk matrisi aşağıdaki gibi tanımlanır;

$$\varepsilon_i = S_{ij}.\sigma_j \qquad i,j = 1,\dots,6$$

(2.10)

$\boldsymbol{S}$ uygunluk matrisi için genleme enerjisi;

$$W = \frac{1}{2} S_{ij}.\sigma_i.\sigma_j$$

(2.11)

şeklinde tanımlanır ve buradan;

$$S_{ij} = S_{ji}$$

(2.12)

eşitliği gösterilebilir. Bu ifadeler ışığında lineer elastik anizotropik bir malzeme;

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \tau_{23}\\ \tau_{31}\\ \tau_{12}\end{bmatrix}=\begin{bmatrix}c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16}\\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26}\\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36}\\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46}\\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56}\\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66}\end{bmatrix}.\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \gamma_{23}\\ \gamma_{31}\\ \gamma_{12}\end{bmatrix}$$

$$[S_{ij}]=[C_{ij}]^{-1}$$

(2.13)

şeklinde tanımlanır. Böyle bir malzemede bir adet simetri ekseni olursa, bir birinden bağımsız eleman sayısı 13 e düşer. Bu tip malzemelere *monoklinik malzeme* adı verilir.

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \tau_{23}\\ \tau_{31}\\ \tau_{12}\end{bmatrix}=\begin{bmatrix}c_{11} & c_{12} & c_{13} & 0 & 0 & c_{16}\\ c_{12} & c_{22} & c_{23} & 0 & 0 & c_{26}\\ c_{13} & c_{23} & c_{33} & 0 & 0 & c_{36}\\ 0 & 0 & 0 & c_{44} & c_{45} & 0\\ 0 & 0 & 0 & c_{45} & c_{55} & 0\\ c_{16} & c_{26} & c_{36} & 0 & 0 & c_{66}\end{bmatrix}.\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \gamma_{23}\\ \gamma_{31}\\ \gamma_{12}\end{bmatrix}$$

$$[S_{ij}]=[C_{ij}]^{-1}$$

(2.14)

Malzemede üç adet simetri ekseni olması durumunda, bir birinden bağımsız eleman sayısı 9 a düşer. Bu tip malzemelere *ortotropik malzeme* adı verilir.

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \tau_{23}\\ \tau_{31}\\ \tau_{12}\end{bmatrix}=\begin{bmatrix}c_{11} & c_{12} & c_{13} & 0 & 0 & 0\\ c_{12} & c_{22} & c_{23} & 0 & 0 & 0\\ c_{13} & c_{23} & c_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & c_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & c_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & c_{66}\end{bmatrix}.\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \gamma_{23}\\ \gamma_{31}\\ \gamma_{12}\end{bmatrix}$$

$$[S_{ij}]=[C_{ij}]^{-1}$$

(2.15)

Ortotropik bir malzeme için ***S*** uygunluk matrisi aşağıdaki şekilde tanımlanır:

$$[S_{ij}]=\begin{bmatrix}s_{11} & s_{12} & s_{13} & 0 & 0 & 0\\ s_{12} & s_{22} & s_{23} & 0 & 0 & 0\\ s_{13} & s_{23} & s_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & s_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & s_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & s_{66}\end{bmatrix}$$

(2.16)

S uygunluk matrisi mühendislik sabitleri cinsinden aşağıdaki gibi yazılabilir:

$$[S_{ij}] = \begin{bmatrix} \frac{1}{E_1} & \frac{-v_{21}}{E_2} & \frac{-v_{31}}{E_3} & 0 & 0 & 0 \\ \frac{-v_{12}}{E_1} & \frac{1}{E_2} & \frac{-v_{32}}{E_3} & 0 & 0 & 0 \\ \frac{-v_{13}}{E_1} & \frac{-v_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{31}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix}$$

(2.17)

Benzer şekilde $\boldsymbol{C}$ sertlik matrisi mühendislik sabitleri cinsinden aşağıdaki gibi yazılabilir;

$$[C_{ij}] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{22} & c_{23} & 0 & 0 & 0 \\ c_{13} & c_{23} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix}$$

$$C_{11} = \frac{1 - v_{23}.v_{32}}{E_2.E_3.\Delta} \ , \ C_{22} = \frac{1 - v_{13}.v_{31}}{E_1.E_3.\Delta} \ , \ C_{33} = \frac{1 - v_{21}.v_{12}}{E_2.E_1.\Delta}$$

$$C_{12} = \frac{v_{21} + v_{31}.v_{23}}{E_2.E_3.\Delta} \quad veya \quad C_{12} = \frac{v_{12} + v_{13}.v_{32}}{E_1.E_3.\Delta}$$

$$C_{13} = \frac{v_{31} + v_{21}.v_{32}}{E_2.E_3.\Delta} \quad veya \quad C_{13} = \frac{v_{13} + v_{12}.v_{23}}{E_1.E_2.\Delta}$$

$$C_{23} = \frac{v_{32} + v_{12}.v_{31}}{E_1.E_3.\Delta} \quad veya \quad C_{23} = \frac{v_{23} + v_{21}.v_{13}}{E_1.E_2.\Delta}$$

$$C_{44} = G_{23} \ , \quad C_{55} = G_{31} \ , \ C_{66} = G_{12}$$

$$\Delta = \frac{1 - v_{21}.v_{12} - v_{13}.v_{31} - v_{23}.v_{32} - 2.v_{21}.v_{32}.v_{13}}{E_1.E_2.E_3}$$

(2.18)

Burada;

E_1 , E_2 , E_3 Young elastiklik modülü (1 , 2 ve 3 yönlerinde) ,

v_{ij} Poisson oranı ($v_{ij} = {}^{-\varepsilon_j}/_{\varepsilon_i}$) ,

G_{12} , G_{31} , G_{23} kayma modülüdür (1-2 , 3-1 ve 2-3 düzlemlerinde) .

Ortotropik Malzemelerde Düzlem Gerilme Hali İçin Gerilme-Genleme İlişkisi:

Lamina en büyük yükü, elyaf doğrultusunda taşıyacak şekilde tasarlanır, diğer yönlerde çok büyük yükler taşımaz. Buna bağlı olarak 3 yönünde taşınan yük neredeyse sıfırdır. Bu durum göz önüne alınarak ve hesaplamaları kolaylaştırmak için laminada düzlem gerilme hali kabulü yapılabilir. Bu kabule göre gerilmeler;

$$\sigma_3 = 0 \quad \tau_{23} = 0 \quad \tau_{31} = 0$$
$$\sigma_1 \neq 0 \quad \sigma_2 \neq 0 \quad \tau_{12} \neq 0 \tag{2.19}$$

şeklinde tanımlanır. Buna göre genleme-gerilme ilişkisi;

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & 0 \\ S_{21} & S_{22} & 0 \\ 0 & 0 & S_{66} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} \tag{2.20}$$

şeklinde ifade edilir. Buradaki ifadeler mühendislik sabitleri cinsinden;

$$[S_{ij}] = \begin{bmatrix} \frac{1}{E_1} & \frac{-v_{21}}{E_2} & 0 \\ \frac{-v_{12}}{E_1} & \frac{1}{E_2} & 0 \\ 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \tag{2.21}$$

şeklinde yazılır. Gerilme-genleme ilişkisi ise;

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{21} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} \tag{2.22}$$

şeklinde ifade edilir. Buradaki ifadeler mühendislik sabitleri cinsinden;

$$Q_{11} = \frac{E_1}{1 - v_{21}.v_{12}} \qquad Q_{22} = \frac{E_2}{1 - v_{21}.v_{12}}$$

$$Q_{12} = Q_{21} = \frac{v_{12}.E_2}{1 - v_{21}.v_{12}}$$

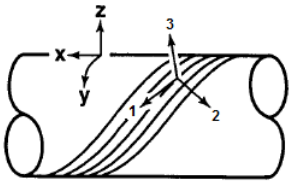
$$Q_{66} = G_{12}$$

(2.23)

şeklinde yazılır. $[Q_{ij}]$ matrisine *indirgenmiş sertlik matrisi* adı verilir. Düzlem gerilme halindeki ortotropik bir malzemeyi tanımlamak için E_1 , E_2 , v_{12} ve G_{12} mühendislik sabitlerinin bilinmesi yeterlidir.

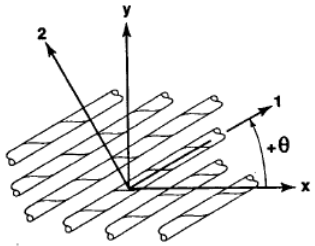
Herhangi Bir Doğrultudaki Lamina İçin Gerilme-Geleme İlişkisi:

Laminanın elyaf doğrultusu ile parçanın doğrultusu her zaman çakışmayabilir, böyle bir durumda Şekil 2.6`daki gibi laminadaki gerilme genleme ilişkisini parçanın doğrultusunda tanımlamak gerekebilir.



Şekil 2.6 Döndürülmüş Koordinatlar

Şekil 2.7`de görüldüğü gibi elyaf doğrultusunda uzanan 1-2 koordinat sistemindeki gerilmeler, her hangi bir x-y koordinat sistemindeki gerilmeler cinsinden aşağıdaki gibi hesaplanır.



Şekil 2.7 Döndürülmüş Lamina

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = [T] \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 2.\sin\theta.\cos\theta \\ \sin^2\theta & \cos^2\theta & -2.\sin\theta.\cos\theta \\ -\sin\theta.\cos\theta & \sin\theta.\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}$$

(2.24)

Burada [T] açısal dönüşüm matrisidir. Benzer şekilde her hangi bir x-y koordinat sistemindeki gerilmeler, elyaf doğrultusunda uzanan 1-2 koordinat sistemindeki gerilmeler cinsinden aşağıdaki gibi hesaplanabilir.

$$\begin{bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{bmatrix} = [T]^{-1}.\begin{bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{bmatrix} = \begin{bmatrix}\cos^2\theta & \sin^2\theta & -2.\sin\theta.\cos\theta\\ \sin^2\theta & \cos^2\theta & 2.\sin\theta.\cos\theta\\ \sin\theta.\cos\theta & -\sin\theta.\cos\theta & \cos^2\theta-\sin^2\theta\end{bmatrix}\begin{bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{bmatrix} \tag{2.25}$$

Buradaki θ, x-ekseni ile 1-ekseni arasındaki açıdır. Aynı şekilde x-y koordinat sistemindeki tensörel genlemeler;

$$\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}/2\end{bmatrix} = \begin{bmatrix}\cos^2\theta & \sin^2\theta & -2.\sin\theta.\cos\theta\\ \sin^2\theta & \cos^2\theta & 2.\sin\theta.\cos\theta\\ \sin\theta.\cos\theta & -\sin\theta.\cos\theta & \cos^2\theta-\sin^2\theta\end{bmatrix}\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}/2\end{bmatrix} \tag{2.26}$$

şeklinde hesaplanır. Mühendislik genlemeleri ise, tensörel genlemelerin [R] Reuter matrisi ile çarpılması ile bulunabilir;

$$\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{bmatrix} = [R]\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}/2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 2\end{bmatrix}\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}/2\end{bmatrix} \quad , \quad \begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{bmatrix} = [R]\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}/2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 2\end{bmatrix}\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}/2\end{bmatrix} \tag{2.27}$$

Düzlem gerilme halinde ortotropik bir malzeme için gerilme genleme ilişkisi;

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{bmatrix} = [Q].\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{bmatrix} \tag{2.28}$$

şeklinde ifade edilir. Bu ifade $[T]^{-1}$ ile soldan çarpılırsa;

$$\begin{bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{bmatrix} = [T]^{-1}.\begin{bmatrix}\sigma_1\\ \sigma_2\\ \tau_{12}\end{bmatrix} = [T]^{-1}.[Q].\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \gamma_{12}\end{bmatrix} \tag{2.29}$$

ifadesi elde edilir. (2.24) nolu denklem ters yönde;

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} = [T]^{-T}. \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix}$$

(2.30)

şeklinde yazılabilir. Buna göre (2.29) nolu denklem;

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = [T]^{-1}.[Q].[T]^{-T}. \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix}$$

(2.31)

haline dönüşür. Buradaki $[T]^{-1}.[Q].[T]^{-T}$ ifadesi bir $\left[\bar{Q}_{ij}\right]$ matrisi şeklinde yazılırsa

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix}$$

(2.32)

ifadesi elde edilir. Buna göre $\bar{Q}_{ij}$ ifadeleri;

$$\bar{Q}_{11} = Q_{11}\cos^4\theta + 2(Q_{12} + Q_{66})\sin^2\theta\cos^2\theta + Q_{22}\sin^4\theta$$
$$\bar{Q}_{12} = (Q_{11} + Q_{22} - 4Q_{66})\sin^2\theta\cos^2\theta + Q_{12}(\sin^4\theta + \cos^4\theta)$$
$$\bar{Q}_{22} = Q_{11}\sin^4\theta + 2(Q_{12} + Q_{66})\sin^2\theta\cos^2\theta + Q_{22}\cos^4\theta$$
$$\bar{Q}_{16} = (Q_{11} - Q_{12} - 2Q_{66})\sin\theta.\cos^3\theta + (Q_{12} - Q_{22} + 2Q_{66})\cos\theta.\sin^3\theta$$
$$\bar{Q}_{26} = (Q_{11} - Q_{12} - 2Q_{66})\cos\theta.\sin^3\theta + (Q_{12} - Q_{22} + 2Q_{66})\sin\theta.\cos^3\theta$$
$$\bar{Q}_{66} = (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66})\sin^2\theta\cos^2\theta + Q_{66}(\sin^4\theta + \cos^4\theta)$$

(2.33)

şekline hesaplanır. $\left[\bar{Q}_{ij}\right]$ matrisine *döndürülmüş indirgenmiş sertlik matrisi* adı verilir. Benzer şekilde her hangi bir x-y düzlemi için genleme-gerilme ilişkisi;

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = [T]^T.[S].[T]. \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}$$

(2.34)

şeklinde tanımlanabilir. Buna göre $\bar{S}_{ij}$ ifadeleri aşağıdaki şekilde hesaplanır:

$$\bar{S}_{11} = S_{11}\cos^4\theta + (2.S_{12} + S_{66})\sin^2\theta\cos^2\theta + S_{22}\sin^4\theta$$
$$\bar{S}_{12} = (S_{11} + S_{22} - S_{66})\sin^2\theta\cos^2\theta + S_{12}(\sin^4\theta + \cos^4\theta)$$

$$\bar{S}_{22} = S_{11}\sin^4\theta + (2.S_{12} + S_{66})\sin^2\theta\cos^2\theta + S_{22}\cos^4\theta$$
$$\bar{S}_{16} = (2S_{11} - 2S_{12} - S_{66})\sin\theta.\cos^3\theta + (2S_{11} - 2S_{12} - S_{66})\cos\theta.\sin^3\theta$$
$$\bar{S}_{26} = (2S_{11} - 2S_{12} - S_{66})\cos\theta.\sin^3\theta + (2S_{11} - 2S_{12} - S_{66})\sin\theta.\cos^3\theta$$
$$\bar{S}_{66} = 2(2S_{11} + 2S_{22} - 4S_{12} - S_{66})\sin^2\theta\cos^2\theta + S_{66}(\sin^4\theta + \cos^4\theta)$$

(2.35)

2.5. Laminatın Makromekanik Davranışı

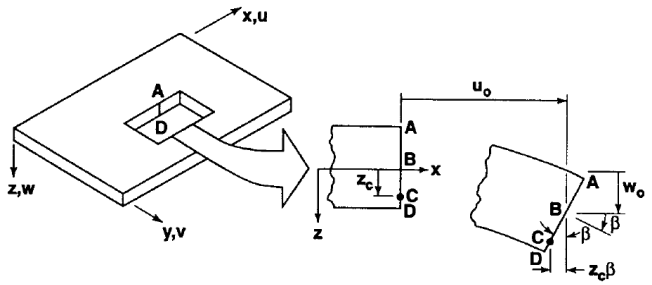
Laminatı bir araya getiren katmanlar ile oynanarak laminatın makro düzeyde nasıl bir davranış göstereceği belirlenebilir. Bu özellik laminatı tasarlanabilir kılar. Laminatın makromekanik davranışını belirleyebilmek için çeşitli teoriler geliştirilmiştir. Bu çalışmada klasik laminasyon teorisi kullanılmıştır.

Klasik Laminasyon Teorisi:

Laminat 3 boyutlu karmaşık bir yapıdır. Böyle bir yapının makromekanik davranışlarını belirleyebilmek ve hesaplamaları kolaylaştırabilmek için teoride bir takım kabuller yapılmıştır. Bu kabuller şöyle sıralanabilir:

- Her bir lamina diğerine bir yapıştırıcı katmanı ile mükemmel bir şekilde yapıştırılmıştır.
- Bu yapıştırıcı katmanı çok incedir ve katmanda kayma gerilmeleri oluşmaz.
- Laminalar bir birleri üzerinde kaymazlar.
- Laminatın orta düzlemine dik çizilen doğrular, deformasyon sonrasında da orta düzleme dik kalırlar.
- Laminatın orta düzlemine dik düzlemlerde kayma gerilmesi oluşmaz. ($\tau_{xz} = 0$, $\tau_{yz} = 0$)
- Deformasyon sonunda laminatın kalınlığı hiçbir yerde değişmez. ($\varepsilon_z = 0$)

Laminatta yükleme sonucu oluşan deformasyonlar Kirchhoff `un plak hipotezi laminata uyarlanarak hesaplanabilir [13].



Şekil 2.8 Laminada Yer Değiştirmeler

Şekil 2.8 de bir laminatın x-y düzleminden alınan bir kesitin deformasyon öncesi ve sonrası hali gösterilmiştir. Burada B noktası laminatın orta düzleminde yer almaktadır. B noktasının x yönündeki deformasyonu u_0 ve z yönündeki deformasyonu w_0 dır. Deformasyon sonrası AD doğrusu x yönünde β kadar eğim yapmıştır. Buna göre AD doğrusu üzerinde bulunan C noktasının deformasyonu;

$$u_c = u_0 - z_c.\beta \tag{2.36}$$

şeklinde tanımlanabilir. Burada β eğimi;

$$\beta = \frac{\partial w_0}{\partial x} \tag{2.37}$$

şeklindedir. Buradan yola çıkılarak laminattaki her hangi bir noktanın x yönündeki u deformasyonu;

$$u = u_0 - z.\frac{\partial w_0}{\partial x} \tag{2.38}$$

ifadesi ile, benzer şekilde y yönündeki v deformasyonu;

$$v = v_0 - z.\frac{\partial w_0}{\partial y} \tag{2.39}$$

ifadesi ile tanımlanabilir. Buna göre ε_x , ε_y ve γ_{xy} genlemeleri;

$$\varepsilon_x = \frac{\partial u}{\partial x} = \frac{\partial u_0}{\partial x} - z.\frac{\partial^2 w_0}{\partial x^2} \tag{2.40}$$

$$\varepsilon_y = \frac{\partial v}{\partial y} = \frac{\partial v_0}{\partial y} - z.\frac{\partial^2 w_0}{\partial y^2}$$

(2.41)

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} - 2.z.\frac{\partial^2 w_0}{\partial x.\partial y}$$

(2.42)

şeklinde ifade edilir. Bu ifadeler matris şeklinde aşağıdaki gibi yazılabilir;

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix} + z \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

(2.43)

$$\begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u_o}{\partial x} \\ \dfrac{\partial v_o}{\partial y} \\ \dfrac{\partial u_o}{\partial y} + \dfrac{\partial v_o}{\partial x} \end{bmatrix} \text{ ve } \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} = - \begin{bmatrix} \dfrac{\partial^2 w_o}{\partial x^2} \\ \dfrac{\partial^2 w_o}{\partial y^2} \\ 2\dfrac{\partial^2 w_o}{\partial x\,\partial y} \end{bmatrix}$$

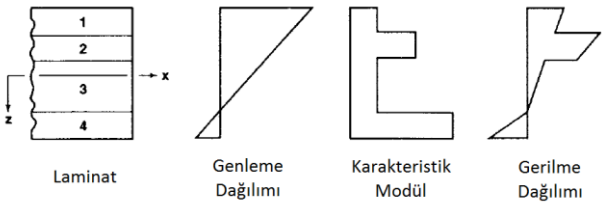
(2.44)

Buna göre laminatın her hangi bir k katmanındaki gerilme genleme ilişkisi;

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}_k = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}_k \left[\begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix} + z \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} \right]$$

(2.45)

şeklinde ifade edilir. Her bir katmandaki $\bar{Q}_{ij}$ farklı özellik gösterebilir, buna bağlı olarak laminatın kalınlığı boyunca oluşan gerilme lineer değildir bu durum Şekil 2.9 de gösterilmiştir.

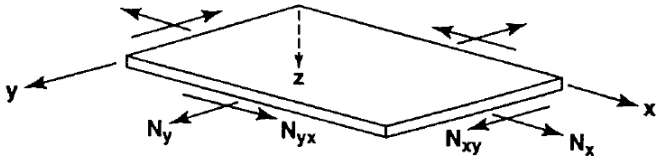


Şekil 2.9 Laminada Gerilme ve Genlemeler

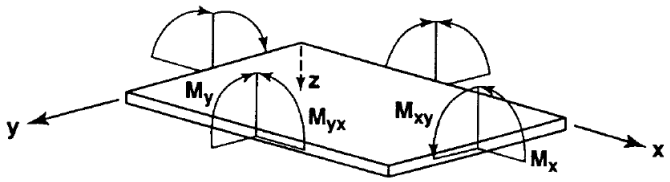
Laminatta oluşan kuvvetler ve momentler, gerilmelerin laminat kalınlığı boyunca integre edilmesi ile hesaplanabilir;

$$N_i = \int_{-t/2}^{t/2} \sigma_i \, dz \qquad M_i = \int_{-t/2}^{t/2} \sigma_i . z \, dz \qquad i = x, y, xy \tag{2.46}$$

Aşağıdaki şekillerde laminatta oluşan kuvvetler ve momentler gösterilmiştir.



Şekil 2.10 Laminada Kuvvetler



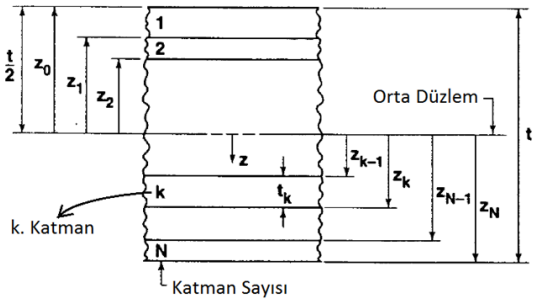
Şekil 2.11 Laminada Momentler

N adet katmandan oluşan bir laminatta oluşan kuvvet ve momentler, katmanlarda oluşan gerilmeler cinsinden;

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} dz = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}_k dz \tag{2.47}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} z . dz = \sum_{k=1}^{N} \int_{z_{k-1}}^{z_k} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}_k z . dz \tag{2.48}$$

şeklinde ifade edilir. Bu ifadelerde z_k ve z_{k-1} , Şekil 2.12 de gösterildiği gibi, bir k katmanının alt ve üst düzlemlerinin orta düzleme uzaklığıdır.



Şekil 2.12 Katman Kalınlıkları

Laminatta oluşan kuvvet ve momentler, lamina gerilme genleme ilişkisi kullanılarak aşağıdaki gibi yazılabilir;

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \sum_{k=1}^{N} \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}_k \left[\int_{z_{k-1}}^{z_k} \begin{bmatrix} {\varepsilon_x}^o \\ {\varepsilon_y}^o \\ {\gamma_{xy}}^o \end{bmatrix} dz + \int_{z_{k-1}}^{z_k} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} z.\, dz \right]$$

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \sum_{k=1}^{N} \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix}_k \left[\int_{z_{k-1}}^{z_k} \begin{bmatrix} {\varepsilon_x}^o \\ {\varepsilon_y}^o \\ {\gamma_{xy}}^o \end{bmatrix} z.\, dz + \int_{z_{k-1}}^{z_k} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix} z^2.\, dz \right]$$

(3.49)

Burada ${\varepsilon_x}^o$, ${\varepsilon_y}^o$, ${\gamma_{xy}}^o$, κ_x , κ_y ve κ_{xy} her bir katman için aynıdır, bu nedenle toplamın dışına alınabilir. Bu ifadeler, toplama ve integral işlemleri yapılarak yeniden düzenlenirse; A, B, D matrisleri cinsinden aşağıdaki gibi yazılır.

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{bmatrix} {\varepsilon_x}^o \\ {\varepsilon_y}^o \\ {\gamma_{xy}}^o \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{bmatrix} {\varepsilon_x}^o \\ {\varepsilon_y}^o \\ {\gamma_{xy}}^o \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

(2.50)

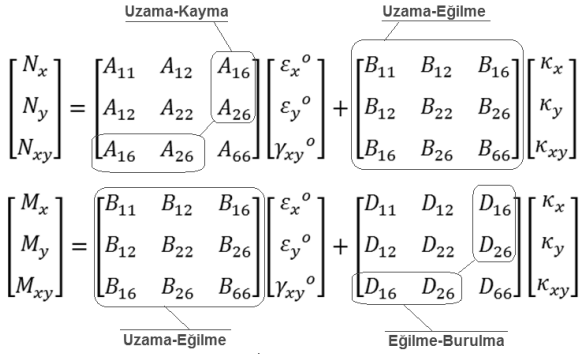
$$A_{ij} = \sum_{k=1}^{N} \left(\bar{Q}_{ij}\right)_k (z_k - z_{k-1})$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} \left(\bar{Q}_{ij}\right)_k ({z^2}_k - {z^2}_{k-1})$$

$$D_{ij} = \frac{1}{3}\sum_{k=1}^{N}\left(\overline{Q}_{ij}\right)_k\left(z^3{}_k - z^3{}_{k-1}\right)$$

(2.51)

A,B,D matrislerine laminat sertlik matrisleri adı verilir, bu matrislerin bütün elemanlarının sıfırdan farksız olması durumunda; laminatta uzama-kayma, uzama-eğilme ve eğilme-burulma bağlılıkları görülür. Bu bağlılıkları oluşturan A,B,D matris elemanları aşağıda verilmiştir.



Laminat Sertlik Matrisleri İle İlgili Özel Durumlar:

Laminatta görülen bağlılık davranışı çoğu uygulamada istenmez. Bu durumda laminatı oluşturan katmanların mekanik özellikleri ve katmanların dizilişi ile oynanarak bağlılıkları oluşturan A,B,D matris elemanları sıfıra eşitlenebilir. A,B,D laminat sertlik matrisleri ile ilgili özel durumlar aşağıda verilmiştir.

Tek Katmanlı Ortotropik Lamina:

Tek katmanlı ortotropik bir laminatta uzama-eğilme bağlılığı görülmez, diğerleri görülebilir. Ortotropik bir laminata ait A,B,D matrisleri aşağıda verilmiştir [13].

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix}$$

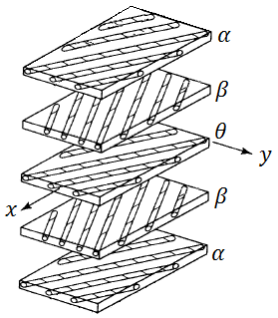
$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

$$A_{ij} = \overline{Q}_{ij}.t \quad , \quad B_{ij} = 0 \quad , \quad D_{ij} = \frac{\overline{Q}_{ij}.t^3}{12}$$

(2.52)

Simetrik Laminatlar:

Her bir katmanın malzeme özelliklerinin ve doğrultusunun, laminatın orta düzlemine göre simetrik olduğu laminatlara *simetrik laminat* adı verilir. Aşağıda bir simetrik laminatın katmanlarının dizilişi gösterilmiştir.



Şekil 2.13 Simetrik Laminat

Simetrik laminatlarda B_{ij} matrisi sıfırdır, bu yüzden uzama-eğilme bağlılığı görülmez. Simetrik laminatlarda A,B,D matrisleri en genel halde aşağıdaki gibidir [13]:

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

$$A_{ij} = \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k (z_k - z_{k-1})$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k (z^3{}_k - z^3{}_{k-1})$$

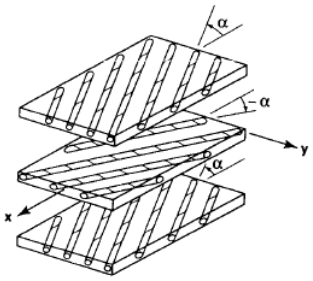
(2.53)

Katman doğrultusu $+\alpha/-\alpha$ şeklinde olan laminatlara açılı laminat adı verilir. Doğrultusu $+\alpha$ ve $-\alpha$ olan iki katman için $\overline{Q}_{16}$ ve $\overline{Q}_{26}$ terimleri aşağıdaki bağıntıya sahiptir

$$(\overline{Q}_{16})_{+\alpha} = -(\overline{Q}_{16})_{-\alpha}$$
$$(\overline{Q}_{26})_{+\alpha} = -(\overline{Q}_{26})_{-\alpha}$$

(2.54)

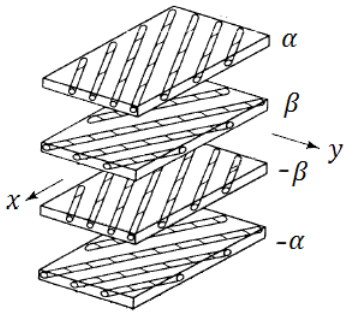
Simetrik açılı bir laminatta katman sayısı arttırıldıkça A_{16}, A_{26}, D_{16} ve D_{26} terimleri diğer terimlere göre çok küçük olur ve hatta bazı durumlarda sıfır kabul edilebilir.



Şekil 2.14 Simetrik Açılı Laminat

Antisimetrik Laminatlar:

Katman doğrultuları, laminatın orta düzlemine göre zıt işaretli olan laminatlara *antisimetrik laminat* adı verilir. Aşağıda bir asimetrik laminatın katman dizilişi gösterilmiştir.



Şekil 3.14 Antisimetrik Laminat

Daha önce de açıklandığı gibi doğrultusu $+\alpha$ ve $-\alpha$ olan iki katman için $\overline{Q}_{16}$ ve $\overline{Q}_{26}$ terimleri aşağıdaki bağıntıya sahiptir

$$(\overline{Q}_{16})_{+\alpha} = -(\overline{Q}_{16})_{-\alpha}$$
$$(\overline{Q}_{26})_{+\alpha} = -(\overline{Q}_{26})_{-\alpha}$$

Asimetrik bir laminatta katmanların doğrultuları orta düzleme göre $+\alpha$ ve $-\alpha$ şeklinde olduğu için

$$A_{ij} = \sum_{k=1}^{N} (\overline{Q}_{ij})_k (z_k - z_{k-1})$$

$$D_{ij} = \frac{1}{3}\sum_{k=1}^{N}\left(\bar{Q}_{ij}\right)_k\left(z^3{}_k - z^3{}_{k-1}\right)$$

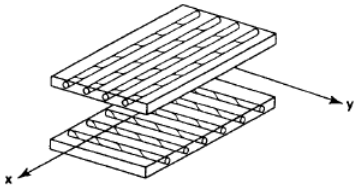
toplamlarına göre A_{16}, A_{26}, D_{16} ve D_{26} terimleri sıfır olarak bulunur. Buna bağlı olarak uzama-kayma ve eğilme-burulma bağlılıkları görülmez. Fakat asimetrik laminatlarda B_{ij} matrisi sıfıra eşit değildir. Bu yüzden uzama-eğilme bağlılığı görülür. Bu durumda bir asimetrik laminat için A,B,D matrisleri en genel halde aşağıdaki gibidir [13]:

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{bmatrix}\begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix}\begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix}\begin{bmatrix} \varepsilon_x{}^o \\ \varepsilon_y{}^o \\ \gamma_{xy}{}^o \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & 0 \\ D_{12} & D_{22} & 0 \\ 0 & 0 & D_{66} \end{bmatrix}\begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}$$

(2.55)

Dik Katmanlı Laminatlar:



Şekil 2.16 Dik Katmanlı Laminat

Katman doğrultuları birbirine dik olan laminatlara *dik katmanlı laminat* (cross-ply) adı verilir. Şekil 2.16 de asimetrik bir dik katmanlı laminat dizilişi gösterilmiştir. Asimetrik bir cross-ply için B_{ij} matrisi aşağıdaki gibi ifade edilir [13]:

$$B_{ij} = \begin{bmatrix} B_{11} & 0 & 0 \\ 0 & -B_{11} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(2.56)

BÖLÜM 3

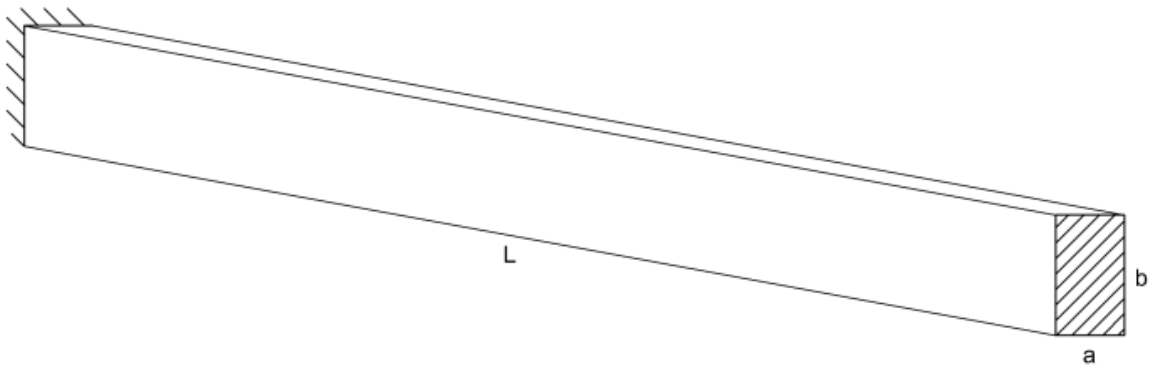
İNCE CİDARLI KOMPOZİT KİRİŞLER

Bu bölümde ince cidarlı kompozit kirişlerin mekanik bağlılıklar dikkate alınarak hareket denklemleri çıkarılmış ve bu denklemlerin sonlu elemanlar yöntemiyle çözümü hakkında teorik bilgi verilmiştir.

3.1. İnce Cidarlı Kompozit Kirişlerin Kinematiği

Genel Tanımlar:

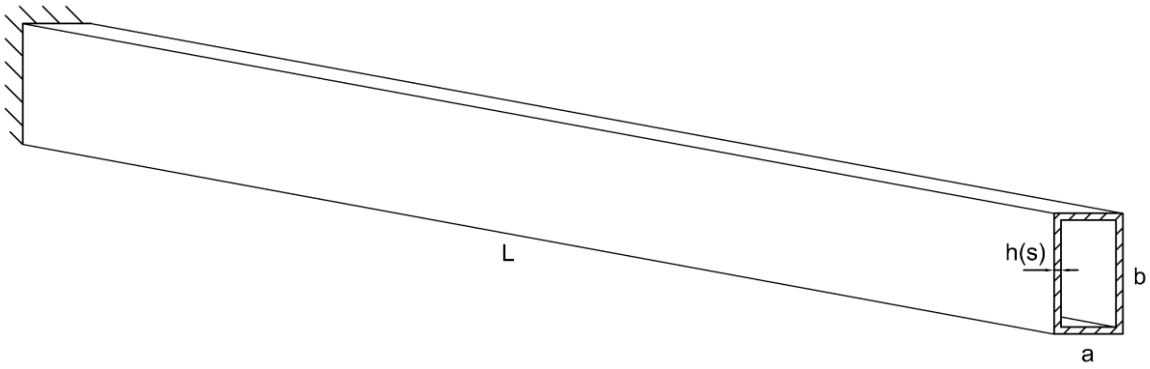
Kiriş, genellikle yanal veya eksenel yükleri taşımak için kullanılan bir tür yapı elemanıdır. Yanal yükler kirişi eğilmeye zorlar, eksenel yük ise kirişi çekmeye ya da basmaya zorlar. Aşağıdaki şekilde L uzunluğunda içi dolu dikdörtgen kesite sahip bir kiriş verilmiştir.



Şekil 3.1 Kiriş Geometrisi

Kesitin en büyük boyutunun kiriş uzunluğuna oranı 0.1 den küçük ise bu tip kirişlere *narin (slender) kiriş* adı verilir. Bu çalışmada "narin kiriş" ifadesi yerine "kiriş" ifadesi kullanılmıştır.

$$\frac{b}{L} \leq 0.1 \Rightarrow \ narin\ kiriş$$

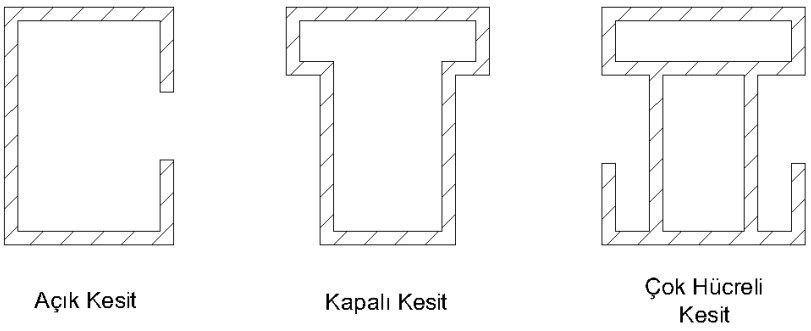


Şekil 3.2 İnce Cidarlı Kiriş

Vlasov`a göre içi boş bir kesite sahip bir kirişte en büyük cidar kalınlığının kesitin en büyük boyutuna oranı 0.1 den küçük ise bu kirişe *ince cidarlı kiriş* değil ise *kalın cidarlı kiriş* adı verilir. Şekilde cidar kalınlığı h olan ince cidarlı bir kiriş verilmiştir. İnce cidarlı bir kirişin yapımında kompozit malzeme kullanılmış ise bu kirişe ince cidarlı kompozit kiriş adı verilir [14].

$$\frac{h_{max}}{b} \leq 0.1 \Rightarrow \ ince\ cidarlı\ kiriş$$

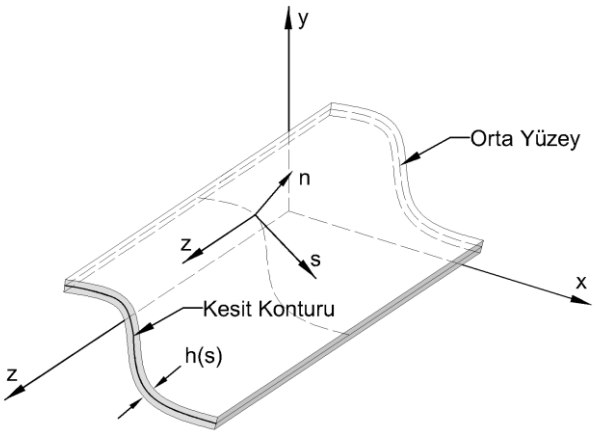
İnce veya kalın cidarlı bir kiriş kesitine göre açık, kapalı ve çok hücreli olmak üzere üç gruba ayrılır. Kesiti oluşturan cidar eğer açık bir form oluşturuyorsa buna *açık kesitli kiriş*, kapalı bir form oluşturuyorsa buna *kapalı kesitli kiriş* adı verilir. Kapalı bir cidar tarafından çevrelenen her bir alana *hücre* adı verilir. Bidren fazla hücreye sahip kirişlere *çok hücreli kesitli kiriş* adı verilir. Çok hücreli kesitli kirişler açık kesit özelliği gösterebilirler.



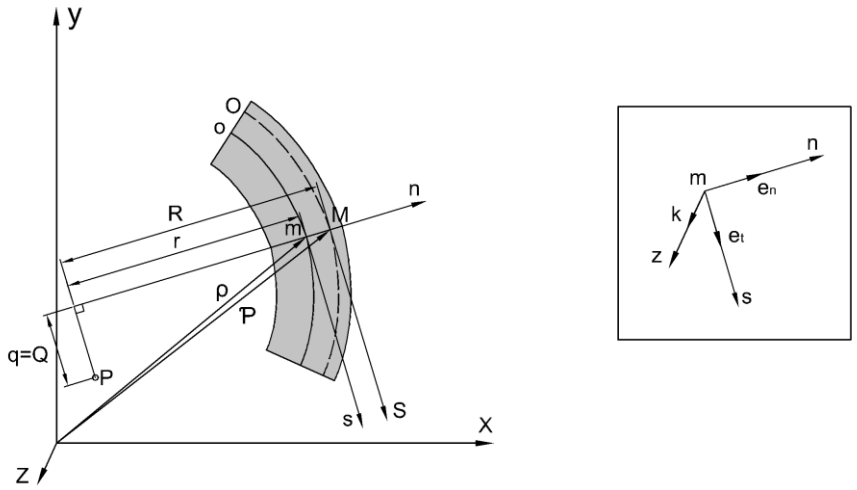
Şekil 3.3 İnce Cidarlı Kiriş Tipleri

İnce cidarlı kirişler iki koordinat sistemi birlikte kullanılarak tanımlanır. Bunlardan birincisi kartezyen koordinat sistemi (x,y,z); ikincisi dik eğrisel koordinat sistemidir (n,s,z). Birinci koordinat sistemi genel koordinat sistemi, ikinci koordinat

sistemi ise lokal koordinat sistemi olarak seçilmiştir. Bu çalışmada z ekseni kiriş doğrultusunda uzanmaktadır. x ve y eksenleri kirişin yanal eksenleridir. s ekseni kesit konturuna teğettir ve bu konturu saatin aksi yönünde takip eder. n ekseni ise cidar boyunca uzanır.



Şekil 3.4 Genel ve Lokal Koordinat Sistemleri



Şekil 3.5 İnce Cidarlı Kiriş Kesiti

İki koordinat sistemi arasındaki bağıntıyı kurmak için (0,0,z) referans noktasından orta düzlemdeki herhangi bir m (x,y,z) noktasına bir $\boldsymbol{\rho}$ (s,z) konum vektörü tanımlanırsa;

$$\boldsymbol{\rho}(s,z) = x(s)\boldsymbol{i} + y(s)\boldsymbol{j} + z\boldsymbol{k} \tag{3.1}$$

bağıntısı elde edilir. Benzer şekilde kesit üzerindeki her hangi bir M (X,Y,Z) noktası ise Ƥ (n,s,z) konum vektörü ile tanımlanabilir.

$$\mathrm{P}(n,s,z) = \rho(s,z) + n.\boldsymbol{e_n} \tag{3.2}$$

Bu iki denklem sayesinde kartezyen koordinat sistemi ile eğrisel koordinat sistemi arasındaki bağıntı kurulmuş olur. Burada eğrisel koordinat sistemine ait e_t (teğet) ve e_n (normal) birim vektörleri, kartezyen koordinat birim vektörleri cinsinden;

$$\boldsymbol{e_t} = \frac{d\rho}{ds} = \frac{dx(s)}{ds}\boldsymbol{i} + \frac{dy(s)}{ds}\boldsymbol{j} \tag{3.3}$$

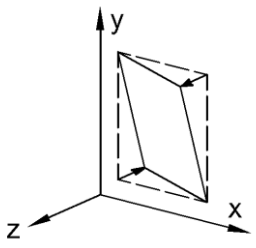
$$\boldsymbol{e_n} = \boldsymbol{e_t} \times \boldsymbol{k} = \frac{dy(s)}{ds}\boldsymbol{i} - \frac{dx(s)}{ds}\boldsymbol{j} \tag{3.4}$$

şeklinde ifade edilir.

Ana Kabuller:

İnce cidarı kompozit kirişler eğilme, burulma ve eksenel yükleme gibi çeşitli yükleme hallerine maruz kalmaktadır ve yapımında anizotropik malzeme kullanıldığı için ince cidarı kompozit kirişlerde uzama-kayma, uzama-burulma, eğilme-burulma gibi bağlılıklar görülür. Bütün bunları dikkate alarak çözüm yapabilmek için aşağıdaki kabuller yapılmıştır [11,12,14]:

- Kiriş kesiti kendi düzleminde rijittir, yani şekil değiştirmez ($\varepsilon_{xx}= \varepsilon_{yy}= \varepsilon_{xy}=0$) fakat kesit düzlemine dik doğrultuda çarpılma(warping) gösterebilir ($\varepsilon_{xz} \neq 0$, $\varepsilon_{yz} \neq 0$, $\varepsilon_{zz} \neq 0$).
- Yanal kayma genlemeleri kesit konturu boyunca eşittir.
- Cidar kalınlığının, herhangi bir noktadaki eğrilik yarıçapına oranı çok küçüktür.



Şekil 3.6 Çarpılma (Warping)

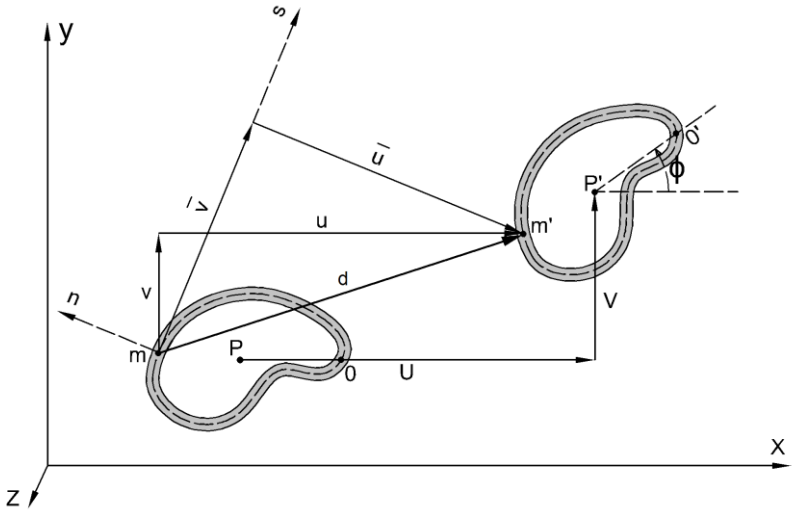
3.2. Yer Değiştirme Alanı

Yapılan kabullere göre kiriş kesiti rijittir, bu durumda kesit düzleminde sadece ötelenme ve dönme hareketi yapabilir. Kesitin bu hareketini tanımlamak için kesit düzleminde herhangi bir P (x_p,y_p) noktası referans alınır. Bu noktaya *kutup (pole)* adı verilir. Kesit x-ekseninde U, y-ekseninde V kadar ötelenir ve z-ekseninde ϕ kadar dönerse kutup noktasının yeni konumu P' olur, benzer şekilde kesit konturu üzerindeki herhangi bir m (x,y) noktasının yeni konumu m' olur. m noktasının yapmış olduğu u ve v ötelenmeleri; U, V ve ϕ cinsinden aşağıdaki şekilde tanımlanabilir [11,14]:

$$u(x,y,z,t) = U(z,t) - \left(y - y_p\right).\phi(z,t) \tag{3.5}$$

$$v(x,y,z,t) = V(z,t) + \left(x - x_p\right).\phi(z,t) \tag{3.6}$$

$$\phi(z,t) = \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial x}\right)_{x=x_p,\ y=y_p} \tag{3.7}$$



Şekil 3.7 Yer Değiştirmeler

Burada d yer değiştirme vektörü genel ve lokal koordinat sistemleri birim vektörleri cinsinden aşağıdaki gibi yazılabilir;

$$d(x,y,z,t) = u.\boldsymbol{i} + v.\boldsymbol{j} + w.\boldsymbol{k} \tag{3.8}$$

$$d(s,z,t) = \bar{u}.\boldsymbol{e_n} + \bar{v}\boldsymbol{e}_{\boldsymbol{t}} + \bar{w}.\boldsymbol{k}$$

(3.9)

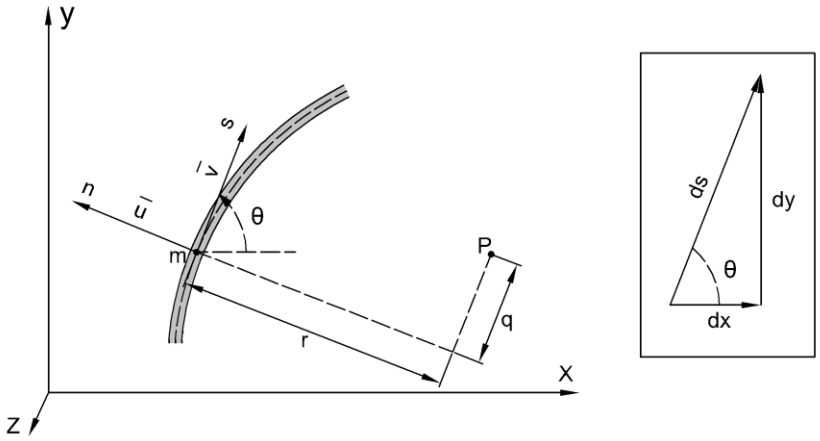
Bu ifadelerde $\bar{\boldsymbol{u}}$ ve $\bar{\boldsymbol{v}}$ orta yüzey üzerinde bulunan herhangi bir m noktasının lokal koordinat sistemi cinsinden n ve s eksenlerindeki yer değişim vektörü bileşenleridir ve değerleri aşağıdaki gibi hesaplanır;

$$\bar{\boldsymbol{u}} = d.\boldsymbol{e_n} = u\frac{dy}{ds} - v\frac{dx}{ds}$$

(3.10)

$$\bar{\boldsymbol{v}} = d.\boldsymbol{e_t} = u\frac{dx}{ds} + v\frac{dy}{ds}$$

(3.11)



Şekil 3.8 Orta Yüzeydeki Bir "m" Noktasının Tanımlanması

Bu ifadeler denklem (3.5) ve (3.6) ile tekrar düzenlenecek olursa $\bar{\boldsymbol{u}}$ ve $\bar{\boldsymbol{v}}$ aşağıdaki gibi tanımlanır;

$$\bar{\boldsymbol{u}} = U\frac{dy}{ds} - V\frac{dx}{ds} - q.\phi$$

(3.12)

$$\bar{\boldsymbol{v}} = U\frac{dx}{ds} + V\frac{dy}{ds} + r.\phi$$

(3.13)

Buradaki q ve r ifadeleri aşağıda verilmiştir;

$$\boldsymbol{q} = (x - x_p)\frac{dy}{ds} - (y - y_p)\frac{dx}{ds}$$

$$\boldsymbol{r} = (x - x_p)\frac{dx}{ds} + (y - y_p)\frac{dy}{ds}$$

(3.14,15)

Bu ifadelerde geometrik bağıntılardan;

$$\frac{dx}{ds} = \cos(\theta) \qquad ve \qquad \frac{dy}{ds} = \sin(\theta)$$

(3.16,17)

olarak tanımlanabilir.

Açık kesitli bir kirişte düzgün burulma sonucu oluşan $\bar{\gamma}_{sz}$ kayma gerilmesi [14];

$$\bar{\gamma}_{sz} = \frac{dx}{ds}\gamma_{xz} + \frac{dy}{ds}\gamma_{yz} + 2n\phi'$$

(3.18)

şeklinde tanımlanmıştır. Aynı zamanda $\bar{\gamma}_{sz}$ kayma gerilmesi aşağıdaki şekilde de tanımlanabilir [11];

$$\bar{\gamma}_{sz} = \frac{\partial \bar{v}}{\partial z} + \frac{\partial \bar{w}}{\partial s} = \phi'(z,t).\frac{F(s)}{h(s)}$$

(3.19)

bu iki ifadeden ve (3.13) numaralı denklemden $\frac{\partial \bar{w}}{\partial s}$ terimi aşağıdaki gibi çekilebilir;

$$\frac{\partial \bar{w}}{\partial s} = (\gamma_{xz} - U').\frac{dx}{ds} + (\gamma_{yz} - V').\frac{dy}{ds} - r(s).\phi'(z,t)$$

(3.20)

Bu ifade **s**`e göre integre edilirse;

$$\bar{w} = W(z,t) + (\gamma_{xz} - U').x(s) + (\gamma_{yz} - V').y(s) - \phi'(z,t).\omega(s)$$

(3.21)

denklemi elde edilir. Burada W kesit düzlemindeki ortalama eksenel yer değiştirme, x(s) ve y(s) kesit konturu üzerindeki herhangi bir m noktasının genel koordinat sistemine göre konum bileşenleridir. Bu denklemde γ_{xz} ve γ_{yz} ifadeleri ihmal edilip sıfıra eşitlenirse denklem aşağıdaki şekli alır;

$$\bar{w} = W(z,t) - U'.x(s) - V'.y(s) - \phi'(z,t).\omega(s)$$

(3.22)

Bu denklemde $\omega(s)$ çarpılma fonksiyonu değeri açık kesit için;

$$\omega(s) = \int_{s_0}^{s} r(s).ds$$

(3.23)

şeklinde kapalı kesit için ise;

$$\omega(s) = \int_{s_0}^{s} \left[r(s) - \frac{\oint r(s).ds}{\oint ds} \right].ds = \int_{s_0}^{s} \left[r(s) - \frac{F(s)}{h(s)} \right].ds$$

(3.24)

$$\oint \left(\frac{F(s)}{h(s)} \right).ds = 2.A$$

(3.25)

şeklinde tanımlanır. Burada $\oint(.)\boldsymbol{ds}$ kesit konturu boyunca alınan kapalı integral, F(s) St.Venant kayma akımı akışı, h(s) cidar kalınlığı, A ise kesit konturunun çevrelediği alandır. Bu konuda daha fazla açıklama için referans 14`e bakınız.

Kesit üzerindeki herhangi bir M noktasının $\bar{\boldsymbol{u}}$, $\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{w}}$ cinsinden yer değiştirmeleri;

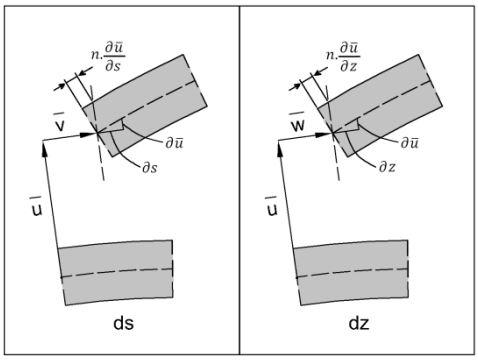
$$u_M(s,z,n) = \bar{u}(s,z)$$

$$v_M(s,z,n) = \bar{v}(s,z) - n.\frac{\partial \bar{u}(s,z)}{\partial s}$$

$$w_M(s,z,n) = \bar{w}(s,z) - n.\frac{\partial \bar{u}(s,z)}{\partial z}$$

(3.26,27,28)

şeklinde hesaplanır. Aşağıdaki şekilde bu denklemlerle ilgili açıklamalar verilmiştir.



Şekil 3.9 Eğrilikler

3.3. Genleme Alanı

Plaklar için çok küçük yer değiştirmeler sonucu oluşan genlemeler üç boyutlu elastisite teorisine göre lokal koordinatlar cinsinden [11];

$$\varepsilon_s = \bar{\varepsilon_s} + n.\bar{\kappa}_s$$

$$\varepsilon_z = \bar{\varepsilon_z} + n.\bar{\kappa}_z$$

$$\gamma_{sz} = \bar{\gamma}_{sz} + n.\bar{\kappa}_{sz}$$

$$\bar{\varepsilon_s} = \frac{\partial \bar{v}}{\partial s} \quad , \quad \bar{\kappa_s} = \frac{\partial^2 \bar{u}}{\partial s^2}$$
$$\bar{\varepsilon_z} = \frac{\partial \bar{w}}{\partial z} \quad , \quad \bar{\kappa_z} = \frac{\partial^2 \bar{u}}{\partial z^2}$$
$$\bar{\kappa}_{sz} = -2\frac{\partial^2 \bar{u}}{\partial s \partial z}$$

(3.29)

şeklinde tanımlanır. Plaklar için geçerli olan bu denklemler ince cidarlı kompozit kirişler teorisine adapte edilebilir. Burada $\bar{\boldsymbol{\varepsilon}}_{\boldsymbol{s}}$ ve $\bar{\boldsymbol{\kappa}}_{\boldsymbol{s}}$ ince cidarlı kompozit kirişler teorisinin birinci kabulüne göre sıfırdır. Orta yüzeydeki herhangi bir m noktası için $\bar{\boldsymbol{\varepsilon}}_{\boldsymbol{z}}$, $\bar{\boldsymbol{\kappa}}_{\boldsymbol{z}}$ ve $\bar{\boldsymbol{\kappa}}_{\boldsymbol{sz}}$ sırasıyla eksenel genleme, bieksenel eğrilik ve çarpılma eğriliğidir. Bu ifadeler (3.12), (3.22) ve (3.29) numaralı denklemler kullanılarak aşağıdaki gibi ifade edilebilir;

$$\bar{\varepsilon_z} = \varepsilon_z^0 + x.\kappa_y + y.\kappa_x + \omega.\kappa_\omega$$
$$\bar{\kappa_z} = \kappa_y.sin\theta - \kappa_x.cos\theta - \kappa_\omega.q$$
$$\bar{\kappa}_{sz} = \kappa_{sz}$$

(3.30)

Burada $\boldsymbol{\varepsilon_z^0}$, $\boldsymbol{\kappa_x}$, $\boldsymbol{\kappa_y}$, $\boldsymbol{\kappa_\omega}$ ve $\boldsymbol{\kappa_{sz}}$ sırasıyla eksenel genleme, x ve y doğrultusunda bieksenel eğrilik, çarpılma eğriliği ve burulma eğriliğidir. Bu ifadeler aşağıdaki gibi hesaplanırlar [11];

$$\varepsilon_z^0 = W'$$
$$\kappa_x = -V''$$
$$\kappa_y = -U''$$
$$\kappa_\omega = -\phi''$$
$$\kappa_{sz} = 2.\phi'$$

(3.31)

burada (.)' z-eksenine göre türevi ifade etmektedir. Sonuç olarak kesit üzerindeki herhangi bir M noktası için $\boldsymbol{\varepsilon_z}$ ve $\boldsymbol{\gamma_{sz}}$ genlemeleri (3.19), (3.26,27,28), (3.29) ve (3.30) nolu denklemlerden aşağıdaki gibi hesaplanır;

$$\varepsilon_z = \varepsilon_z^0 + (x + n.\sin\theta).\kappa_y + (y - n.cos\theta).\kappa_x + (\omega - n.q).\kappa_\omega$$
$$\gamma_{sz} = \left(n + \frac{\mathrm{F}}{2.\mathrm{h}}\right).\kappa_{sz}$$

(3.32)

3.4. Varyasyonel Formülasyon

Bir sistemde oluşan genleme enerjisi;

$$\mathcal{W} = \frac{1}{2}\int_{\tau} \sigma_{ij}.\varepsilon_{ij}.d\tau \qquad (i,j = 1,2,3)$$

$$d\tau = dn.ds.dz$$

(3.33)

şeklinde ifade edilir. İnce cidarlı kompozit bir kiriş için genleme enerjisi benzer şekilde;

$$\mathcal{W} = \frac{1}{2}\int_{\tau} (\sigma_z.\varepsilon_z + \sigma_{sz}.\gamma_{sz})d\tau$$

(3.34)

olarak yazılır. Bu ifade de (3.32) nolu denklem yerine konulursa;

$$\mathcal{W} = \frac{1}{2}\int_{\tau} \Big\{\sigma_z.\left(\varepsilon_z^0 + (x + n.\sin\theta).\kappa_y + (y - n.cos\theta).\kappa_x + (\omega - n.q).\kappa_\omega\right) + \sigma_{sz}.\left(n + \frac{\mathrm{F}}{2.\mathrm{h}}\right).\kappa_{sz}\Big\} d\tau$$

(3.35)

denklemi elde edilir. Bu denklemden genleme enerjisinin varyasyonu;

$$\delta\mathcal{W} = \int_0^l \{N_z.\delta\varepsilon_z + M_y.\delta\kappa_y + M_x.\delta\kappa_x + M_\omega.\delta\kappa_\omega + M_t.\delta\kappa_{sz}\}d\tau$$

(3.36)

şeklinde yazılabilir. Burada $\boldsymbol{N_z}$ eksenel kuvvet, $\boldsymbol{M_x}$ x-eksenindeki eğilme momenti, $\boldsymbol{M_y}$ y-eksenindeki eğilme momenti, $\boldsymbol{M_w}$ çarpılma momenti (bimonent), $\boldsymbol{M_t}$ ise burulma momentidir. Bu ifadeler aşağıdaki gibi tanımlanmıştır [11,12];

$$N_z = \int_A \sigma_z.ds.dn$$

$$M_y = \int_A \sigma_z(x + n.sin\theta)ds.dn$$

$$M_x = \int_A \sigma_z(y - n.cos\theta)ds.dn$$

$$M_\omega = \int_A \sigma_z(\omega - n.q)ds.dn$$

$$M_t = \int_A \sigma_z(n + \frac{F}{2.h})ds.dn$$

(3.37)

Bir sistemde oluşan kinetik enerji;

$$\boldsymbol{\mathcal{T}} = \frac{1}{2}\int_{\tau} \rho(\dot{V}_i.\dot{V}_i)d\tau \qquad (i,j = 1,2,3)$$

$$d\tau = dn.ds.dz$$

(3.38)

şeklinde ifade edilir. Burada $\boldsymbol{\rho}$ yoğunluk, $\boldsymbol{\dot{V}_i}$ i-yönündeki yer değiştirmenin zamana göre türevidir. İnce cidarlı kompozit bir kiriş için kinetik enerji;

$$\boldsymbol{\mathcal{T}} = \frac{1}{2}\int_{\tau} \rho(\dot{u}^2 + \dot{v}^2 + \dot{w}^2)d\tau$$

(3.39)

şeklinde yazılır. Kinetik enerjinin varyasyonu yer değişim denklemleri kullanılarak;

$$\boldsymbol{\mathcal{T}} = \frac{1}{2}\int_{\tau} \rho\{\dot{U}\delta\dot{U} + \dot{V}\delta\dot{V} + \dot{W}\delta\dot{W} + (q^2 + r^2 + 2.r.n + n^2)\dot{\phi}\delta\dot{\phi} + (\dot{\phi}\delta\dot{U} + \dot{U}\delta\dot{\phi})[n.cos\theta - (y - y_p)] + (\dot{\phi}\delta\dot{V} + \dot{V}\delta\dot{\phi})[n.sin\theta + (x - x_p)]\}d\tau$$

$$(x - x_p) = q.cos\theta + r.sin\theta$$

$$(y - y_p) = q.sin\theta - r.cos\theta$$

(3.40)

şeklinde yazılabilir [12].

Varyasyonel kinetik enerji ve genleme enerjisi ifadelerine Hamilton prensibi uygulanırsa

$$\delta\int_{t1}^{t2} (\mathcal{T} - \mathcal{W}).dt = 0$$

(3.41)

denklemi elde edilir. Bu denklem kullanılarak zayıf bağ denklemi (weak statement);

$$\delta\boldsymbol{\mathcal{T}} = \int_{t1}^{t2}\int_{0}^{l} \{m_0\dot{W}\delta\dot{W} + [m_0\dot{U} + (m_c - m_y + m_0y_p)\dot{\phi}]\delta\dot{U} + [m_0\dot{V} + (m_s + m_x - m_0x_p)\dot{\phi}]\delta\dot{V} + [(m_c - m_y + m_0y_p)\dot{U} + (m_s + m_x - m_0x_p)\dot{V} + (m_p + m_2 + 2.m_\omega)\dot{\phi}]\delta\dot{\phi} - N_z\delta W' + M_y\delta U'' + M_x\delta V'' + M_\omega\delta\phi'' - M_t\delta\phi\}\, dz.dt$$

(3.42)

şeklinde yazılabilir. Burada $\boldsymbol{m_0}$, $\boldsymbol{m_c}$, $\boldsymbol{m_p}$, $\boldsymbol{m_s}$, $\boldsymbol{m_x}$, $\boldsymbol{m_y}$, $\boldsymbol{m_\omega}$ ve $\boldsymbol{m_2}$ atalet katsayılarıdır ve aşağıdaki şekilde verilmiştir [12];

$$m_0 = I_0 \int_s ds$$

$$m_c = I_1 \int_s cos\theta . ds$$

$$m_s = I_1 \int_s sin\theta . ds$$

$$m_p = I_0 \int_s (q^2 + r^2) . ds$$

$$m_x = I_0 \int_s x . ds$$

$$m_y = I_0 \int_s y . ds$$

$$m_\omega = I_1 \int_s r . ds$$

$$m_2 = I_2 \int_s ds$$

$$(I_0, I_1, I_2) = \int_n \rho(1, n, n^2) dn$$

(3.43)

3.5. Bünye Denklemleri

İnce cidarlı kompozit kirişlerin yapımında, sürekli elyaf takviyeli tabakalı kompozit malzeme kullanılır. Elyaf takviyeli tabakalı kompozit malzemelerin mekanik özelliklerini tanımlayan denklemler bölüm 2 de açıklanmıştır. Bu denklemlerden yola çıkılarak kiriş cidarındaki bir k laminasının gerilme genleme ilişkisi lokal koordinatlar cinsinden;

$$\begin{Bmatrix} \sigma_z \\ \sigma_{sz} \end{Bmatrix}_k = \begin{bmatrix} \overline{Q^*}_{11} & \overline{Q^*}_{16} \\ \overline{Q^*}_{16} & \overline{Q^*}_{66} \end{bmatrix}_k \begin{Bmatrix} \varepsilon_z \\ \gamma_{sz} \end{Bmatrix}$$

$$\overline{Q^*}_{11} = \overline{Q}_{11} - \frac{\overline{Q}_{12} . \overline{Q}_{12}}{\overline{Q}_{22}}$$

$$\overline{Q^*}_{16} = \overline{Q}_{16} - \frac{\overline{Q}_{12} . \overline{Q}_{26}}{\overline{Q}_{22}}$$

$$\overline{Q^*}_{66} = \overline{Q}_{66} - \frac{\overline{Q}_{26} . \overline{Q}_{26}}{\overline{Q}_{22}}$$

(3.44)

şeklinde tanımlanır. Burada $[\overline{\boldsymbol{Q}^*}]$ matrisi lokal koordinatlar cinsinden düzlem gerilme kabulüne göre döndürülmüş indirgenmiş sertlik matrisidir. Kirişin bünye denklemleri ise (3.32), (3.37) ve (3.44) nolu denklemlerden yola çıkarak aşağıdaki gibi yazılabilir;

$$\begin{bmatrix} N_z \\ M_y \\ M_x \\ M_\omega \\ M_t \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} \\ & E_{22} & E_{23} & E_{24} & E_{25} \\ & & E_{33} & E_{34} & E_{35} \\ & sim. & & E_{44} & E_{45} \\ & & & & E_{55} \end{bmatrix} . \begin{bmatrix} \varepsilon_z^0 \\ \kappa_y \\ \kappa_x \\ \kappa_\omega \\ \kappa_{sz} \end{bmatrix} \tag{3.45}$$

Burada $[E_{ij}]$ sertlik matrisidir ve elemanları aşağıdaki şekilde hesaplanır [12];

$$E_{11} = \int_s A_{11}.ds$$

$$E_{12} = \int_s (A_{11}.x + B_{11}.sin\theta)ds$$

$$E_{13} = \int_s (A_{11}.y - B_{11}.cos\theta)ds$$

$$E_{14} = \int_s (A_{11}.w - B_{11}.q)ds$$

$$E_{15} = \int_s \left(A_{16}.\frac{F}{2.h} + B_{16}\right)ds$$

$$E_{22} = \int_s (A_{11}.x^2 + 2.B_{11}.x.sin\theta + D_{11}.sin^2\theta)ds$$

$$E_{23} = \int_s (A_{11}.x.y + B_{11}.(y.sin\theta - x.cos\theta) - D_{11}.sin\theta.cos\theta)ds$$

$$E_{24} = \int_s (A_{11}.x.w + B_{11}.(w.sin\theta - q.x) - D_{11}.q.sin\theta)ds$$

$$E_{25} = \int_s \left(A_{16}.\frac{F}{2.h}.x + B_{16}\left(x + \frac{F.sin\theta}{2.h}\right) + D_{16}.sin\theta\right)ds$$

$$E_{33} = \int_s (A_{11}.y^2 - 2.B_{11}.y.cos\theta + D_{11}.cos^2\theta)ds$$

$$E_{34} = \int_s (A_{11}.y.w - B_{11}.(w.cos\theta + q.y) + D_{11}.q.cos\theta)ds$$

$$E_{35} = \int_s \left(A_{16}.\frac{F}{2.h}.y + B_{16}\left(y - \frac{F.cos\theta}{2.h}\right) - D_{16}.cos\theta\right)ds$$

$$E_{44} = \int_s (A_{11}.w^2 - 2.B_{11}.w.q + D_{11}.q^2)ds$$

$$E_{45} = \int_s \left(A_{16}.\frac{F}{2.h}.w + B_{16}\left(w - \frac{F.q}{2.h}\right) - D_{16}.q\right)ds$$

$$E_{55} = \int_s \left(A_{66}.\frac{F^2}{4.h^2}.w + B_{66}.\frac{F}{h} + D_{66}\right)ds$$

(3.46)

burada $\boldsymbol{A_{ij}}$, $\boldsymbol{B_{ij}}$ ve $\boldsymbol{D_{ij}}$ sertlik matrisleri;

$$\mathrm{A_{ij}} = \sum_{\mathrm{k=1}}^{\mathrm{N}} \left(\overline{\mathrm{Q}}_{\mathrm{ij}}\right)_{\mathrm{k}} (\mathrm{n_k} - \mathrm{n_{k-1}})$$

$$\mathrm{B_{ij}} = \frac{1}{2} \sum_{\mathrm{k=1}}^{\mathrm{N}} \left(\overline{\mathrm{Q}}_{\mathrm{ij}}\right)_{\mathrm{k}} (\mathrm{n^2_k} - \mathrm{n^2_{k-1}})$$

$$\mathrm{D_{ij}} = \frac{1}{3} \sum_{\mathrm{k=1}}^{\mathrm{N}} \left(\overline{\mathrm{Q}}_{\mathrm{ij}}\right)_{\mathrm{k}} (\mathrm{n^3_k} - \mathrm{n^3_{k-1}})$$

(3.47)

şeklinde hesaplanır.

3.6. Hareket Denklemleri

İnce cidarlı kompozit kirişin hareket denklemleri (3.42) nolu denklem integre edilip $\boldsymbol{\delta U}$, $\boldsymbol{\delta V}$, $\boldsymbol{\delta W}$ ve $\boldsymbol{\delta}\phi$ katsayılarına göre tekrar düzenlenirse;

$$N'_z = m_0.\ddot{W}$$

$$M''_y = m_0.\ddot{U} + (m_c - m_y + m_0.y_p)\ddot{\Phi}$$

$$M''_x = m_0.\ddot{V} + (m_s + m_x - m_0.x_p)\ddot{\Phi}$$

$$M''_\omega + 2.M'_t = (m_c - m_y + m_0.y_p)\ddot{U} + (m_s + m_x - m_0.x_p)\ddot{V} + (m_p + m_2 + 2.m_\omega)\ddot{\Phi}$$

(3.48)

şeklinde bulunur. Bu ifadeler (3.45) nolu denklem kullanılarak $\boldsymbol{E_{ij}}$ lamina sertlikleri cinsinden aşağıdaki gibi yazılabilir [12];

$$
\begin{aligned}
&E_{11}.W^{''} - E_{12}.U^{'''} - E_{13}.V^{'''} - E_{14}.\phi^{'''} + 2.E_{15}.\phi^{''} = m_0.\ddot{W} \\
&E_{12}.W^{'''} - E_{22}.U^{iv} - E_{23}.V^{iv} - E_{24}.\phi^{iv} + 2.E_{15}.\phi^{'''} \\
&= m_0.\ddot{U} + (m_c - m_y + m_0.y_p)\ddot{\phi} \\
&E_{13}.W^{'''} - E_{23}.U^{iv} - E_{33}.V^{iv} - E_{34}.\phi^{iv} + 2.E_{35}.\phi^{'''} \\
&= m_0.\ddot{V} + (m_s + m_x - m_0.x_p)\ddot{\phi} \\
&E_{14}.W^{'''} + 2.E_{15}.W^{''} - E_{24}.U^{iv} - 2.E_{25}.U^{'''} - E_{34}.V^{iv} - 2.E_{35}.V^{'''} - E_{44}.\phi^{iv} \\
&+ 4.E_{55}.\phi^{''} \\
&= (m_c - m_y + m_0.y_p)\ddot{U} + (m_s + m_x - m_0.x_p)\ddot{V} + (m_p + m_2 + 2.m_\omega)\ddot{\phi}
\end{aligned}
\tag{3.49}
$$

Bu ifadelerde U, V, W ve ϕ`nin bir birine tamamen bağlı olduğu görülür.

3.7. Sonlu Elemanlar Formülasyonu

İnce cidarlı kompozit kirişin hareket denklemleri görüldüğü üzere U, V, W ve ϕ`yer değiştirmeleri cinsinden birçok bağlılık içermektedir. Bu tip bağlılıklar içeren denklemler yer değiştirme temelli sonlu elemanlar yöntemi kullanılarak çözülebilir. Bu yönteme göre U, V, W ve ϕ`yer değiştirmeleri $\boldsymbol{\varphi_j}$ ve $\boldsymbol{\psi_j}$ şekil fonksiyonları cinsinden aşağıdaki gibi tanımlanabilir;

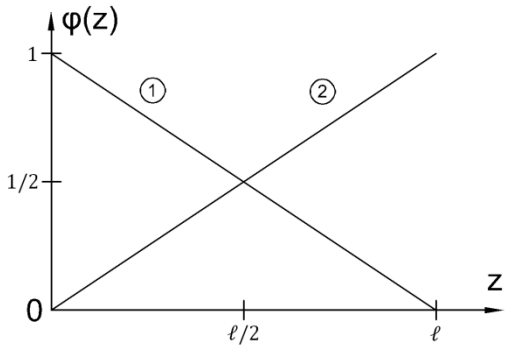
$$
\begin{aligned}
W &= \sum_{j=1}^{n} w_j.\varphi_j \\
U &= \sum_{j=1}^{n} u_j.\psi_j \\
V &= \sum_{j=1}^{n} v_j.\psi_j \\
\phi &= \sum_{j=1}^{n} \phi_j.\psi_j
\end{aligned}
\tag{3.50}
$$

Burada n sonlu elemanlar formülasyonundaki toplam düğüm sayısını, j ise herhangi bir düğümü ifade eder. Sonlu elemanlar yönteminde bir ifadenin iki düğüm noktası arasındaki değişimi, şekil fonksiyonları ile hesaplanır. (3.50) nolu denklemlerde kullanılan $\boldsymbol{\varphi}$ şekil fonksiyonu, tek boyutlu Lagrange interpolasyon fonksiyonu

cinsinden; $\boldsymbol{\psi}$ şekil fonksiyonu ise hermetik-kübik interpolasyon fonksiyonu cinsinden aşağıdaki gibi tanımlanmıştır;

$$\varphi(z) = \left\{1 - \frac{z}{\ell} \quad , \quad z/\ell\right\}$$

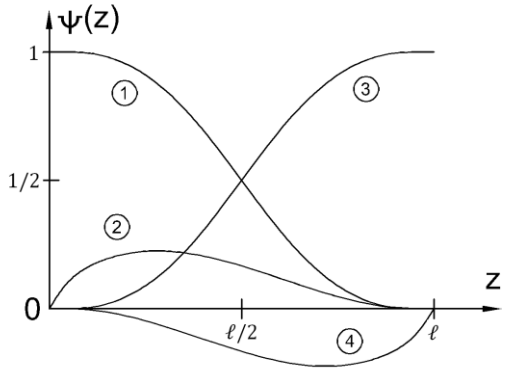
(3.51)



Şekil 3.10 Lagrange Interpolasyon Fonksiyonu

$$\psi(z) = \left\{1 - \frac{3z^2}{\ell^2} + \frac{2z^3}{\ell^3} \quad , \quad \frac{3z^2}{\ell^2} - \frac{2z^3}{\ell^3} \quad , \quad z - \frac{2z^2}{\ell} + \frac{z^3}{\ell^3} \quad , \quad -\frac{z^2}{\ell} + \frac{z^3}{\ell^2}\right\}$$

(3.52)



Şekil 3.11 Hermetik-Kübik İnterpolasyon Fonksiyonu

Bu ifadeler (3.42) nolu zayıf bağ denklemi ile tekrar düzenlenirse;

$$([K] - \lambda[M])\{\Delta\} = \{f\}$$

(3.53)

sonlu elemanlar formülasyonu elde edilir. Burada $\{\Delta\}$ yer değiştirme vektörü, $\{\boldsymbol{f}\}$ kuvvet vektörü, $[\boldsymbol{K}]$ eleman sertlik matrisi, $[\boldsymbol{M}]$ ise eleman kütle matrisidir ve aşağıdaki gibi tanımlanmışlardır [12];

$$\{\Delta\} = \begin{bmatrix} W \\ U \\ V \\ \phi \end{bmatrix}$$

$$\{f\} = \begin{bmatrix} F_w \\ F_u \\ F_v \\ F_\phi \end{bmatrix}$$

$$[K] = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ & K_{22} & K_{23} & K_{24} \\ & & K_{33} & K_{34} \\ sim & & & K_{44} \end{bmatrix}$$

$$[M] = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ & M_{22} & M_{23} & M_{24} \\ & & M_{33} & M_{34} \\ sim & & & M_{44} \end{bmatrix} \tag{3.54}$$

$[\boldsymbol{K}]$ ve $[\boldsymbol{M}]$ matrislerinin elemanları aşağıdaki şekilde hesaplanır;

$$K_{ij}^{11} = \int_0^l E_{11}\varphi'_i\varphi'_j dz$$

$$K_{ij}^{12} = -\int_0^l E_{12}\varphi'_i\psi''_j dz$$

$$K_{ij}^{13} = -\int_0^l E_{13}\varphi'_i\psi''_j dz$$

$$K_{ij}^{14} = \int_0^l (2E_{15}\varphi'_i\psi'_j - E_{14}\varphi'_i\psi''_j) dz$$

$$K_{ij}^{22} = \int_0^l E_{22}\psi''_i\psi''_j dz$$

$$K_{ij}^{23} = \int_0^l E_{23}\psi''_i\psi''_j dz$$

$$K_{ij}^{24} = \int_0^l \left(E_{24}\psi''_i\psi''_j - 2E_{25}\psi''_i\psi'_j\right) dz$$

$$K_{ij}^{33} = \int_0^l E_{33}\psi'_i\psi'_j dz$$

$$K_{ij}^{34} = \int_0^l \left(E_{34}\psi'_i\psi'_j - 2E_{35}\psi'_i\psi'_j\right) dz$$

$$K_{ij}^{44} = \int_0^l \left(E_{44}\psi'_i\psi'_j - 2E_{45}\left(\psi'_i\psi''_j + \psi'_i\psi'_j\right) + 4E_{55}\psi'_i\psi'_j\right) dz$$

(3.55)

$$M_{ij}^{11} = \int_0^l m_0\varphi_i\varphi_j dz$$

$$M_{ij}^{22} = M_{ij}^{33} = \int_0^l m_0\psi_i\psi_j dz$$

$$M_{ij}^{24} = \int_0^l (m_c - m_y + m_0.y_p)\psi_i\psi_j dz$$

$$M_{ij}^{34} = \int_0^l (m_s + m_x - m_0.x_p)\psi_i\psi_j dz$$

$$M_{ij}^{44} = \int_0^l (m_p + m_2 + 2m_\omega)\psi_i\psi_j dz$$

(3.56)

Diğer [$\boldsymbol{M}$] matrisi elemanları sıfıra eşittir.

Bu bölümde tanımlanan denklemler yardımıyla ince cidarlı kompozit bir kirişin her hangi bir yükleme halinde her hangi bir noktasındaki yer değiştirmesi bulunabilir ve kirişin serbest titreşim analizi yapılabilir; fakat bu denklemleri el ile çözmek zordur, çözümleri yapabilmek için bir bilgisayar yazılımı geliştirilmiştir. Yazılım ile ilgili açıklamalar Bölüm 4 de verilmiştir.

BÖLÜM 4

GELİŞTİRİLEN SONLU ELEMANLAR YAZILIMI

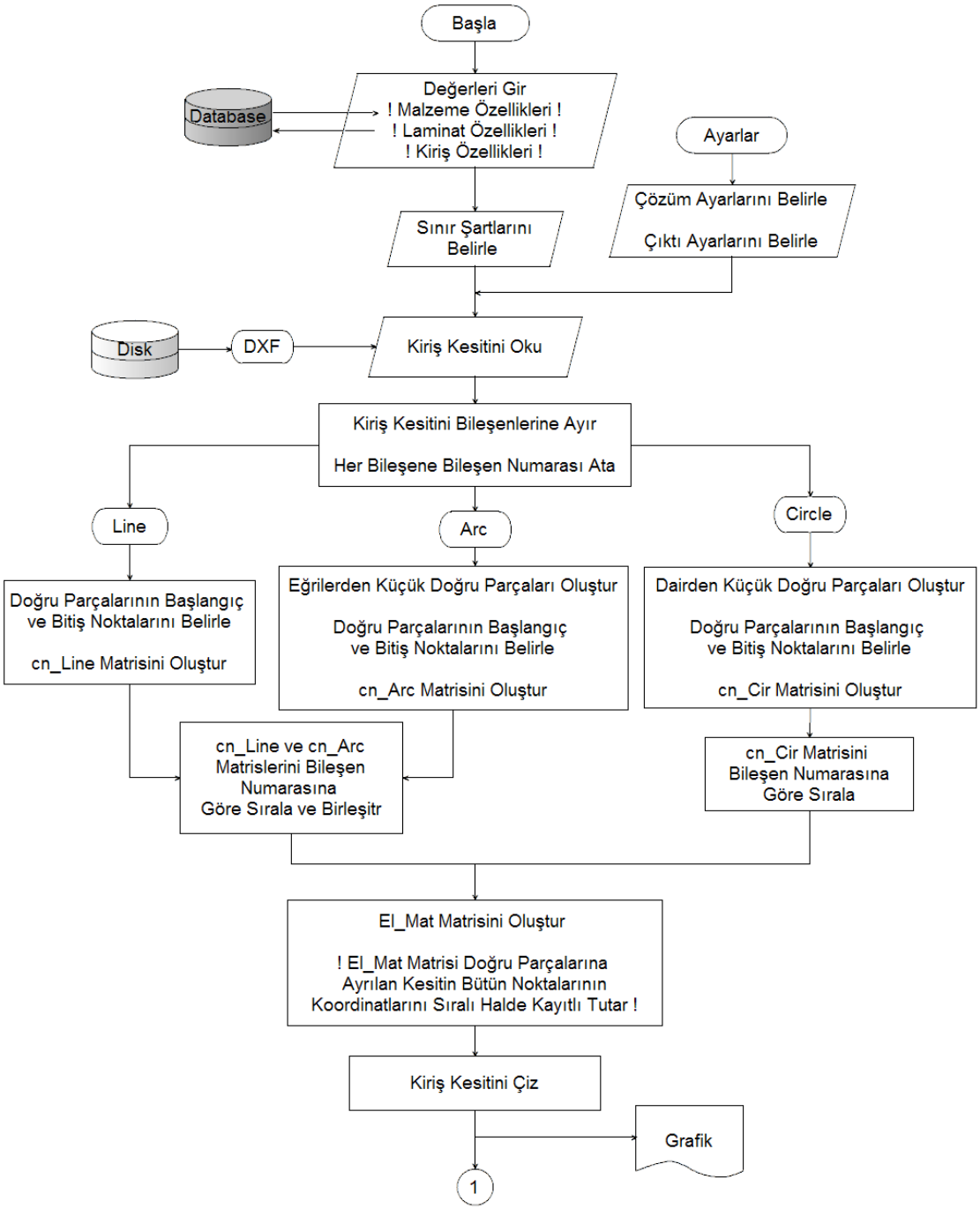
Bu bölümde ince cidarlı kompozit kirişlerin statik ve dinamik analizini yapmak için geliştirilen sonlu elemanlar yazılımı, yazılımın akış şeması ve çalışma prensibi hakkında bilgiler verilmiştir.

4.1. Yazılım Hakkında Genel Bilgi

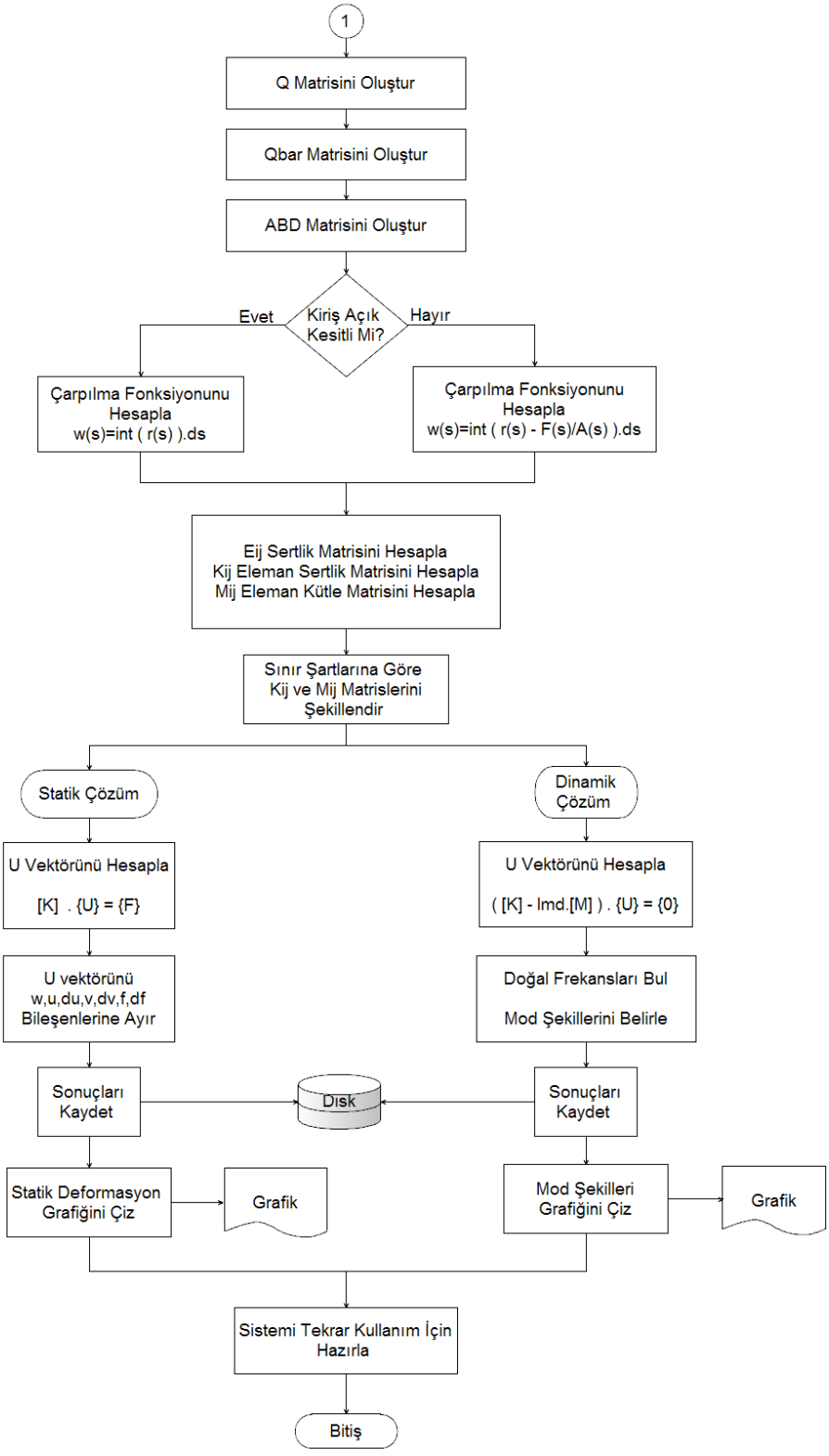
İnce cidarlı kompozit kirişler ile ilgili teori ve çözüm yöntemi Bölüm-3 de açıklanmıştır. İnce cidarlı kompozit kirişler teorisi herhangi bir kesite sahip kirişe uygulanmaya müsaittir fakat denklemlerde görüldüğü üzere çözümün elde edilmesi çok zor ve zaman alıcıdır. Daha önce yapılan çalışmalarda ise dikdörtgen, daire, I ve U gibi basit bir kesite sahip kompozit kirişler incelenmiş ve bu kesitlere ait çeşitli çözümler, çözüm yöntemleri verilmiştir fakat kesitin değişmesi durumunda verilen hazır çözümlerin bir geçerliliği kalmamakta, hesaplamaların baştan yapılması gerekmektedir. Günümüzdeki uygulamalarda özellikle rüzgâr türbini kanadı gibi uygulamalarda ise giderek daha karmaşık kesitlere sahip elemanlar kullanılmaktadır ve çoğu uygulamada kesitin tekrar tekrar modellenip optimize edilmesi gerekmektedir. Bu ve benzeri sebeplerden dolayı bu çalışmada günümüz uygulamalarında verimli bir şekilde kullanılabilecek açık veya kapalı herhangi bir kesite sahip ince cidarlı kompozit kirişlerin çeşitli yükleme koşulları altında statik ve dinamik analizini yapabilecek bir yazılım geliştirmek hedeflenmiştir. Yazılımın ismi *Thin Walled Beam – Finite Element Analysis* kelimelerinin baş harflerinden yola çıkılarak **TWB-FEA** olarak seçilmiştir. Yazılım şu anda tek hücreli açık veya kapalı sabit kesitli ince cidarlı kirişlerin çeşitli yükleme koşulları altında statik analizini ve serbest titreşim analizini yapabilmektedir.

4.2. Yazılımın Akış Şeması ve Çalışma Prensibi

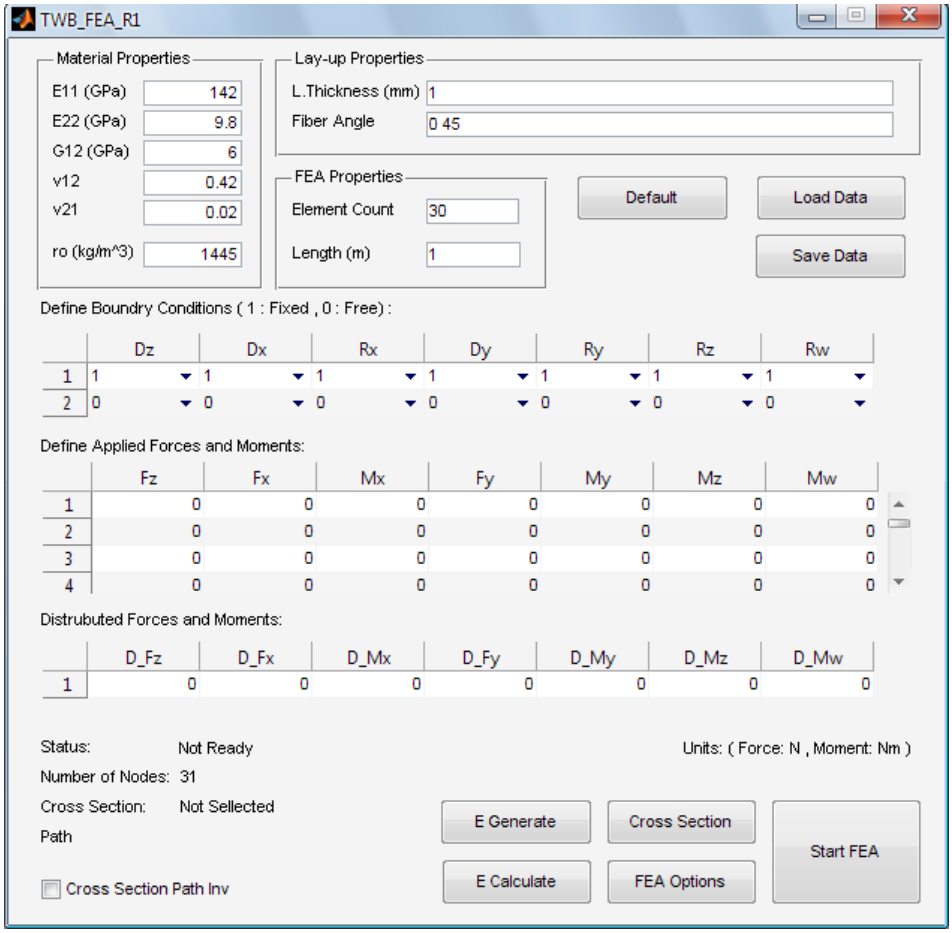
Programın akış şeması Şekil 4.1a ve 4.1b`de, kullanıcı arayüzü Şekil 4.2 ve 4.3`de gösterilmiştir. Daha detaylı bir akış şeması Ek-A`de, yazılımın kodları ise Ek-B`de verilmiştir.



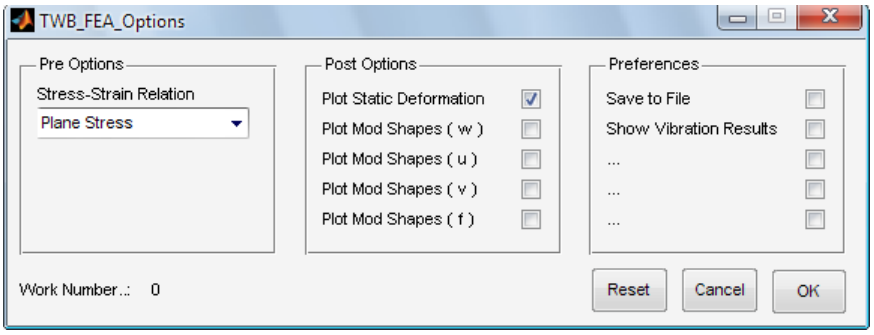
Şekil 4.1.a Akış Şeması



Şekil 4.1.b Akış Şeması



Şekil 4.2 Ana Arayüz



Şekil 4.3 Ayarlar Arayüzü

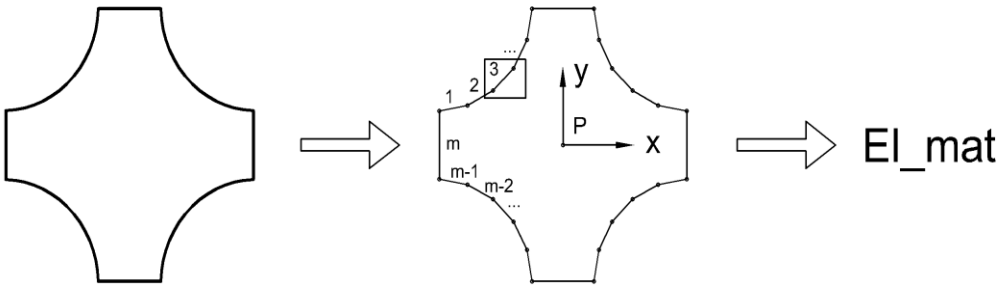
Programanın iki arayüzü vardır. Şekil 4.2 de verilen ana arayüz; malzeme özellikleri (Material Properties), katman özellikleri (Lay-up Properties), sonlu eleman özellikleri (FE Properties), sınır koşulları (Boundary Conditions), yükleme koşulları (Applied Forces and Moments) ve yayılı yükler (Distributed Forces and Moments)

olmak üzere beş alt bölüme ayrılmıştır. Malzemeyi ve problemi tanımlamak için ilgili değerler bu bölümler kullanılarak girilir.

Programın ikinci arayüzü ise Şekil 4.3`de gösterilen ayarlar arayüzüdür, bu arayüze ana arayüzdeki *FEA Options* butonuna basılarak geçilir. Bu arayüz ile analizde kullanılacak gerilme-genleme ilişkisi belirlenir ve analiz sonucu alınmak istenen çıktılar seçilir.

Kiriş kesiti ise daha önce hazırlanmış bir DXF dosyası ile programa tanıtılır. DXF dosyası teknik resimde kullanılan doğru, eğri, daire, yay gibi çizim elemanlarının (entity) geometrik tanımlarını barındıran bir teknik resim dosya formatıdır. Ek-C de çalışmada kullanılan bir kesitin teknik resmi ve bu resmi tanımlayan DXF dosyasının *entities* bölümü örnek olarak verilmiştir. Kiriş kesitini seçmek için ana arayüzdeki *Cross Section* butonuna basılır ve açılan dosya yükleyicisi ile ilgili DXF dosyası seçilir.

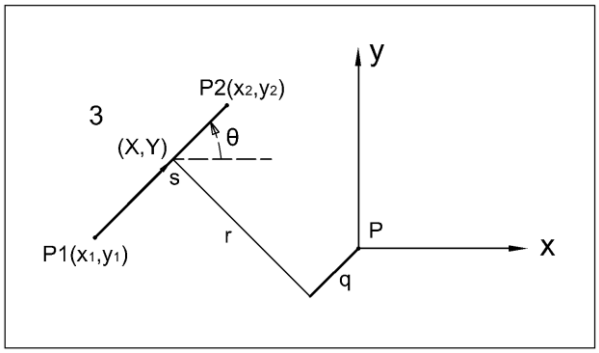
Gerekli olan bütün değerler girildikten ve kiriş kesiti seçildikten sonra ana arayüzdeki *Start FEA* butonuna basılarak hesaplamalar başlatılır. Program ilk etepta DXF dosyasından doğru parçası (line), yay (arc) ve daire (circle) çizim elemanlarını okur ve çiziliş sırasına göre numaralandırır. Çizim elemanı bir doğru parçası (line) ise program bu doğru parçasının başlangıç ve bitiş noktasının koordinatlarını cn_Line adlı bir matrise kaydeder. Çizim elemanı bir yay (arc) ise hesaplamaları kolaylaştırmak için Şekil 4.4 deki gibi yay küçük doğru parçalarına ayrılır ve bu doğru parçalarının başlangıç ve bitiş noktalarının koordinatları cn_Arc matrisine sıralı olarak kaydedilir. Çizim elemanı bir daire (circle) ise yaydakine benzer şekilde daire küçük doğru parçalarına ayrılır ve bu doğru parçalarının başlangıç ve bitiş noktalarının koordinatları cn_Cir adlı bir matrise sıralı olarak kaydedilir.



Şekil 4.4 Kesitten Dataların Toplanması

Çizim elemanlarının okunması bittikten sonra cn_Line ve cn_Arc matrislerindeki değerler çizim sırasına göre El_mat matrisine aktarılır. Eğer teknik resimde bir daire elemanı varsa cn_Cir matrisindeki çizilen ilk daireye ait değerler El_mat matrisine aktarılır, diğer çizim elemanlarına ait değerler ihmal edilir. Bu işlemler sonucu kiriş kesiti küçüklü büyüklü doğru parçaları şeklinde yeniden tanımlanmış olur ve bu doğru parçalarına ait değerler sıralı bir şekilde El_mat matrisinde saklanır.

Ardından her bir doğru parçası için Şekil 4.4 de gösterilen X, Y, Q, R, W ve S değerleri hesaplanır. Bu değerler kullanılarak her bir doğru parçası için (3.46) nolu denklemlerle tanımlanan Eij sertlik matrisi değerleri bulunur. Kiriş kesitinin tümünü tanımlayan Eij sertlik matrisi, her bir doğru parçası için bulunan Eij sertlik matrisi değerlerin toplamına eşittir.



Şekil 4.5 Doğru Parçası

Eij sertlik matrisi bulunduktan sonra (3.55) ve (3.56) nolu denklemler kullanılarak Kij eleman sertlik matrisi ve Mij eleman kütle matrisi hesaplanır.

El_mat ⟹ X, Y, Q, R, W , S ⟹ [Eij] ⟹ [Kij]

Şekil 4.6 Dataların İşlenmesi

Bundan sonra sınır şartlarına göre {U} yerdeğiştirme vektörü ve {F} kuvvet vektörü tanımlanır.

$$\begin{bmatrix} K11 & K12 & K13 & \cdots & K1n \\ K21 & K22 & K23 & \cdots & K2n \\ K31 & K32 & K33 & \cdots & K3n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ Kn1 & Kn2 & Kn3 & \cdots & Knn \end{bmatrix} \cdot \begin{Bmatrix} 0 \\ U2 \\ 0 \\ \vdots \\ Un \end{Bmatrix} = \begin{Bmatrix} F1 \\ F2 \\ F3 \\ \vdots \\ Fn \end{Bmatrix} \tag{4.1}$$

Yer değiştirme vektöründe sabit olan elemanlar sıfır olarak yazılır ve bu elemanlara karşılık gelen satır ve sütunlar silinerek sistem çözüm yapılabilecek hale indirgenir.

$$\begin{bmatrix} K22 & K24 & . & K2n \\ K42 & K44 & . & K4n \\ . & . & . & . \\ Kn2 & Kn4 & . & Knn \end{bmatrix}_s . \begin{Bmatrix} U2 \\ U4 \\ . \\ Un \end{Bmatrix}_s = \begin{Bmatrix} F2 \\ F4 \\ . \\ Fn \end{Bmatrix}_s \tag{4.2}$$

Böylece {U}s yerdeğiştirme vektöründe sadece bilinmeyenler kalır, bunandan sonra statik analiz için {U}s yerdeğiştirme vektörü çekilerek çözüm yapılabilir.

$$\{U\}_s = [K]_s^{-1} . \{F\}_s \tag{4.3}$$

Titreşim analizi için ise (4.2) nolu ifadedekine benzer şekilde [M] eleman kütle matrisi de sadeleştirilir ve sitemin öz değerleri ve öz vektörleri aşağıdaki Matlab kodu ile hesaplanır;

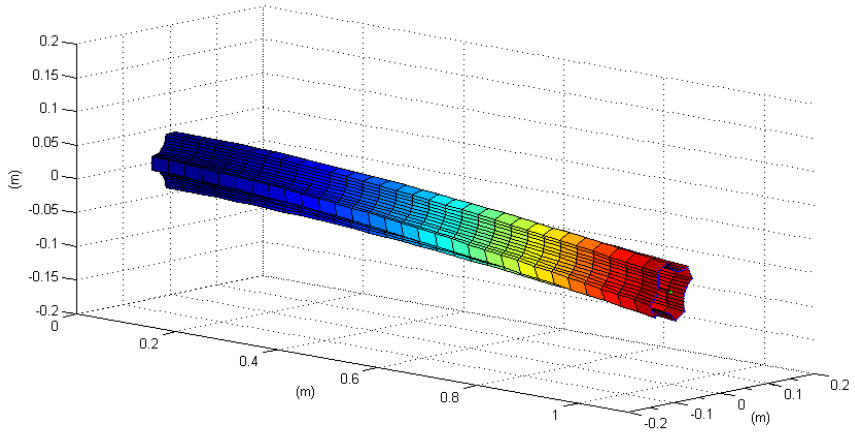
$$[Vs, Ds] = eig(Ks, Ms) \tag{4.4}$$

Burada Ds öz değerleri, Vs ise öz vektörleri içerir. Sistemin doğal frekansları öz değerlerin karekökü alınarak, bu frekanslara karşılık gelen titreşim modları ise öz vektörler normalize edilerek bulunur.

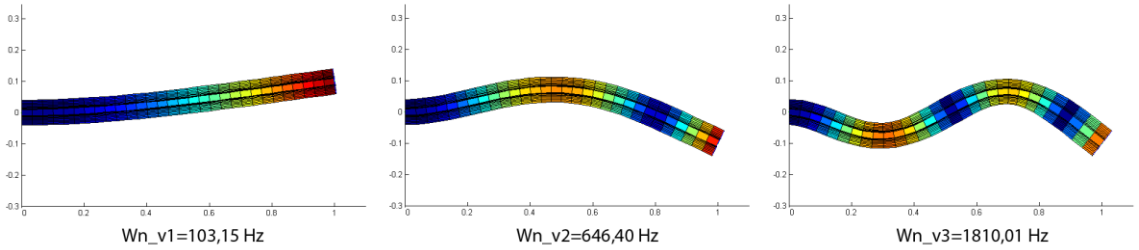
$$Wn = \sqrt{Dn} \tag{4.5}$$

Sonlu elemanlar çözümü yapıldıktan sonra program kutup noktasına ait U, V, W ve ϕ yer değiştirmelerini ayrı ayrı matrislere kaydeder ve bu matrisler üzerinden üç boyutlu deformasyon çıktısını yayımlar. İstenirse U, V, W ve ϕ matrisleri kullanılarak daha detaylı deformasyon çıktıları alınabilir. Benzer şekilde titreşim analizine ait sonuçlar da ayrı ayrı matrislere kaydedilir ve istenilen çıktılar bu matrisler kullanılarak alınabilir.

Örnek olarak şekil 4.7`de statik deformasyon çıktısı ve şekil 4.8`de aynı kirişe ait serbest titreşim analizi çıktıları gösterilmiştir. Bu örnekte kullanılan malzeme özellikleri, yükleme koşulları ve kiriş kesiti Ek-D`de verilmiştir.



Şekil 4.7 Statik Deformasyon Çıktı Örneği



Şekil 4.8 Serbest Titreşim Çıktı Örneği

Kirişin herhangi bir noktasındaki gerilme hali U,V,W ve ϕ yer değişimleri cinsinden hesaplanabilir. Fakat kullanılan kompozit malzemenin hasarlanmadığı ve sürekli olarak lineer elastik davranış gösterdiği kabul edildiği için bulunan gerilme sonuçları belli bir yer değişimi sonrası doğruluk arz etmeyecektir.

BÖLÜM 5

ANALİZLER

Bu bölümde yapılan çalışmanın doğruluğunu sınamak için bulunan sonuçlar, bu konuda yapılmış diğer çalışmalarla, Ansys ve Patran sonuçlarıyla karşılaştırılmıştır. Bu çalışmada bulunan sonuçlar geliştirilen yazılımın ismi TWB-FEA ile gösterilmiştir.

5.1. Doğrulama ve Karşılaştırmalar

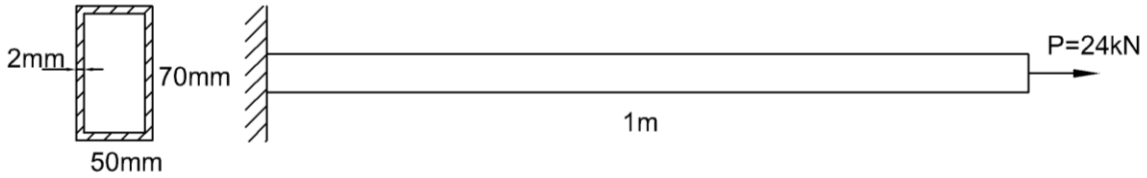
Geliştirilen sonlu elemanlar yazılımının doğruluğunu sınamak için çeşitli yükleme koşulları altında bulunan kutu, I ve çeşitli kesitlere sahip ince cidarlı kompozit kirişler incelenmiş ve bulunan sonuçlar bu konuda yapılmış diğer çalışmalarla, Ansys ve Patran sonuçlarıyla karşılaştırılmıştır. Tablo-1`de bu çalışmada ve daha önceki çalışmalarda kullanılan malzemelerin mekanik özellikleri verilmiştir.

Tablo - 1 Malzeme Özellikleri

Malzeme	E_{11} (GPa)	E_{22} (GPa)	G_{12} (GPa)	v_{12}	ro (kg/m^3)
A [7,11,17,18]	148	9,65	4,55	0,34	1445
B [12,15,16,19]	142	9,80	6,00	0,42	1445

Kutu Kirişin Eksenel Yük Altındaki Deformasyonu:

Boyutları 50mm x 70mm, cidar kalınlığı 2mm ve uzunluğu 1m olan çeşitli katman dizilimlerine sahip kutu kirişlere Şekil 5.1`de gösterildiği gibi eksenel 24kN`luk çekme kuvveti uygulanmıştır. Kutu kirişlerin bir ucu ankastre sabitlenmiştir, kuvvet uygulanan ucu ise serbesttir. Kutu kirişlerde Tablo-1`de gösterilen A malzemesi kullanılmıştır. Tablo-2`de bu yükleme koşulu altında çeşitli katman dizilimlerine sahip kutu kirişlerin serbest ucunda oluşan eksenel yerdeğiştirme ve eksenel dönme değerleri verilmiştir.



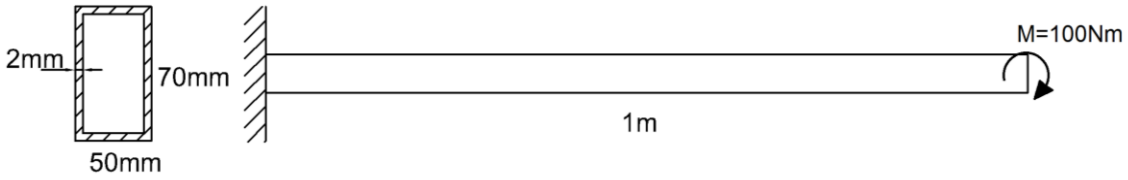
Şekil 5.1 Eksenel Yük Altındaki Kutu Kiriş

Tablo -2 Eksenel Yük Altındaki Kutu Kirişin Eksenel Yerdeğiştirmesi ve Eksenel Dönmesi

Katman Dizilimi	Referans	w (m x 10^{-3})	ϕ(rad x 10^{-3})
$[0_{10}/45_{10}]$	Kollar & Pluzsik [7]	0,616	9,95
	Ansys CPP	0,625	15,50
	Patran	0,620	10,21
	Vo & Lee* [11]	0,620	9,11
	Vo & Lee**[11]	0,603	16,40
* Düzlem Gerilme (σ_3=0)	TWB-FEA*	0,620	9,06
** Düzlem Genleme (ε_3=0)	TWB-FEA**	0,599	16,98
$[0_{10}/0_{10}]$	Kollar & Pluzsik [7]	0,338	0,00
	Ansys CPP	0,338	0,00
	Patran	0,338	0,00
	TWB-FEA	0,338	0,00
$[45_{10}/-45_{10}]$	Kollar & Pluzsik [7]	3,052	0,00
	Ansys CPP	3,053	0,00
	Patran	3,018	0,00
	TWB-FEA	3,051	0,00
$[45_{10}/45_{10}]$	Kollar & Pluzsik [7]	4,069	82,50
	Ansys CPP	4,064	82,19
	Patran	4,064	82,20
	TWB-FEA	4,064	82,40
$[10_6/20_6/30_6]$	Kollar & Pluzsik [7]	1,049	64,70
	Ansys CPP	1,051	67,23
	Patran	1,037	63,95
	TWB-FEA	1,067	66,90

Kutu Kirişin Moment Etkisindeki Deformasyonu:

Boyutları 50mm x 70mm, cidar kalınlığı 2mm ve uzunluğu 1m olan bir kutu kirişe Şekil 5.2`de gösterildiği gibi 100Mn`lik moment uygulanmıştır. Kutu kirişin bir ucu ankastre sabitlenmiştir, kuvvet uygulanan ucu ise serbesttir. Kutu kirişte Tablo-1`de gösterilen A malzemesi kullanılmıştır ve kirişin katman dizilimi [0/45] şeklindedir. Tablo-3`de bu yükleme koşulu altında kutu kirişin serbest ucunda oluşan yatay yerdeğiştirme (v) değerleri verilmiştir.



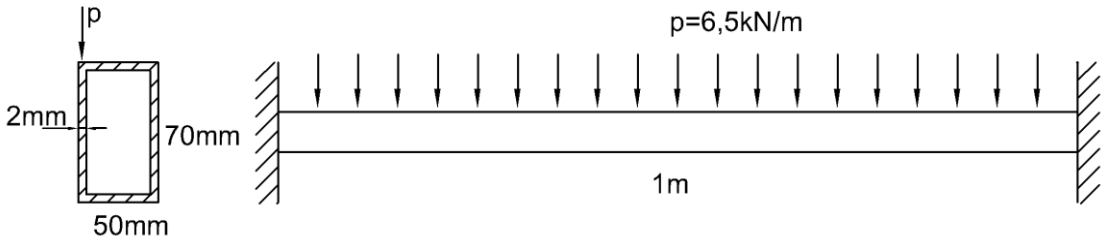
Şekil 5.2 Moment Etkisindeki Kutu Kiriş

Tablo - 3 Moment Etkisindeki Kutu Kirişin Düşey Yerdeğiştirmesi

Katman Dizilimi	Referans	v (m x 10^{-3})
$[0_{10}/45_{10}]$	Kollar & Springer [18]	1,695
	Ansys CPP	1,761
	Patran	1,758
	TWB-FEA	1,742

Kutu Kirişin Eksantrik Yayılı Yük Altındaki Deformasyonu:

Boyutları 50mm x 70mm, cidar kalınlığı 2mm ve uzunluğu 1m olan bir kutu kirişe Şekil 5.3`de gösterildiği gibi 6,5kN/m`lik eksantrik yayılı yük uygulanmıştır. Yayılı yükün kuvvet doğrultusu kutu kirişin 70mm`lik bir kenarının ortasından geçecek şekildedir. Kutu kirişin iki ucu da ankastre sabitlenmiştir. Kutu kirişte Tablo-1`de gösterilen A malzemesi kullanılmıştır ve kirişin katman dizilimi $[[45/\overline{}45]_2/0_{12}/[45/\overline{}45]_2]$ şeklindedir. Tablo-4`de bu yükleme koşulu altında kutu kirişinde oluşan maksimum yatay yerdeğiştirme (v) ve eksenel dönme (ϕ) değerleri verilmiştir.



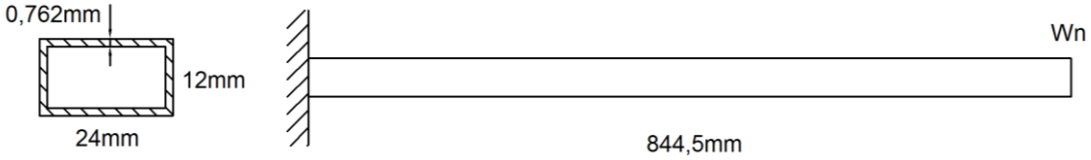
Şekil 5.3 Eksantrik Yayılı Yük Altındaki Kutu Kiriş

Tablo -4 Yayılı Yük Altındaki Kutu Kirişin Düşey Yerdeğiştirmesi ve Eksenel Dönmesi

Katman Dizilimi	Referans	v (m x 10^{-4})	ϕ(rad x 10^{-3})
[[45/-45] $_2$/0$_{12}$/[45/-45] $_2$]	Sheikh & Thomsen* [17]	4,940	6,70
	Sheikh & Thomsen**[17]	4,378	2,75
	Vo & Lee* [11]	4,940	6,43
	Vo & Lee**[11]	4,380	2,68
	Kollar & Springer [18]	4,880	2,76
	Ansys CPP	6,658	3,72
	Patran	6,545	3,25
* Düzlem Gerilme (σ_3=0)	TWB-FEA*	4,943	6,24
** Düzlem Genleme (ε_3=0)	TWB-FEA**	4,371	2,62

Bir Ucu Ankastre Sabitlenmiş Kutu Kirişin Serbest Titreşimi:

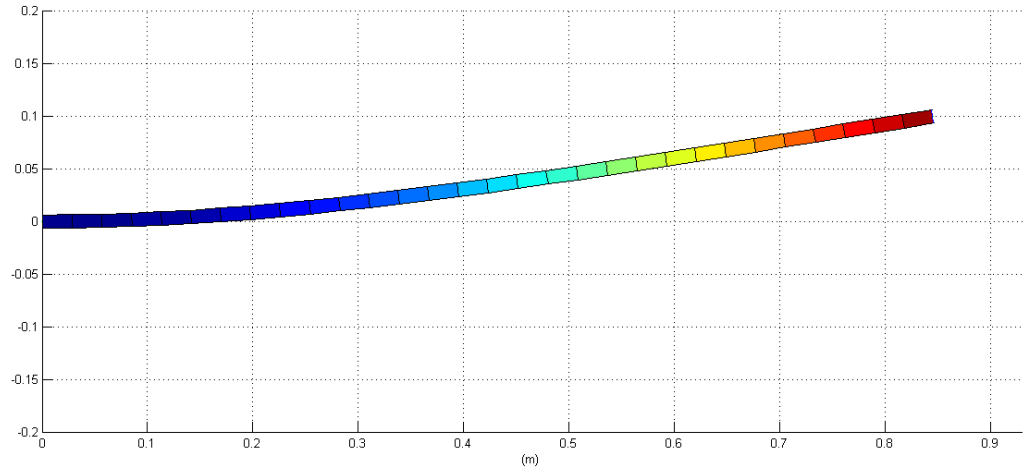
Boyutları 24mm x 12mm, cidar kalınlığı 0,762mm ve uzunluğu 0,845m olan çeşitli katman dizilimlerine sahip kutu kirişlerin serbest titreşim analizi yapılmıştır. Şekil 5.4`de gösterildiği kutu kirişlerin bir ucu ankastre sabitlenmiş diğer ucu serbesttir. Kutu kirişlerde Tablo-1`de gösterilen B malzemesi kullanılmıştır. Tablo-5`de çeşitli katman dizilimlerine sahip kutu kirişlerin y eksenindeki birinci doğal frekansları (Wn-v1) verilmiştir. Şekil 5.4`de $[0/45]_3$ katmandizilimine sahip kutu kirişin y eksenindeki ilk üç doğal frekansı ve mod şekilleri verilmiştir.



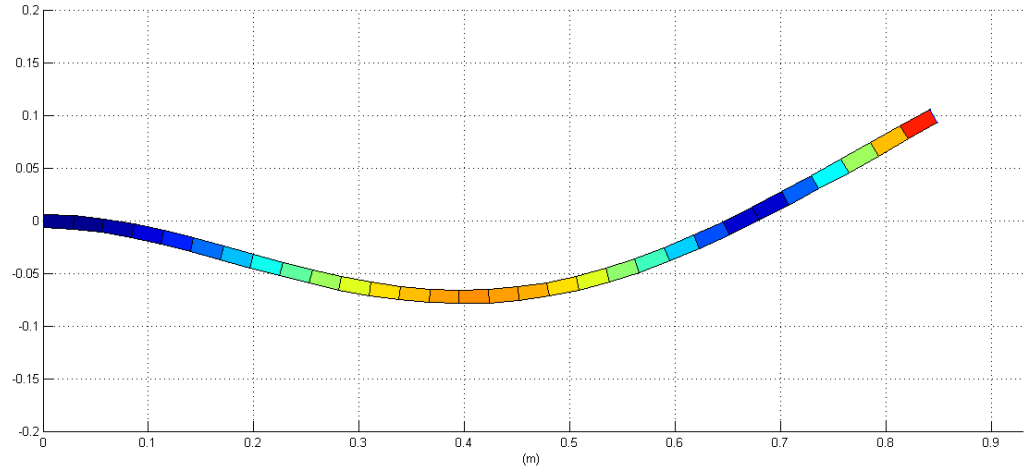
Şekil 5.4 Bir Ucu Ankastre Sabitlenmiş Kutu Kiriş

Tablo - 5 Kutu Kirişin Serbest Titreşim Halindeki Doğal Frekansları

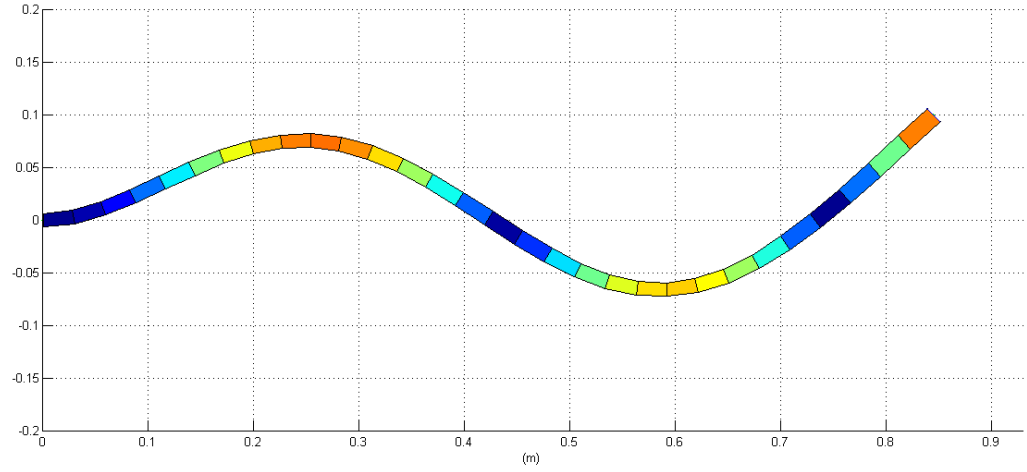
Katman Dizilimi	Referans	Wn-v_1 (Hz)
$[15/15]_3$	Chandra & Chopra [15]	28,66
	Qin & Librescu [16]	30,06
	Vo & Lee [12]	38,65
	Günay & Tımarcı [19]	28,37
	Ansys CPP	28,28
	Patran	29,04
	TWB-FEA	39,17
$[0/30]_3$	Chandra & Chopra [15]	30,66
	Qin & Librescu [16]	34,58
	Vo & Lee [12]	35,53
	Günay & Tımarcı [19]	34,29
	Ansys CPP	34,36
	Patran	34,73
	TWB-FEA	35,60
$[0/45]_3$	Chandra & Chopra [15]	30,00
	Qin & Librescu [16]	32,64
	Vo & Lee [12]	32,52
	Günay & Tımarcı [19]	32,35
	Ansys CPP	31,98
	Patran	33,11
	TWB-FEA	32,36



Wn(v1) 32.36 Hz



Wn(v2) 202.82 Hz

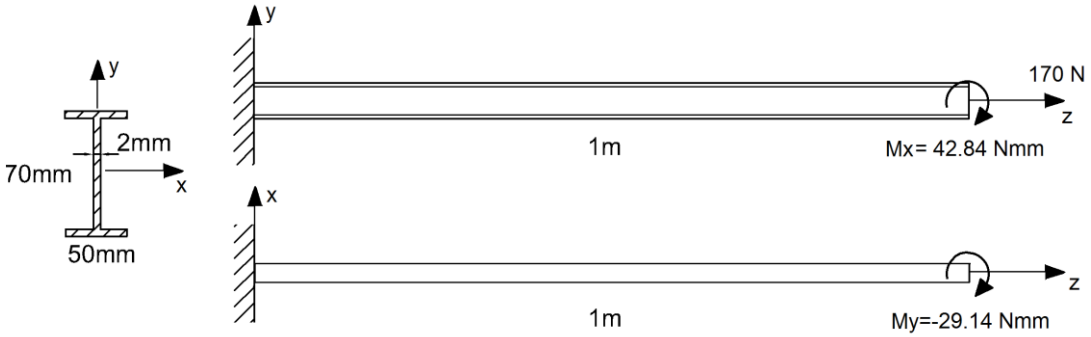


Wn(v3) 567.90

Şekil 5.5 Katman Dizilimi $[0/45]_3$ Olan Kutu Kirişin Doğal Frekansları ve Mod Şekilleri

I-Kirişin Çeşitli Yükler Altındaki Deformasyonu:

Boyutları 50mm x 70mm, cidar kalınlığı 2mm ve uzunluğu 1m olan bir I kirişe Şekil 5.5`de gösterildiği gibi çekme kuvveti ve moment uygulanmıştır. I kirişin bir ucu ankastre sabitlenmiştir, yükleme yapılan diğer ucu ise serbesttir. I kirişte Tablo-1`de gösterilen A malzemesi kullanılmıştır ve kirişin katman dizilimi [0/45] şeklindedir. Tablo-6`da bu yükleme koşulu altında I kirişin serbest ucunda oluşan yerdeğiştirme ve eksenel dönme değerleri verilmiştir.



Şekil 5.6 Çeşitli Yükler Altındaki I Kiriş

Tablo-6 I-Kiriş Yer Değiştirmeler

Referanslar	w(m x 10^{-2})	u (m x 10^{-2})	v (m x 10^{-2})	ϕ(rad x 10^{-3})
Kollar & Pluzsik [7]	**0,6289**	**-0,4325**	**0,8920**	**0,1009**
Ansys	**0,6216**	**-0,4395**	**0,8662**	**0,1046**
Patran	**0,6237**	**-0,4418**	**0,8726**	**0,1079**
TWB-FEA	**0,6121**	**-0,5829**	**0,8903**	**0,1223**

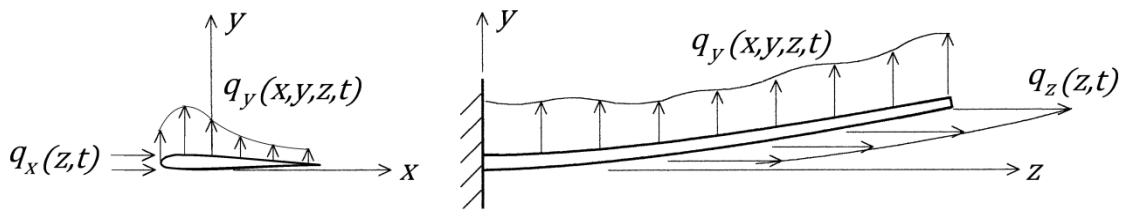
Yukarıda verilen tablolarda görüldüğü üzere dikdörtgen ve I kesite sahip ince cidarlı kompozit kirişlerin geliştirilen yazılım ile bulunan statik ve dinamik analiz sonuçları daha önce yapılan çalışmalarla ve diğer sonlu elemanlar yazılımlarının sonuçları ile tutarlıdır.

5.2. Çeşitli Kesitlere Sahip Kirişlerin Analizleri ve Karşılaştırmalar

Aşağıda çeşitli kesitlere sahip kirişlerin analizleri yapılmış, bulunan sonuçlar Ansys ve Patran sonuçları ile karşılaştırılmıştır.

Bir Rüzgar Türbini Kanadının Analizi

İnce cidarlı kompozit kirişler çeşitli hava araçlarının ve rüzgâr türbinlerinin kanatlarında kullanılmaktadır. Aşağıda 5 metre uzunluğunda tek hücreli sabit kesitli bir türbin kanadının statik ve serbest titreşim analizi yapılmıştır. Kullanılan kanat kesiti NACA (Amerikan Ulusal Havacılık Danışma Kurulu) kütüphanesinden alınmıştır. Kullanılan kompozit malzemenin özellikleri Tablo-1b`de, yükleme hali Şekil 5.6`da ve kullanılan kiriş kesiti EK-A`da verilmiştir. Her bir katman 1mm kalınlığında ve katman dizilimi [0/45/-45/0] şeklindedir.



Şekil 5.7 Yayılı Yükler Altındaki Rüzgar Türbin Kanadı
q_x=2 kN/m, q_y=10 kN/m, q_z=3,5 kN/m

Tablo-7 NACA-63-415 Statik Analiz

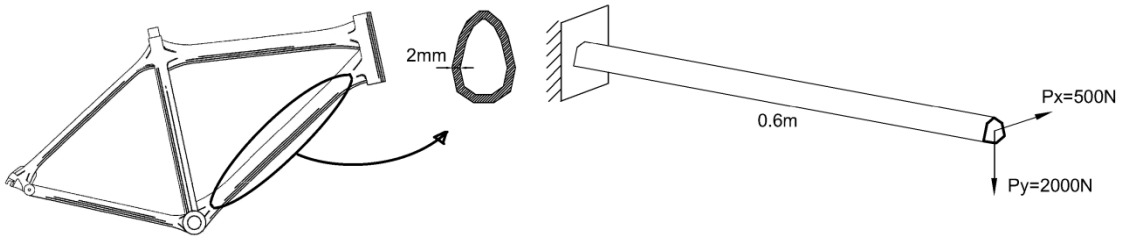
Referanslar	w(m x 10^{-5})	u (m x 10^{-4})	v (m x 10^{-2})	ϕ(rad x 10^{-3})
Ansys	1,396	2,496	9,842	-3,740
Patran	1,408	2,536	9,862	-3,971
TWB-FEA	1,415	2,982	9,670	-4,583

Tablo-8 NACA-63-415 Serbest Titreşim Analizi

Referanslar	Wn-u1 (Hz)	Wn-v1 (Hz)	Wn- ϕ1 (Hz)
Ansys	8,124	46,819	28,659
Patran	8,362	47,013	27,965
TWB-FEA	8,557	47,161	26,489

Bir Kompozit Bisiklet Çerçevesi Kirişinin Analizi

İnce cidarlı kompozit kirişlerin bir diğer kullanım alanı da bisiklet ve motosiklet çerçevesi yapımıdır. Bisiklet ve motosikletlerde kullanılan kompozit kirişler genelde tek hücrelidir bazı tasarımlarda kiriş kesiti değişkendir. Aşağıda bir kompozit bisiklet çerçevesinde kullanılan sabit kesitli 0,6m boyundaki bir ince cidarlı kompozit kirişin analizi yapılmıştır. Kullanılan malzeme özellikleri Tablo-1b`de, yükleme hali Şekil 5.7`de ve kullanılan kiriş kesiti EK-A`da verilmiştir. Kiriş her biri 0,5mm kalındığında olan katmanlardan hazırlanmıştır, katman dizilimi [0/45/-45/0] şeklindedir.



Şekil 5.8 Kompozit Bisiklet Çerçevesi Alt Kirişi

Tablo-9 Bisiklet Alt Kirişi Statik Analiz Sonuçları

	u (m x 10^{-2})	v (m x 10^{-2})	ϕ (rad x 10^{-5})
Ansys	**3,195**	**-6,781**	**1,583**
Patran	**3,194**	**-6,791**	**1,567**
TWB-FEA	**3,193**	**-6,792**	**1,603**

Tablo-10 Bisiklet Alt Kirişi Serbest Titreşim Analizi Sonuçları

	Wn-u1 (Hz)	Wn-v1 (Hz)
Ansys	**100,191**	**137,193**
Patran	**100,153**	**137,167**
TWB-FEA	**100,116**	**137,284**

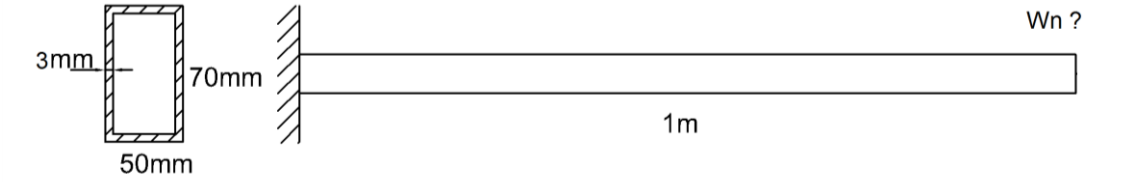
Yukarıda verilen tablolarda görüldüğü üzere çeşitli kesitere sahip ince cidarlı kompozit kirişlerin geliştirilen yazılımla bulunan statik ve dinamik analiz sonuçları diğer sonlu elemanlar yazılımlarının sonuçları ile tutarlıdır. Buda geliştirilen yazılımın karmaşık kesitlerde de düzgün çalıştığını göstermektedir.

5.3. Doğal Frekansların Elyaf Açılarına Bağlı Olarak Değişimi

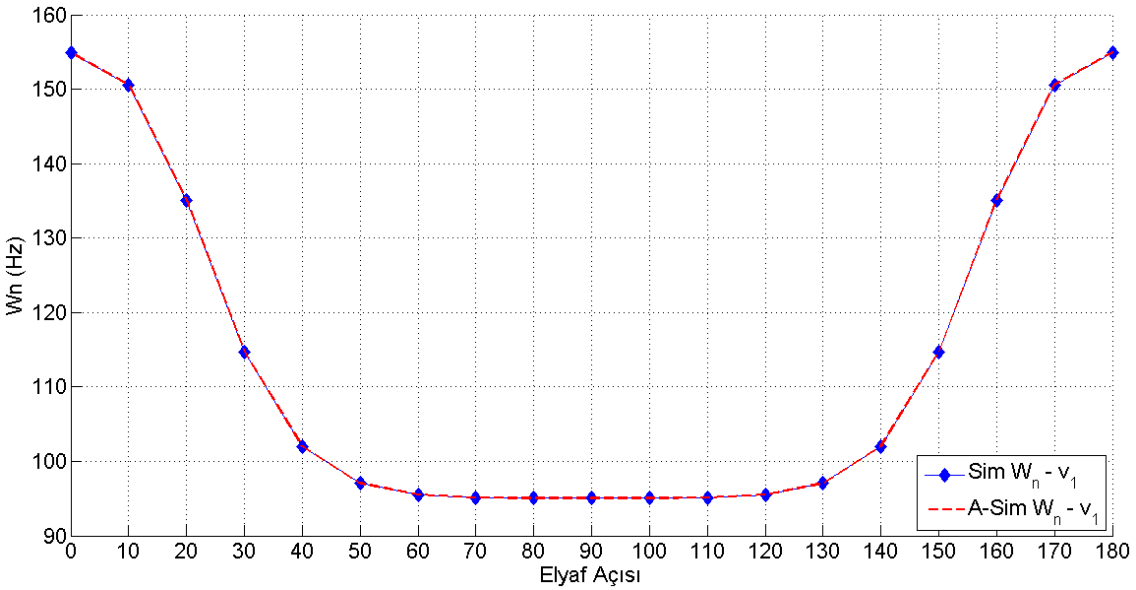
Aşağıda simetrik [α/0/α] ve antisimetrik [α/0/-α] laminatlar kullanılarak tasarlanmış, çeşitli kesitlere sahip ince cidarlı kompozit kirişlerin doğal frekanslarının elyaf açısına (α) bağlı olarak değişimi verilmiştir. Kirişlerde Tablo-1`de verilen A malzemesi kullanılmıştır.

Kutu Kiriş

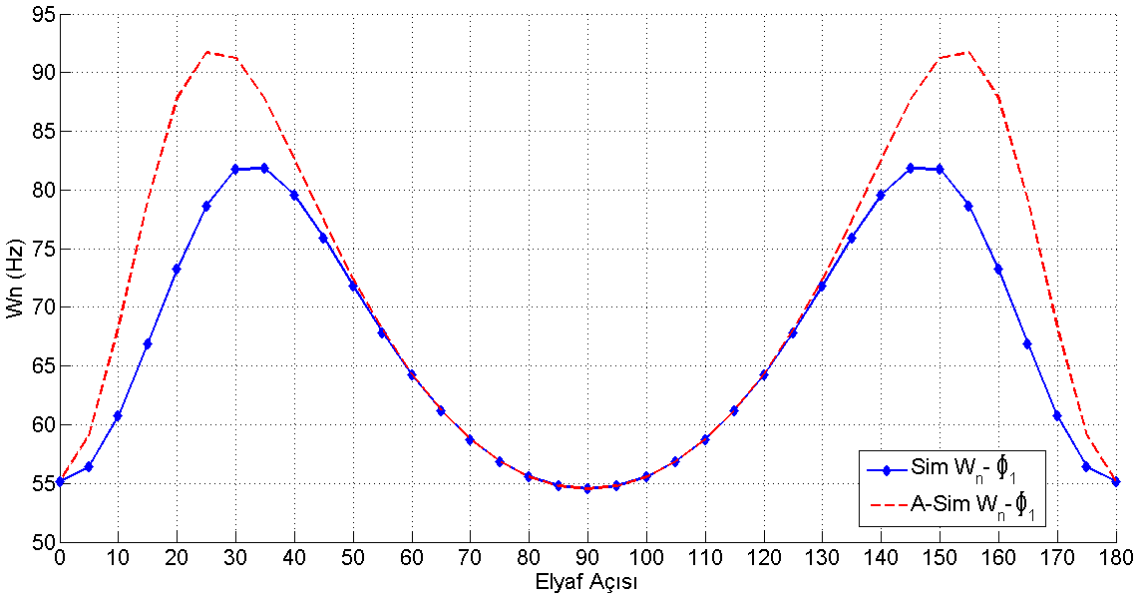
Aşağıda boyutları 50mm x 70mm, cidar kalınlığı 3mm ve uzunluğu 1m olan bir kutu kirişin elyaf açısına bağlı olarak doğal frakansının değişimi verilmiştir. Kiriş şekil-5.9`da görüldüğü gibi bir ucundan ankastre mesnetlenmiştir. Şekil-5.10`da elyaf açısına bağlı olarak birinci düşey doğal frekansının (Wn-v1) değişimi verilmiştir, burada simetrik ve antisimetrik laminatlar için neredeyse aynı sonuçlar bulunmuştur. Şekil-5.11`de ise elyaf açısına bağlı olarak birinci eksenel dönme doğal frekansının (Wn- ϕ1) değişimi verilmiştir, burada ise antisimetrik laminatın Wn- ϕ1`in 5^o-45^o ve 135^o -175^o aralıklarında daha büyük çıktığı görülmektedir.



Şekil 5.9 Kutu Kiriş



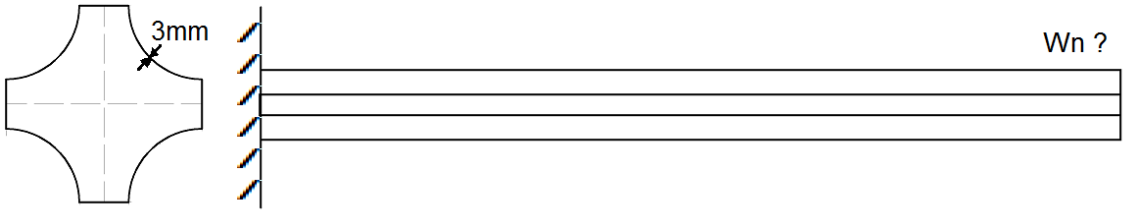
Şekil 5.10 Kutu Kiriş İçin Elyaf Açısına Bağlı Olarak Wn-v1`in Değişimi



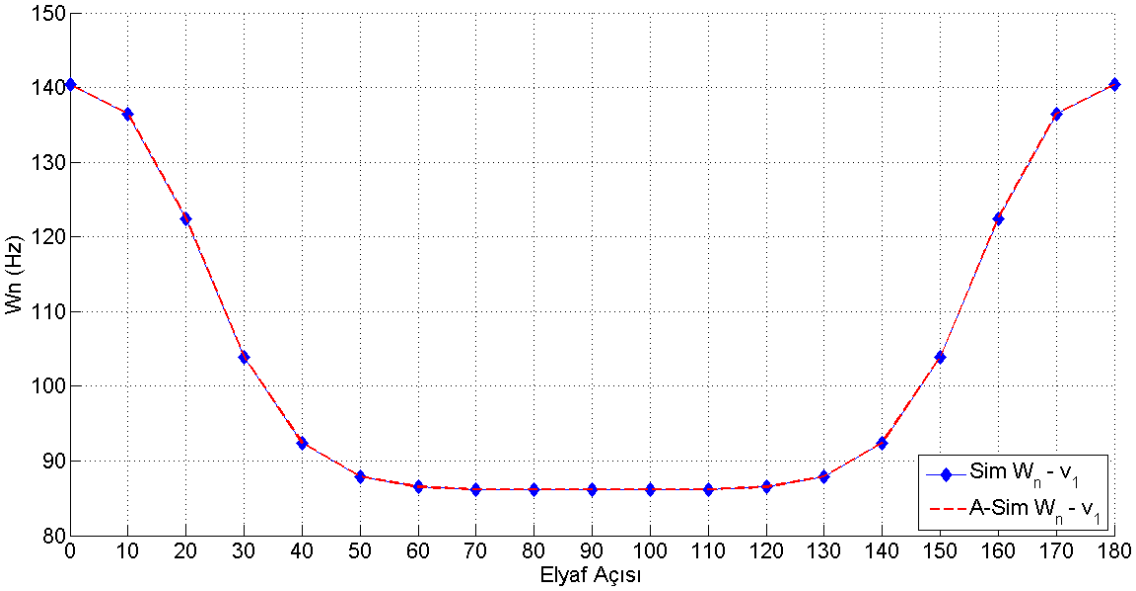
Şekil 5.11 Kutu Kiriş İçin Elyaf Açısına Bağlı Olarak Wn- ϕ1`in Değişimi

Özel Kiriş

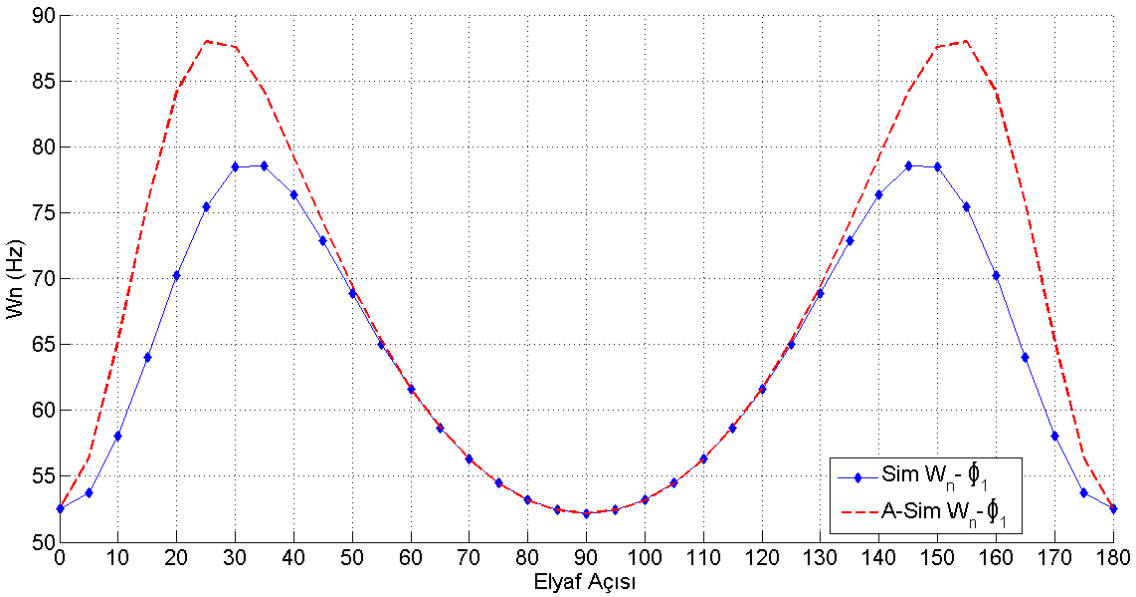
Aşağıda kesit boyutları Ek-C`de verilen 3mm cidar kalınlığında ve 1m uzunluğundaki özel bir kirişin elyaf açısına bağlı olarak doğal frakansının değişimi verilmiştir. Kiriş şekil-5.12`da görüldüğü gibi bir ucundan ankastre mesnetlenmiştir. Şekil-5.13`da elyaf açısına bağlı olarak Wn-v1`in değişimi verilmiştir, burada kutu kirişe benzer şekilde simetrik ve asimetrik laminatlar için neredeyse aynı sonuçlar bulunmuştur. Şekil-5.14`de ise elyaf açısına bağlı olarak birinci eksenel dönme doğal frekansının (Wn- ϕ1) değişimi verilmiştir, burada asimetrik laminatın Wn- ϕ1`in yine kutu kirişe benzer şekilde 5^o-45^o ve 135^o -175^o aralıklarında daha büyük çıktığı görülmektedir.



Şekil 5.12 Özel Kiriş



Şekil 5.13 Özel Kiriş İçin Elyaf Açısına Bağlı Olarak Wn-v1`in Değişimi



Şekil 5.14 Özel Kiriş İçin Elyaf Açısına Bağlı Olarak Wn- ϕ1`in Değişimi

BÖLÜM 6

SONUÇ VE YORUM

Bu çalışmada tek hücreli açık veya kapalı herhangi bir kesite sahip ince cidarlı kompozit kirişlerin statik ve dinamik analizini için yapmak için bir sonlu elemanlar yazılımı geliştirilmiştir. Yazılımın sonuçları bu konuda yapılan diğer çalışmalarla ve yazılımlarla karşılaştırılmıştır. Bölüm 5`de görüldüğü üzere sonuçlar diğer çalışmalarla ve yazılımlarla tutarlıdır.

Günümüzde uygulamada kullanılan ince cidarlı kompozit kirişlerin çoğu çok hücrelidir veya değişken kesitlere sahiptir. Bazı uygulamalarda ise kiriş ekseni doğrusal değildir. Bazı kirişlerde ise kesitin farklı yerlerinde değişik malzemeler kullanılmaktadır. Yapılan bu çalışma ve geliştirilen yazılım günümüz uygulamalarının ihtiyaçlarını karşılayacak şekilde geliştirilerek daha karmaşık şekil ve yapıdaki kompozit elemanların çözümlerini yapabilecek hale getirilebilir.

Yazılım ilk etapta Matlab platformu kullanılarak hazırlanmıştır, kodların Python platformuna uyumlu hale dönüştürülerek açık kaynak kodlu olarak yayınlanması ve ortak gelişime açılması planlanmaktadır.

Yazılımın ileri sürümlerinde hesaplamalarla ilgili eklenmesi düşünülen özellikler aşağıda verilmiştir;

- Çok hücreli kirişlerin analizi,
- Değişken kesitli kirişlerin analizi,
- Örülmüş elyaflı, kıyılmış elyaflı, sandviç gibi çeşitli yapıdaki kompozit malzemelerin tanımlanabilmesi ve analizi,
- Shell ve solid sonlu eleman tipleri üzerinden analiz,
- Hasar kriterlerinin belirlenmesi ve hasar analizi,
- Non-lineer özelliklerin dikkate alınması,
- Plastik davranışın dikkate alınması.

Yazılımın ileri sürümlerinde kullanım ile ilgili yapılması düşünülen yenilikler aşağıda verilmiştir;

- 2B/3B bir çizim ara yüzünün geliştirilmesi veya mevcut bir yazılımla köprü kurulması,
- Malzeme özelliklerinin, sınır şartlarının, yükleme koşullarının ve mesh özelliklerinin tanımlanabileceği 2B/3B bir grafik ara yüzün geliştirilmesi,
- Daha gelişmiş bir grafik çıktı ara yüzünün geliştirilmesi.

Yazılımın kod yapısı ve işleyişi ile ilgili yapılması düşünülen değişiklikler aşağıda verilmiştir;

- Yazılımın alt yazılımlara/kodlara ayrılarak daha modüler ve esnek bir yapı kazandırılması,
- Daha gelişmiş bir sonlu elemanlar çözücüsünün hazırlanması veya mevcut çözücülerle bağ kurulması.
- Diğer yazılımlarla data paylaşımı için giriş/çıkış noktalarının hazırlanması.

KAYNAKLAR

[1] Vlasov VZ. Thin-Walled Elastic Beams. 2nd Ed. Jerusalem, Israel: Israel Program For Scientific Translation; 1961.

[2] Gjelsvik A. The Theory of Thin-Walled Bars. New York: John Wiley and Sons Inc.; 1981.

[3] Chandra R, Stemple AD, Chopra I. Thin-Walled Composite Beams Under Bending, Torsional and Extensional Loads. Journal of Aircraft 1990; 27(7):619–36.

[4] Song O, Librescu L. Free Vibration of Anisotropic Composite Thin-Walled Beams of Closed Cross-Section Contour. Journal of Sound and Vibration 1993; 167(1):129–47.

[5] Puspita G, Barrau JJ, Gray D. Computation of Flexural and Torsional Homogeneous Properties and Stresses in Composite Beams with Orthotropic Phases. Composite Structures 1993; 24:43–9.

[6] Jeon SM, Cho MH, Lee I. Static and Dynamic Analysis of Composite Box Beams Using Large Deflection Theory. Computers and Structures 1995; 57(4):635–42.

[7] Kollar LP, Pluzsik A. Analysis of Thin-Walled Composite Beams with Arbitrary Layup. Journal of Reinforced Plastics and Composites 2002; 21(16):1423–65.

[8] Salim HA, Davalos JF. Torsion of Open and Closed Thin-Walled Laminated Composite Sections. Journal of Composite Materials 2005; 39(6):497–524.

[9] Cortinez VH, Piovan MT. Stability of Composite Thin-Walled Beams with Shear Deformability. Computers and Structures 2006; 84:978–90.

[10] Lee J, Lee S. Flexural–Torsional Behavior of Thin-Walled Composite Beams. Thin-Walled Structures 2004; 42:1293–305.

[11] Vo T.P, Lee J. Flexural–Torsional Behavior of Thin-Walled Closed-Section Composite Box Beams Engineering Structures 2007; 29:1774–1782.

[12] Vo T.P, Lee J. Free Vibration of Thin-Walled Composite Box Beams Composite Structures 2008; 84:11–20.

[13] Jones RM. Mechanics of Composite Materials. New York: Hemisphere Publishing Corp; 1975.

[14] Librescu L, Song O. Thin-Walled Composite Beams: Theory and Application (Solid Mechanics and Its Applications). 1st Ed. Springer; 2006.

[15] Chandra R, Chopra I. Experimental–Theoretical Investigation of The Vibration Characteristics of Rotating Composite Boxbeams. J Aircraft 1992; 29(4):657–64.

[16] Qin Z, Librescu L. On A Shear-Deformable Theory of Anisotropic Thinwalled Beams: Further Contribution and Validations. Compos Struct 2002; 56:345–58.

[17] Sheikh, Abdul Hamid; Thomsen, Ole Thybo. An Efficient Beam Element For The Analysis of Laminated Composite Beams of Thin-Walled Open and Closed Cross Sections. Composites Science and Technology 2008; 68(10):2273-2281.

[18] Kollar LP, Springer GS. Mechanics of Composite Structures, 1st Edition. Cambridge University Press; 2003.

[19] Günay M.G, Tımarcı T. Free Vibration of Copmpoite Box-Beams by Ansys. UNITECH-12, Gabrovo, Bulgaria, Proceedings of Int. Sci. Conference 2005;2,102-106.

[20] J. J. Connor, St. Venant Theory of Torsion-Flexure of Prismatic Members, Analysis of Structural Member System. Ronald Press Co; 1976.

[21] David V. Hutton. Fundamentals of Finite Element Analysis. Mcgraw-Hill Education Pvt Limited; 2004.

[22] BOERAEVE P. Introduction To The Finite Element Method. Institut Gramme – LIEGE; 2010.

[23] ANSYS Inc. Tutorials and Examples. ANSYS Composite Preppost V14.5 Documentation; 2013.

[24] MSC Software Co, Composite Materials Construction. Patran 2012 User`S Guide; 2012.

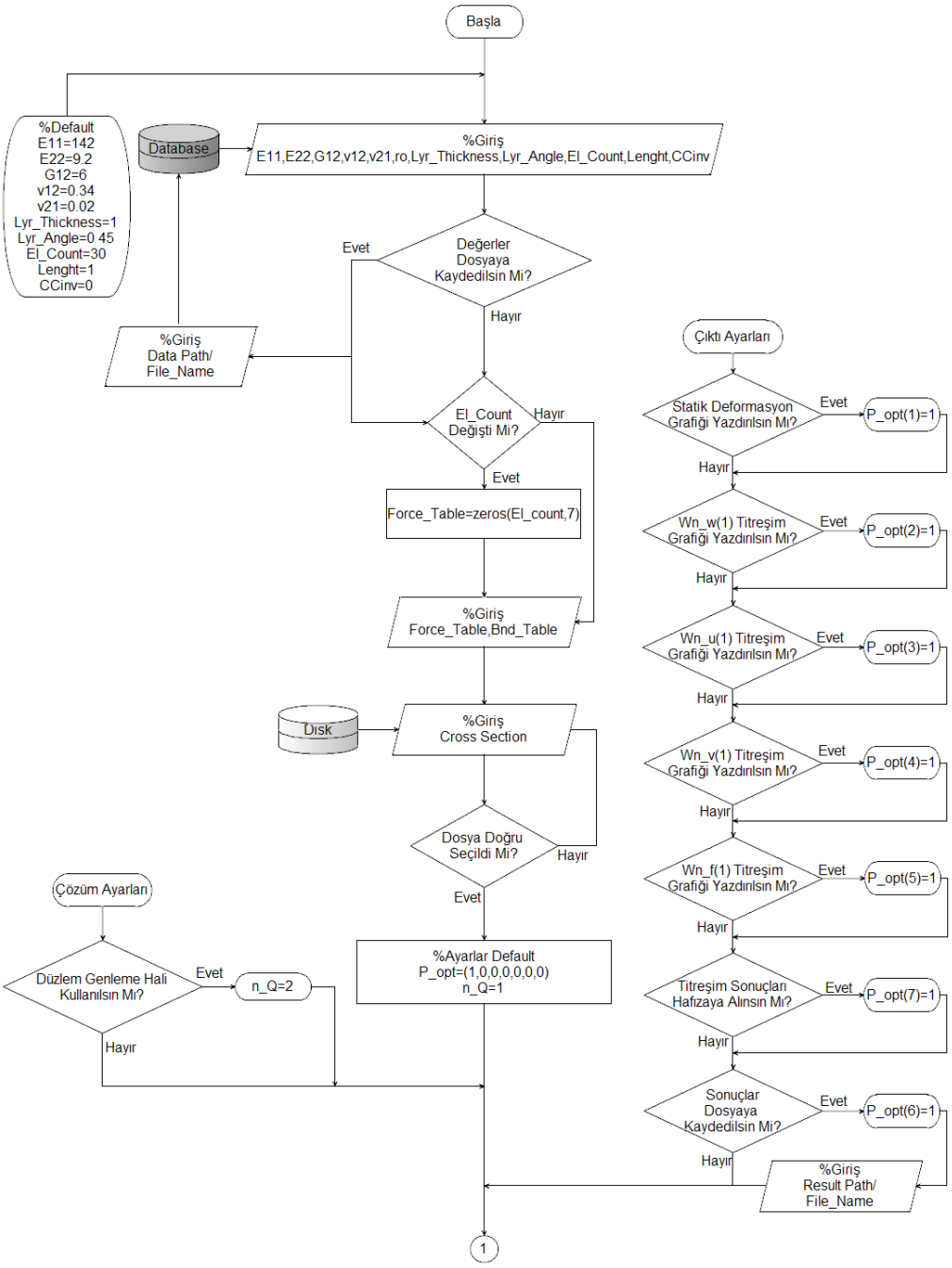
[25] Mathworks Inc. GUI Building Basics. Matlab R2012b Documentation; 2012.

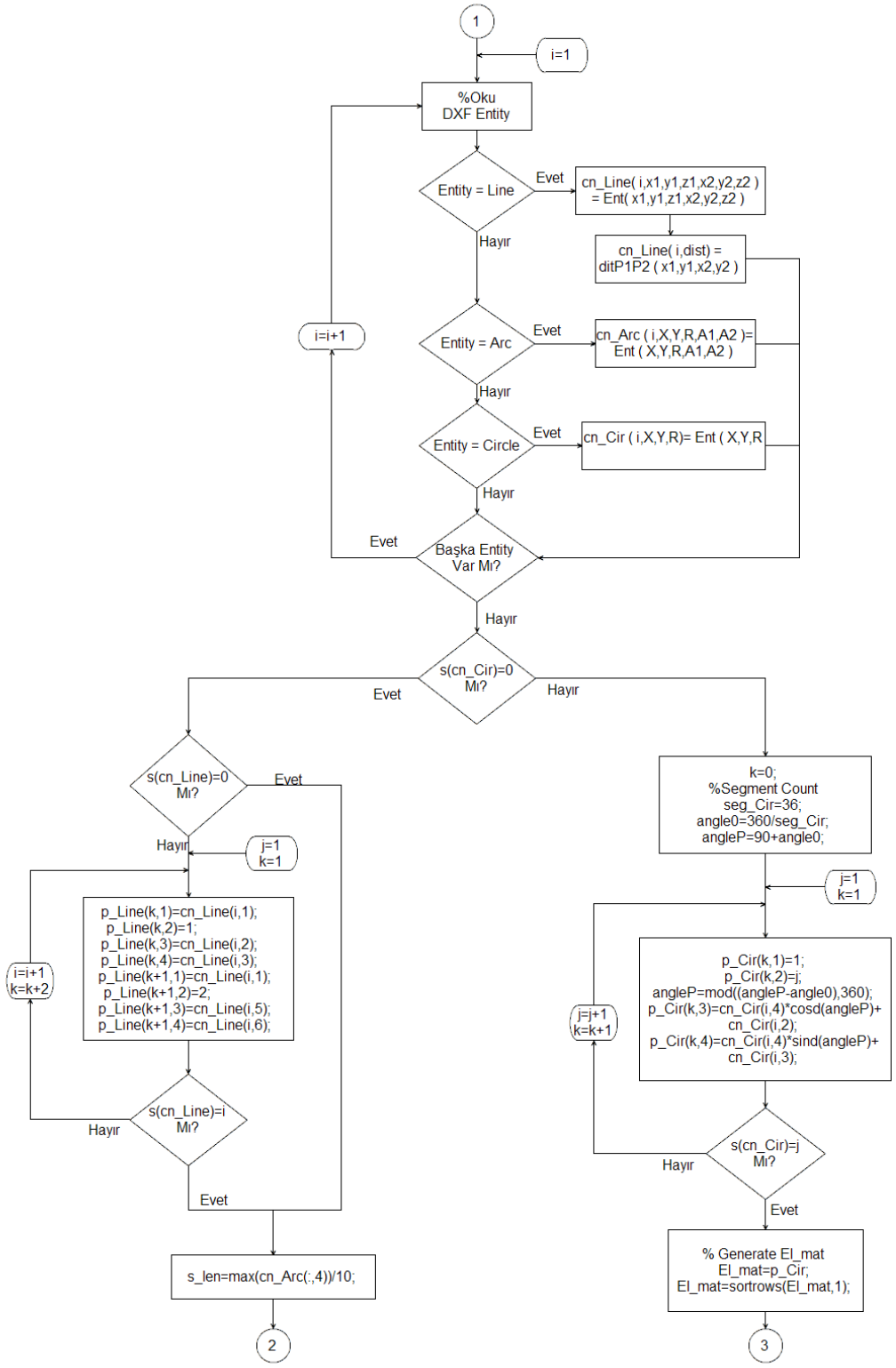
[26] A.J.M. Ferreira, MATLAB Codes For Finite Element Analysis. Solid Mechanics and Its Applications Springer 2009; 157.

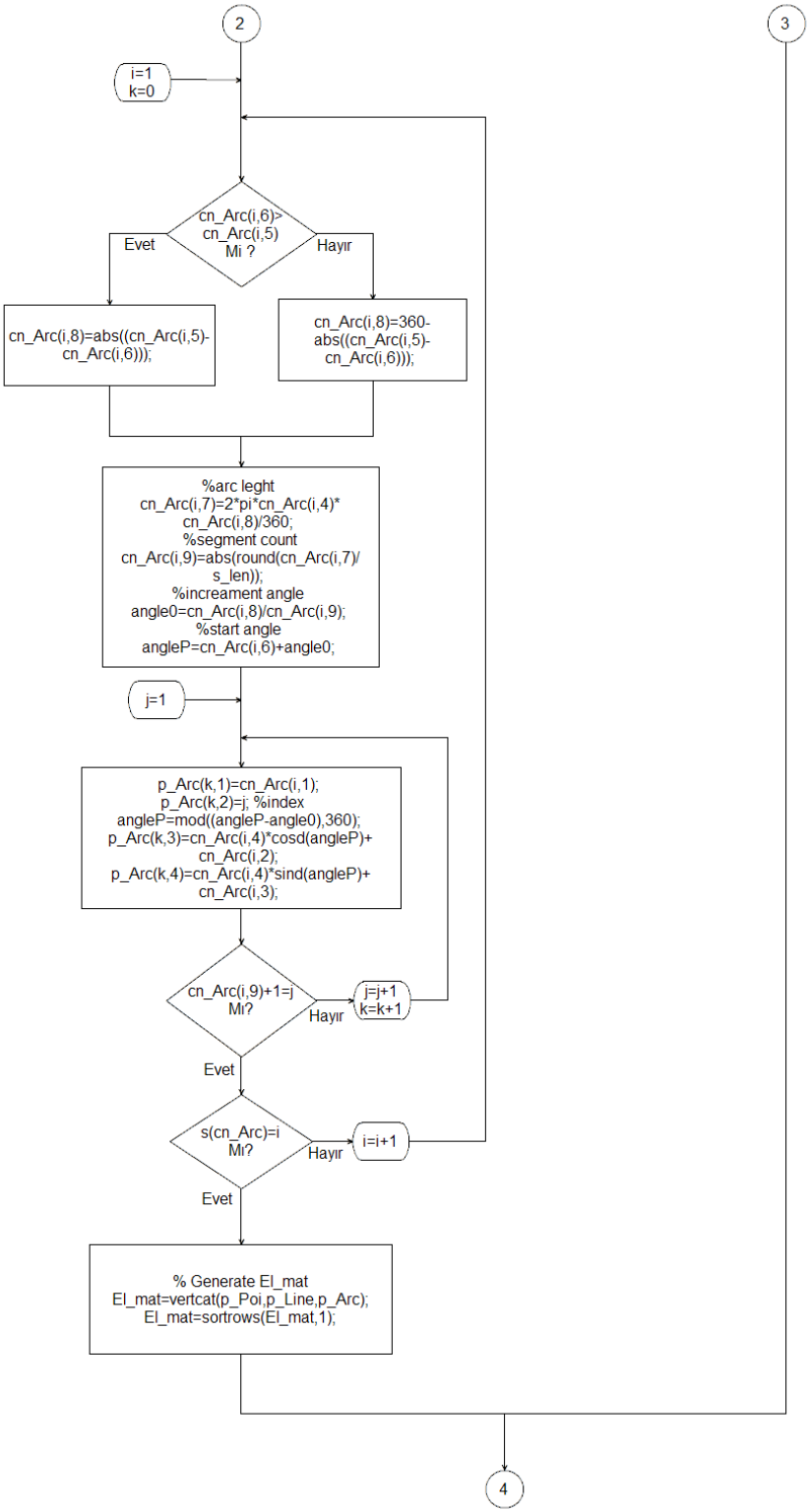
[27] Autodesk, Inc. Autocad 2008 DXF Reference. Autodesk; 2007.

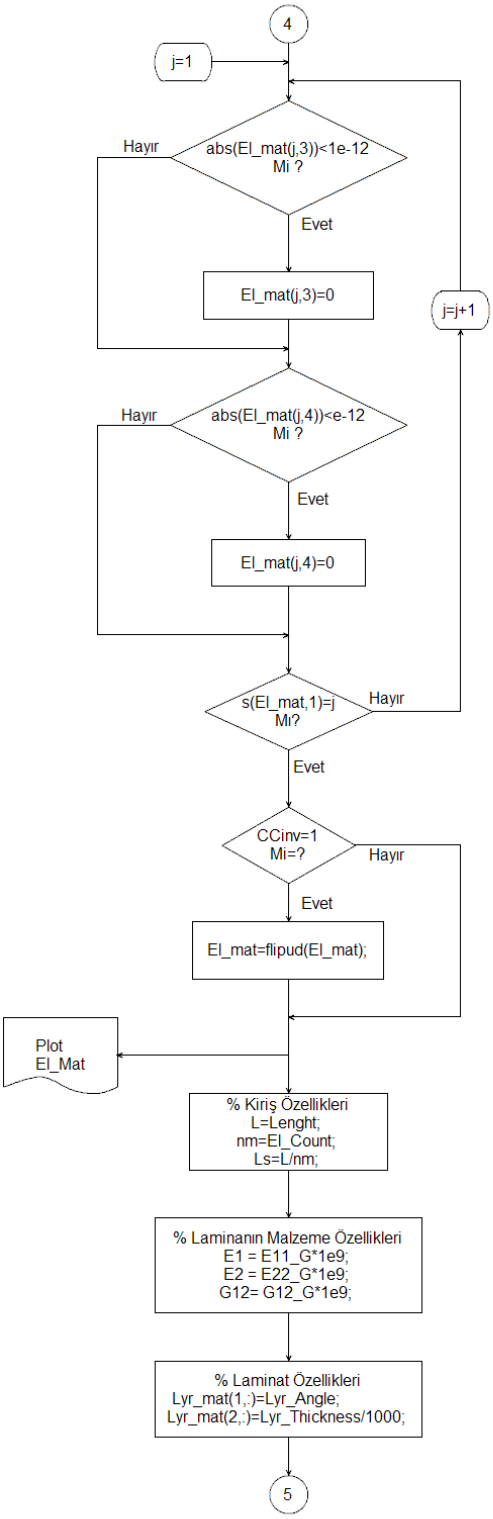
[28] Sebastian, Read DXF File Data. Http://Www.Mathworks.Com/Matlabcentral /Fileexchange/24572-Read-Dxf-File-Data; 2009.

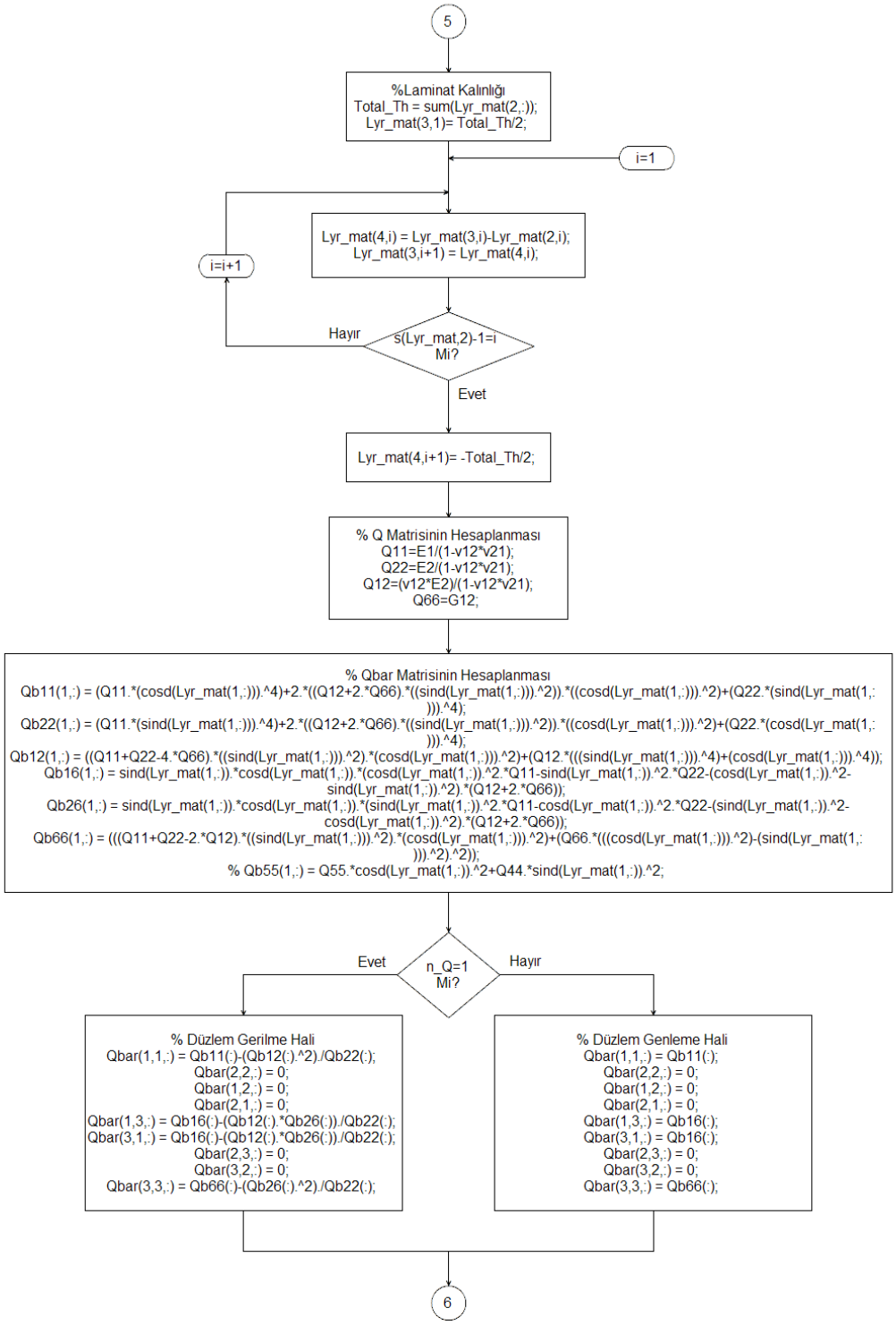
EK-A Yazılımın Akış Şeması

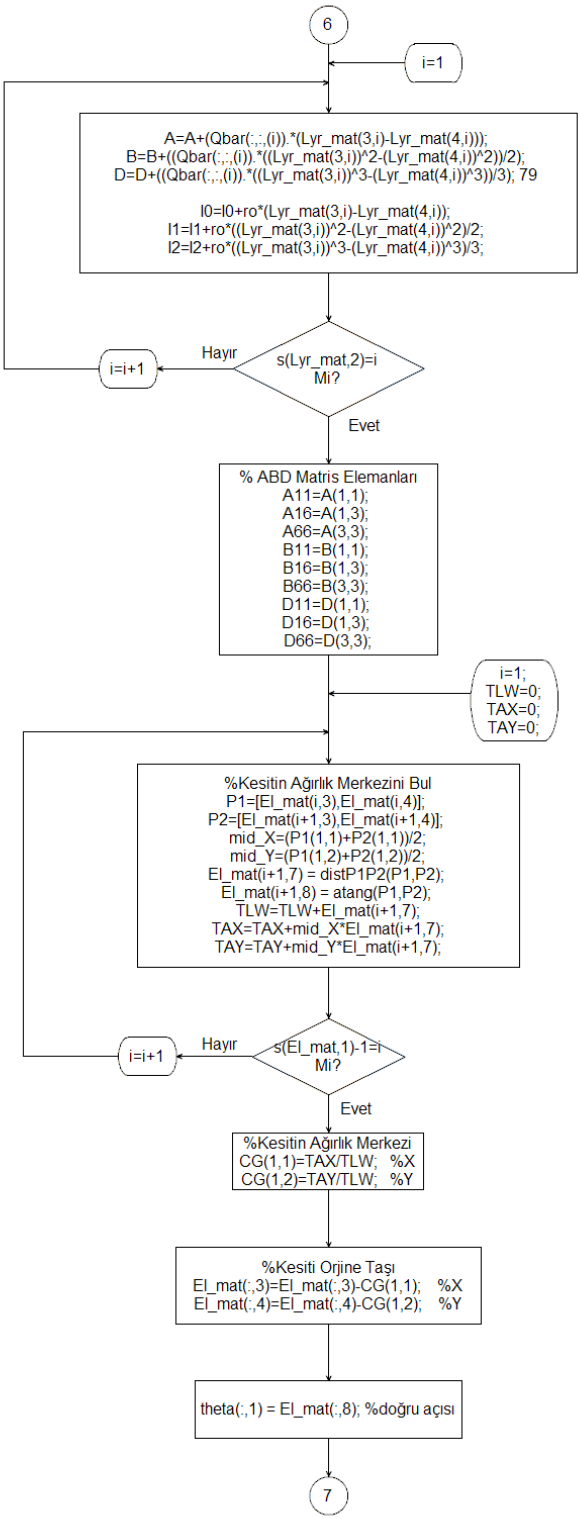


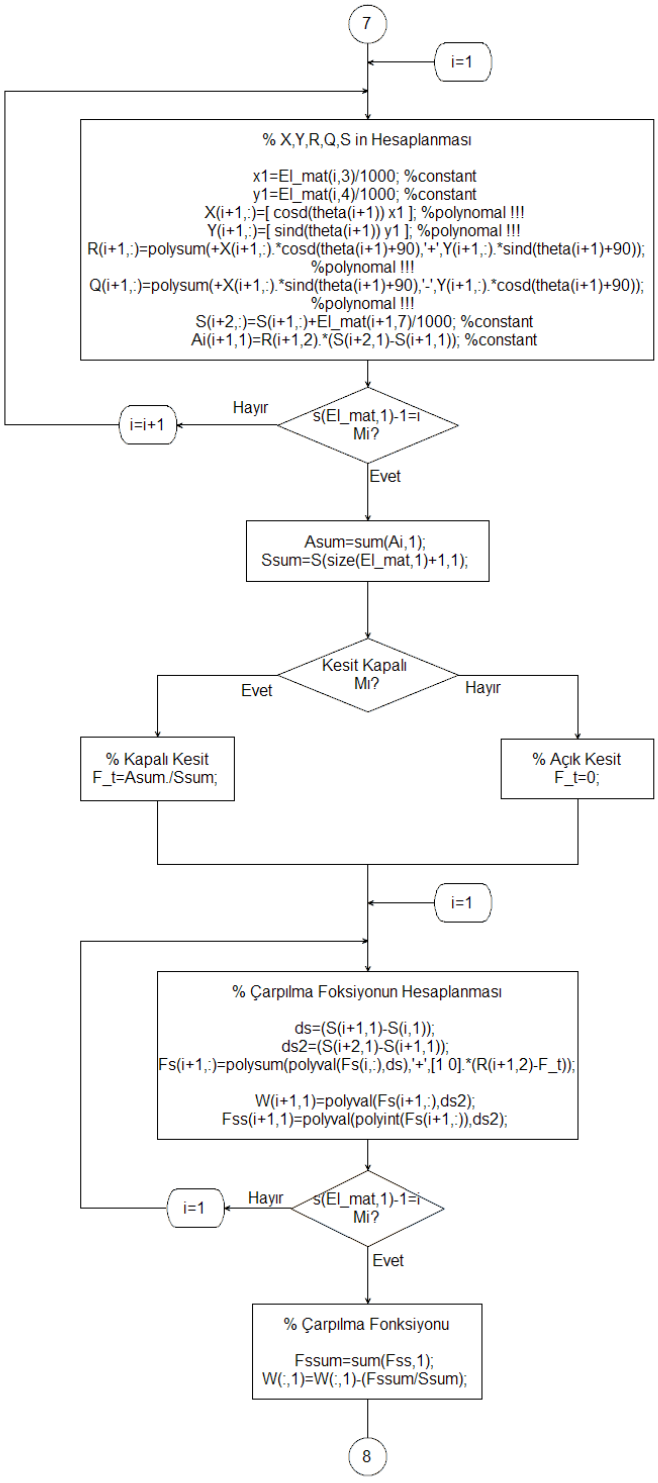

1
i=1
%Oku
DXF Entity
Entity = Line
Evet
cn_Line(i,x1,y1,z1,x2,y2,z2)
= Ent(x1,y1,z1,x2,y2,z2)
Hayır
cn_Line(i,dist) =
ditP1P2 (x1,y1,x2,y2)
i=i+1
Entity = Arc
Evet
cn_Arc (i,X,Y,R,A1,A2)=
Ent (X,Y,R,A1,A2)
Hayır
Entity = Circle
Evet
cn_Cir (i,X,Y,R)= Ent (X,Y,R
Hayır
Evet
Başka Entity
Var Mı?
Hayır
s(cn_Cir)=0
Mı?
Evet
Hayır
s(cn_Line)=0
Mı?
Evet
Hayır
j=1
k=1
p_Line(k,1)=cn_Line(i,1);
p_Line(k,2)=1;
p_Line(k,3)=cn_Line(i,2);
p_Line(k,4)=cn_Line(i,3);
p_Line(k+1,1)=cn_Line(i,1);
p_Line(k+1,2)=2;
p_Line(k+1,3)=cn_Line(i,5);
p_Line(k+1,4)=cn_Line(i,6);
i=i+1
k=k+2
s(cn_Line)=i
Mı?
Hayır
Evet
s_len=max(cn_Arc(:,4))/10;
2
k=0;
%Segment Count
seg_Cir=36;
angle0=360/seg_Cir;
angleP=90+angle0;
j=1
k=1
p_Cir(k,1)=1;
p_Cir(k,2)=j;
angleP=mod((angleP-angle0),360);
p_Cir(k,3)=cn_Cir(i,4)*cosd(angleP)+
cn_Cir(i,2);
p_Cir(k,4)=cn_Cir(i,4)*sind(angleP)+
cn_Cir(i,3);
j=j+1
k=k+1
s(cn_Cir)=j
Mı?
Hayır
Evet
% Generate El_mat
El_mat=p_Cir;
El_mat=sortrows(El_mat,1);
3

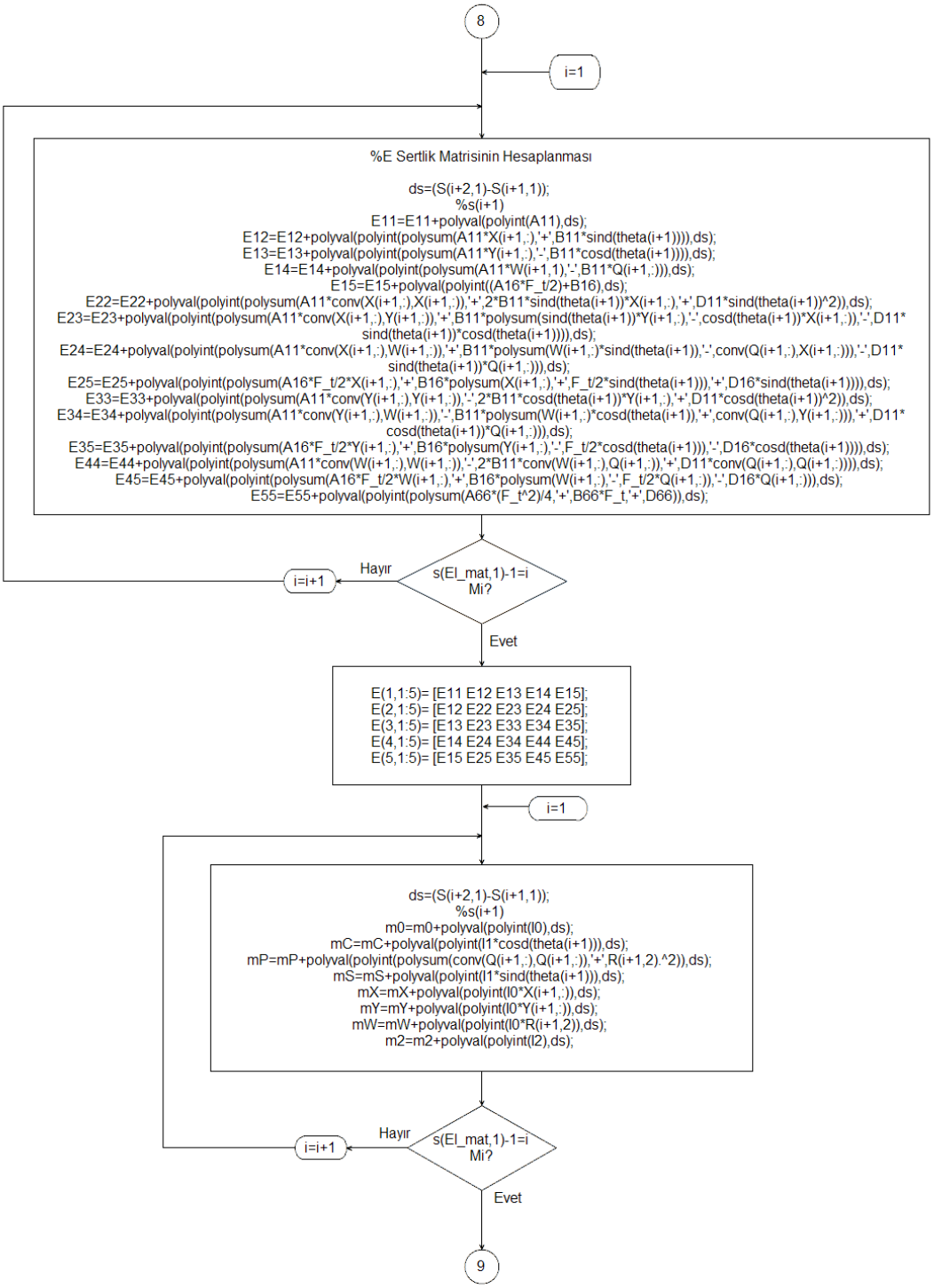

2
3
i=1
k=0
cn_Arc(i,6)>
cn_Arc(i,5)
Mi ?
Evet
Hayır
cn_Arc(i,8)=abs((cn_Arc(i,5)-
cn_Arc(i,6)));
cn_Arc(i,8)=360-
abs((cn_Arc(i,5)-
cn_Arc(i,6)));
%arc leght
cn_Arc(i,7)=2*pi*cn_Arc(i,4)*
cn_Arc(i,8)/360;
%segment count
cn_Arc(i,9)=abs(round(cn_Arc(i,7)/
s_len));
%increament angle
angle0=cn_Arc(i,8)/cn_Arc(i,9);
%start angle
angleP=cn_Arc(i,6)+angle0;
j=1
p_Arc(k,1)=cn_Arc(i,1);
p_Arc(k,2)=j; %index
angleP=mod((angleP-angle0),360);
p_Arc(k,3)=cn_Arc(i,4)*cosd(angleP)+
cn_Arc(i,2);
p_Arc(k,4)=cn_Arc(i,4)*sind(angleP)+
cn_Arc(i,3);
cn_Arc(i,9)+1=j
Mı?
Hayır
j=j+1
k=k+1
Evet
s(cn_Arc)=i
Mı?
Hayır
i=i+1
Evet
% Generate El_mat
El_mat=vertcat(p_Poi,p_Line,p_Arc);
El_mat=sortrows(El_mat,1);
4

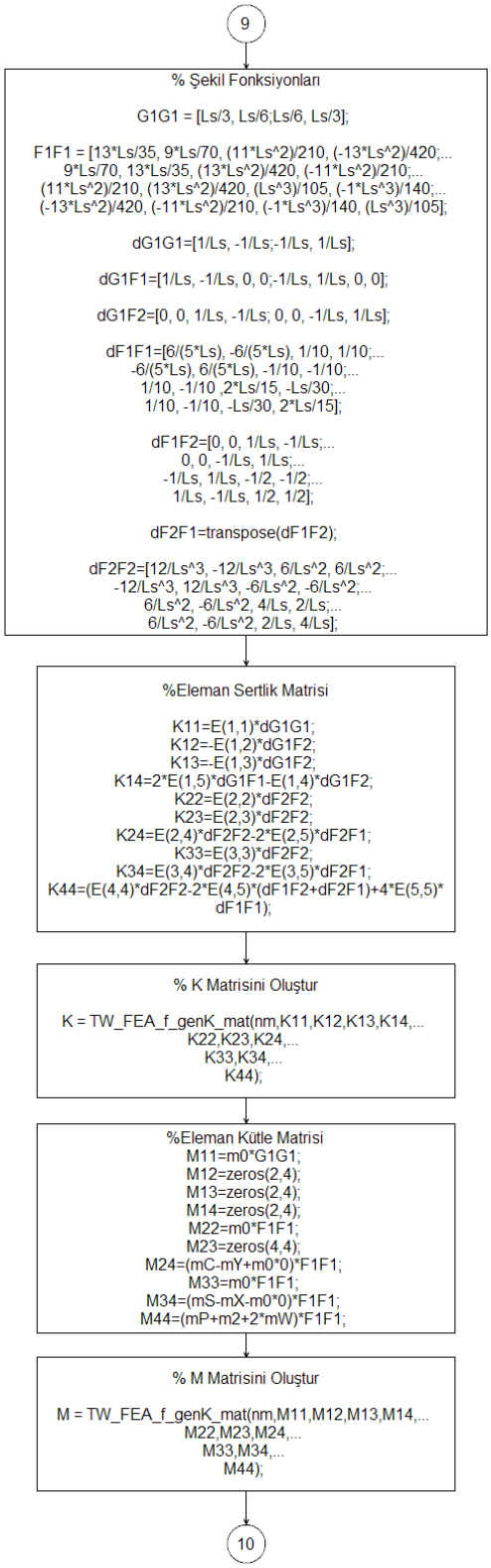

4
j=1
Hayır
abs(El_mat(j,3))<1e-12
Mi ?
Evet
El_mat(j,3)=0
j=j+1
Hayır
abs(El_mat(j,4))<e-12
Mi ?
Evet
El_mat(j,4)=0
s(El_mat,1)=j
Mı?
Hayır
Evet
CCinv=1
Mi=?
Hayır
Evet
El_mat=flipud(El_mat);
Plot
El_Mat
% Kiriş Özellikleri
L=Lenght;
nm=El_Count;
Ls=L/nm;
% Laminanın Malzeme Özellikleri
E1 = E11_G*1e9;
E2 = E22_G*1e9;
G12= G12_G*1e9;
% Laminat Özellikleri
Lyr_mat(1,:)=Lyr_Angle;
Lyr_mat(2,:)=Lyr_Thickness/1000;
5

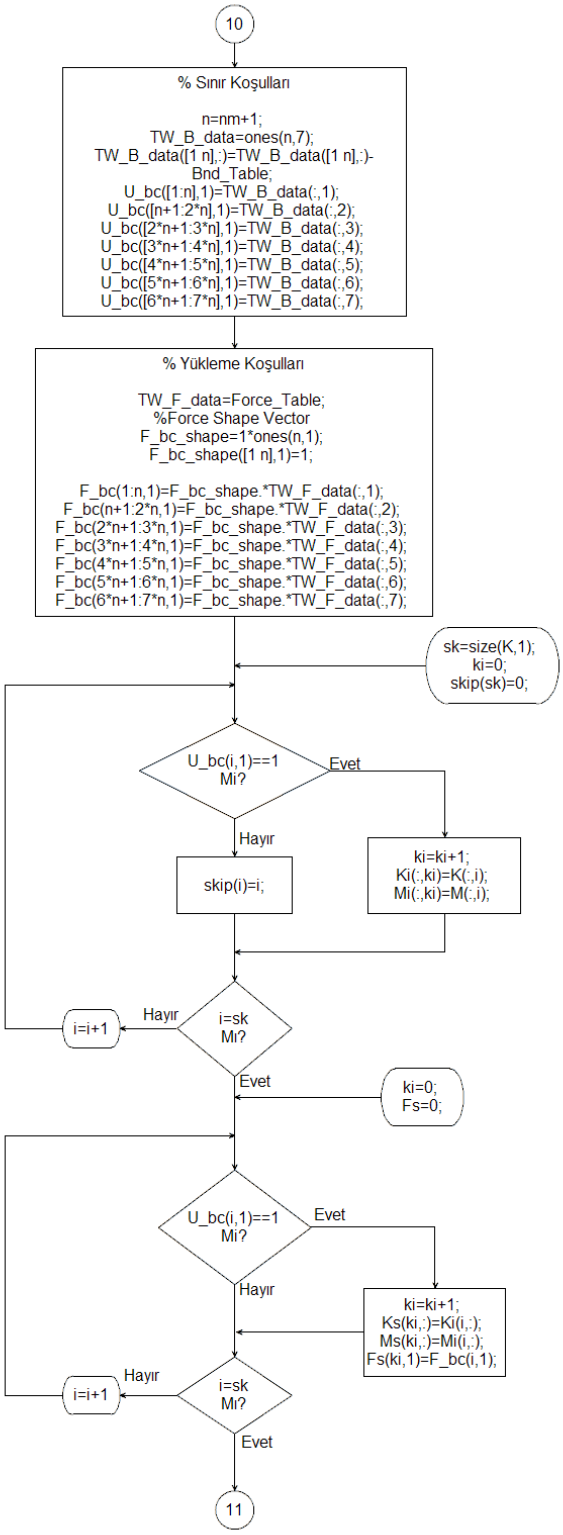

5
%Laminat Kalınlığı
Total_Th = sum(Lyr_mat(2,:));
Lyr_mat(3,1)= Total_Th/2;
i=1
Lyr_mat(4,i) = Lyr_mat(3,i)-Lyr_mat(2,i);
Lyr_mat(3,i+1) = Lyr_mat(4,i);
i=i+1
Hayır
s(Lyr_mat,2)-1=i
Mi?
Evet
Lyr_mat(4,i+1)= -Total_Th/2;
% Q Matrisinin Hesaplanması
Q11=E1/(1-v12*v21);
Q22=E2/(1-v12*v21);
Q12=(v12*E2)/(1-v12*v21);
Q66=G12;
% Qbar Matrisinin Hesaplanması
Qb11(1,:) = (Q11.*(cosd(Lyr_mat(1,:))).^4)+2.*((Q12+2.*Q66).*((sind(Lyr_mat(1,:))).^2)).*((cosd(Lyr_mat(1,:))).^2)+(Q22.*(sind(Lyr_mat(1,:))).^4);
Qb22(1,:) = (Q11.*(sind(Lyr_mat(1,:))).^4)+2.*((Q12+2.*Q66).*((sind(Lyr_mat(1,:))).^2)).*((cosd(Lyr_mat(1,:))).^2)+(Q22.*(cosd(Lyr_mat(1,:))).^4);
Qb12(1,:) = ((Q11+Q22-4.*Q66).*((sind(Lyr_mat(1,:))).^2).*(cosd(Lyr_mat(1,:))).^2)+(Q12.*(((sind(Lyr_mat(1,:))).^4)+(cosd(Lyr_mat(1,:))).^4));
Qb16(1,:) = sind(Lyr_mat(1,:)).*cosd(Lyr_mat(1,:)).*(cosd(Lyr_mat(1,:)).^2.*Q11-sind(Lyr_mat(1,:)).^2.*Q22-(cosd(Lyr_mat(1,:)).^2-sind(Lyr_mat(1,:)).^2).*(Q12+2.*Q66));
Qb26(1,:) = sind(Lyr_mat(1,:)).*cosd(Lyr_mat(1,:)).*(sind(Lyr_mat(1,:)).^2.*Q11-cosd(Lyr_mat(1,:)).^2.*Q22-(sind(Lyr_mat(1,:)).^2-cosd(Lyr_mat(1,:)).^2).*(Q12+2.*Q66));
Qb66(1,:) = (((Q11+Q22-2.*Q12).*((sind(Lyr_mat(1,:))).^2).*(cosd(Lyr_mat(1,:))).^2)+(Q66.*(((cosd(Lyr_mat(1,:))).^2)-(sind(Lyr_mat(1,:))).^2).^2));
% Qb55(1,:) = Q55.*cosd(Lyr_mat(1,:)).^2+Q44.*sind(Lyr_mat(1,:)).^2;
Evet
n_Q=1
Mi?
Hayır
% Düzlem Gerilme Hali
Qbar(1,1,:) = Qb11(:)-(Qb12(:).^2)./Qb22(:);
Qbar(2,2,:) = 0;
Qbar(1,2,:) = 0;
Qbar(2,1,:) = 0;
Qbar(1,3,:) = Qb16(:)-(Qb12(:).*Qb26(:))./Qb22(:);
Qbar(3,1,:) = Qb16(:)-(Qb12(:).*Qb26(:))./Qb22(:);
Qbar(2,3,:) = 0;
Qbar(3,2,:) = 0;
Qbar(3,3,:) = Qb66(:)-(Qb26(:).^2)./Qb22(:);
% Düzlem Genleme Hali
Qbar(1,1,:) = Qb11(:);
Qbar(2,2,:) = 0;
Qbar(1,2,:) = 0;
Qbar(2,1,:) = 0;
Qbar(1,3,:) = Qb16(:);
Qbar(3,1,:) = Qb16(:);
Qbar(2,3,:) = 0;
Qbar(3,2,:) = 0;
Qbar(3,3,:) = Qb66(:);
6

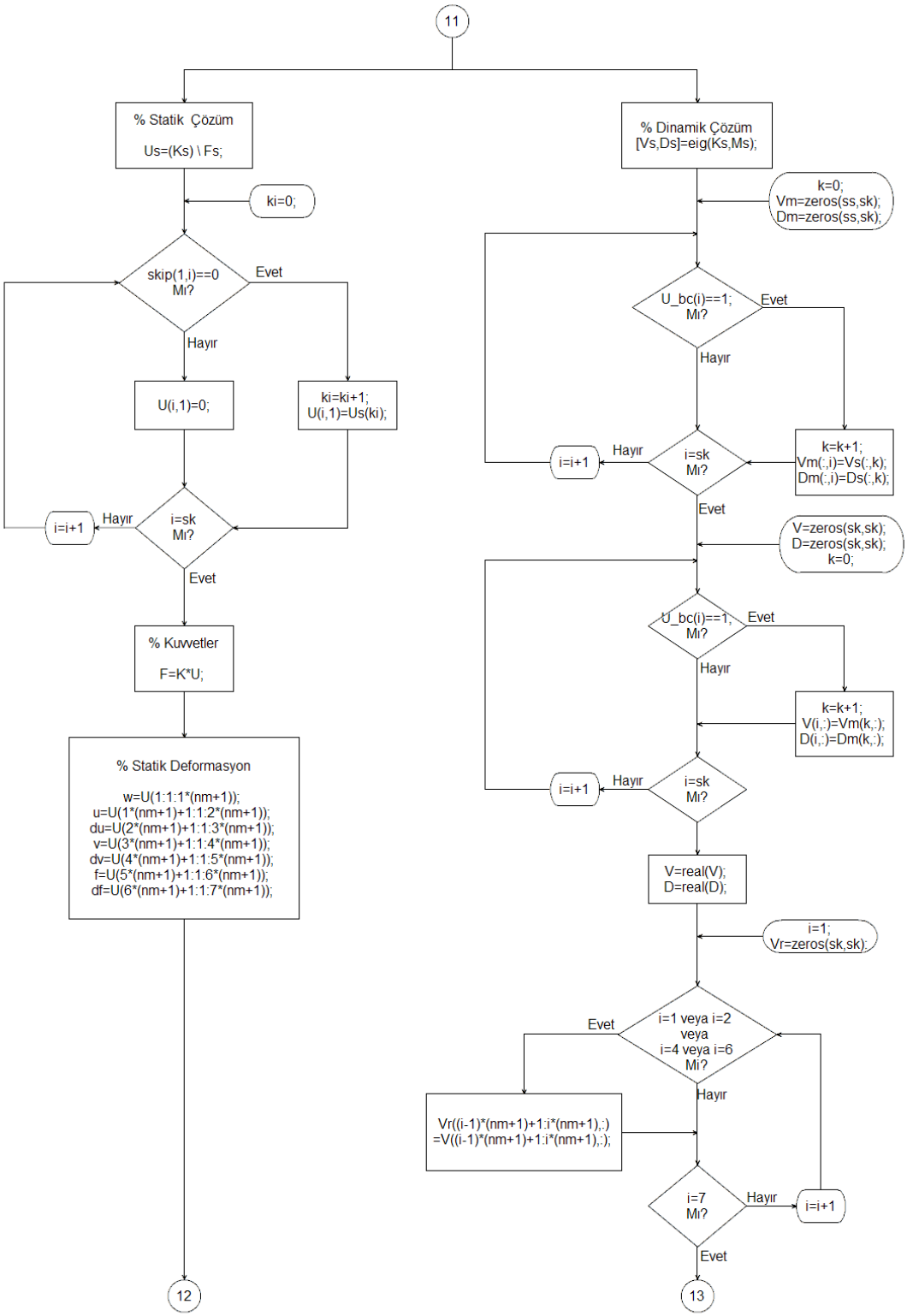

6
i=1
A=A+(Qbar(:,:,(i)).*(Lyr_mat(3,i)-Lyr_mat(4,i)));
B=B+((Qbar(:,:,(i)).*((Lyr_mat(3,i))^2-(Lyr_mat(4,i))^2))/2);
D=D+((Qbar(:,:,(i)).*((Lyr_mat(3,i))^3-(Lyr_mat(4,i))^3))/3); 79
I0=I0+ro*(Lyr_mat(3,i)-Lyr_mat(4,i));
I1=I1+ro*((Lyr_mat(3,i))^2-(Lyr_mat(4,i))^2)/2;
I2=I2+ro*((Lyr_mat(3,i))^3-(Lyr_mat(4,i))^3)/3;
s(Lyr_mat,2)=i Mi?
Hayır
i=i+1
Evet
% ABD Matris Elemanları
A11=A(1,1);
A16=A(1,3);
A66=A(3,3);
B11=B(1,1);
B16=B(1,3);
B66=B(3,3);
D11=D(1,1);
D16=D(1,3);
D66=D(3,3);
i=1;
TLW=0;
TAX=0;
TAY=0;
%Kesitin Ağırlık Merkezini Bul
P1=[El_mat(i,3),El_mat(i,4)];
P2=[El_mat(i+1,3),El_mat(i+1,4)];
mid_X=(P1(1,1)+P2(1,1))/2;
mid_Y=(P1(1,2)+P2(1,2))/2;
El_mat(i+1,7) = distP1P2(P1,P2);
El_mat(i+1,8) = atang(P1,P2);
TLW=TLW+El_mat(i+1,7);
TAX=TAX+mid_X*El_mat(i+1,7);
TAY=TAY+mid_Y*El_mat(i+1,7);
s(El_mat,1)-1=i Mi?
Hayır
i=i+1
Evet
%Kesitin Ağırlık Merkezi
CG(1,1)=TAX/TLW; %X
CG(1,2)=TAY/TLW; %Y
%Kesiti Orjine Taşı
El_mat(:,3)=El_mat(:,3)-CG(1,1); %X
El_mat(:,4)=El_mat(:,4)-CG(1,2); %Y
theta(:,1) = El_mat(:,8); %doğru açısı
7

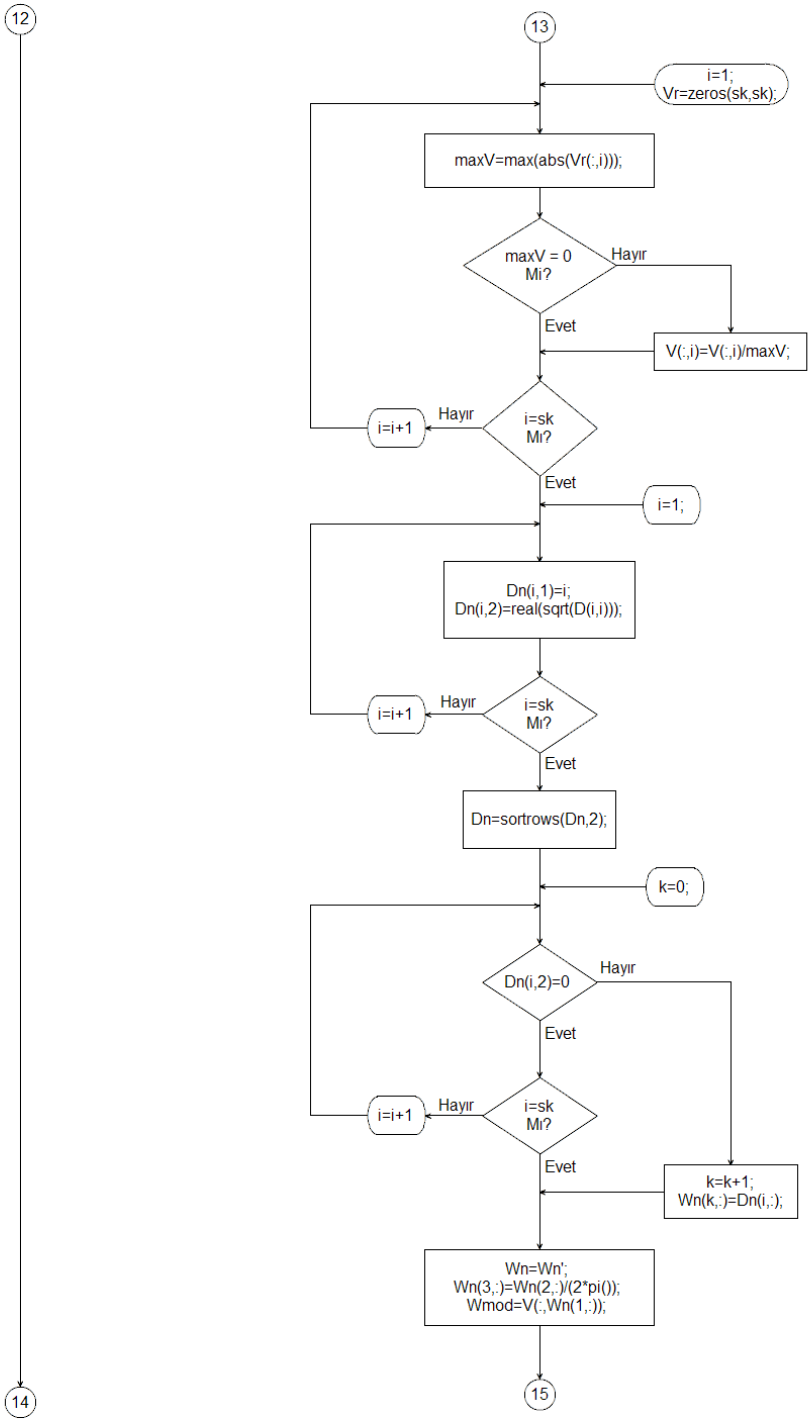

7
i=1
% X,Y,R,Q,S in Hesaplanması
x1=El_mat(i,3)/1000; %constant
y1=El_mat(i,4)/1000; %constant
X(i+1,:)=[cosd(theta(i+1)) x1]; %polynomal !!!
Y(i+1,:)=[sind(theta(i+1)) y1]; %polynomal !!!
R(i+1,:)=polysum(+X(i+1,:).*cosd(theta(i+1)+90),'+',Y(i+1,:).*sind(theta(i+1)+90));
%polynomal !!!
Q(i+1,:)=polysum(+X(i+1,:).*sind(theta(i+1)+90),'-',Y(i+1,:).*cosd(theta(i+1)+90));
%polynomal !!!
S(i+2,:)=S(i+1,:)+El_mat(i+1,7)/1000; %constant
Ai(i+1,1)=R(i+1,2).*(S(i+2,1)-S(i+1,1)); %constant
s(El_mat,1)-1=i Mı?
Hayır
i=i+1
Evet
Asum=sum(Ai,1);
Ssum=S(size(El_mat,1)+1,1);
Kesit Kapalı Mı?
Evet
Hayır
% Kapalı Kesit
F_t=Asum./Ssum;
% Açık Kesit
F_t=0;
i=1
% Çarpılma Foksiyonun Hesaplanması
ds=(S(i+1,1)-S(i,1));
ds2=(S(i+2,1)-S(i+1,1));
Fs(i+1,:)=polysum(polyval(Fs(i,:),ds),'+',[1 0].*(R(i+1,2)-F_t));
W(i+1,1)=polyval(Fs(i+1,:),ds2);
Fss(i+1,1)=polyval(polyint(Fs(i+1,:)),ds2);
s(El_mat,1)-1=i Mı?
Hayır
i=1
Evet
% Çarpılma Fonksiyonu
Fssum=sum(Fss,1);
W(:,1)=W(:,1)-(Fssum/Ssum);
8

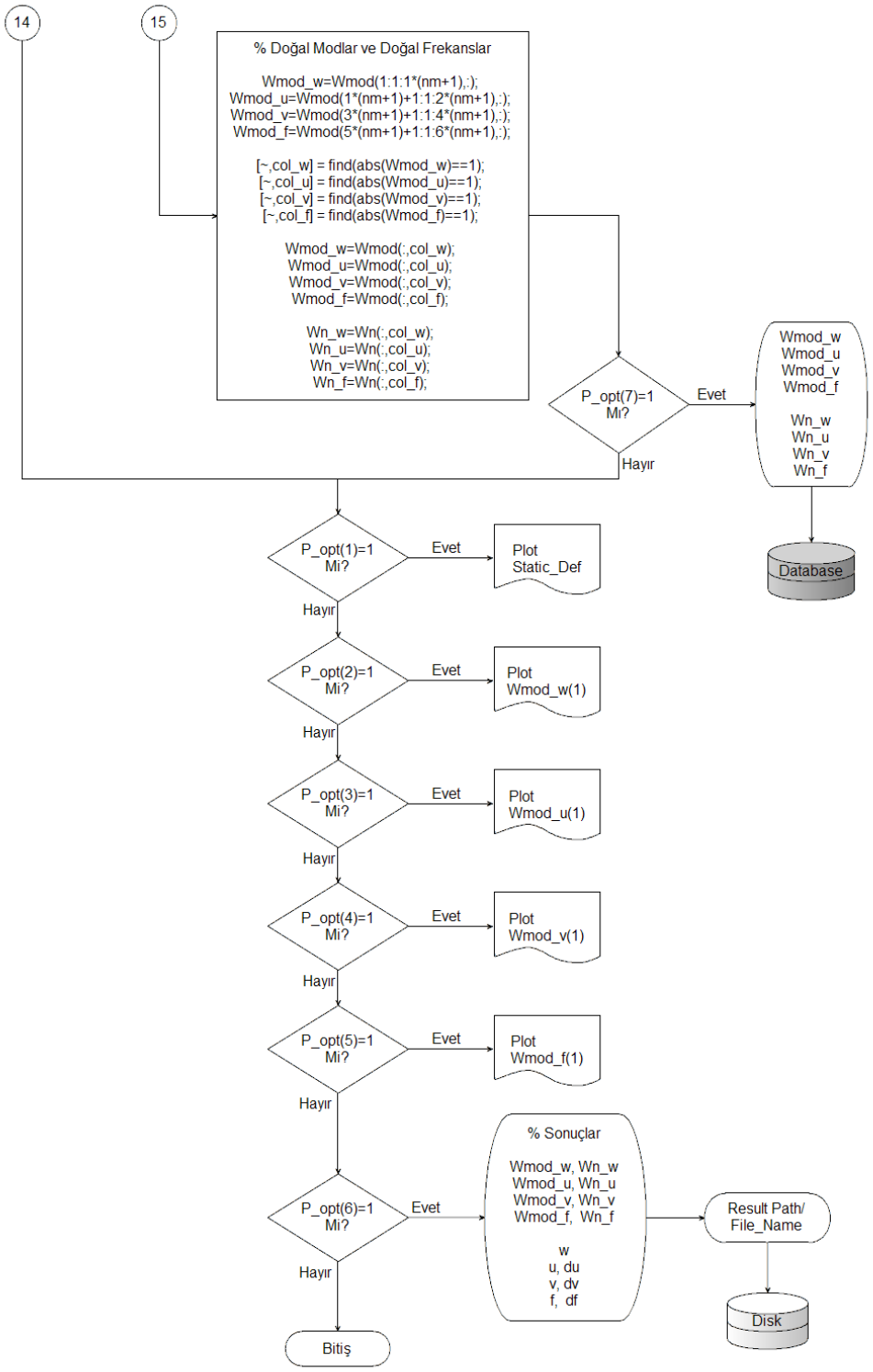

8
i=1
%E Sertlik Matrisinin Hesaplanması
ds=(S(i+2,1)-S(i+1,1));
%s(i+1)
E11=E11+polyval(polyint(A11),ds);
E12=E12+polyval(polyint(polysum(A11*X(i+1,:),'+',B11*sind(theta(i+1)))),ds);
E13=E13+polyval(polyint(polysum(A11*Y(i+1,:),'-',B11*cosd(theta(i+1)))),ds);
E14=E14+polyval(polyint(polysum(A11*W(i+1,1),'-',B11*Q(i+1,:))),ds);
E15=E15+polyval(polyint((A16*F_t/2)+B16),ds);
E22=E22+polyval(polyint(polysum(A11*conv(X(i+1,:),X(i+1,:)),'+',2*B11*sind(theta(i+1))*X(i+1,:),'+',D11*sind(theta(i+1))^2)),ds);
E23=E23+polyval(polyint(polysum(A11*conv(X(i+1,:),Y(i+1,:)),'+',B11*polysum(sind(theta(i+1))*Y(i+1,:),'-',cosd(theta(i+1))*X(i+1,:)),'-',D11*sind(theta(i+1))*cosd(theta(i+1)))),ds);
E24=E24+polyval(polyint(polysum(A11*conv(X(i+1,:),W(i+1,:)),'+',B11*polysum(W(i+1,:)*sind(theta(i+1)),'-',conv(Q(i+1,:),X(i+1,:))),'-',D11*sind(theta(i+1))*Q(i+1,:))),ds);
E25=E25+polyval(polyint(polysum(A16*F_t/2*X(i+1,:),'+',B16*polysum(X(i+1,:),'+',F_t/2*sind(theta(i+1))),'+',D16*sind(theta(i+1)))),ds);
E33=E33+polyval(polyint(polysum(A11*conv(Y(i+1,:),Y(i+1,:)),'-',2*B11*cosd(theta(i+1))*Y(i+1,:),'+',D11*cosd(theta(i+1))^2)),ds);
E34=E34+polyval(polyint(polysum(A11*conv(Y(i+1,:),W(i+1,:)),'-',B11*polysum(W(i+1,:)*cosd(theta(i+1)),'+',conv(Q(i+1,:),Y(i+1,:))),'+',D11*cosd(theta(i+1))*Q(i+1,:))),ds);
E35=E35+polyval(polyint(polysum(A16*F_t/2*Y(i+1,:),'+',B16*polysum(Y(i+1,:),'-',F_t/2*cosd(theta(i+1))),'-',D16*cosd(theta(i+1)))),ds);
E44=E44+polyval(polyint(polysum(A11*conv(W(i+1,:),W(i+1,:)),'-',2*B11*conv(W(i+1,:),Q(i+1,:)),'+',D11*conv(Q(i+1,:),Q(i+1,:)))),ds);
E45=E45+polyval(polyint(polysum(A16*F_t/2*W(i+1,:),'+',B16*polysum(W(i+1,:),'-',F_t/2*Q(i+1,:)),'-',D16*Q(i+1,:))),ds);
E55=E55+polyval(polyint(polysum(A66*(F_t^2)/4,'+',B66*F_t,'+',D66)),ds);
s(El_mat,1)-1=i
Mi?
Hayır
i=i+1
Evet
E(1,1:5)= [E11 E12 E13 E14 E15];
E(2,1:5)= [E12 E22 E23 E24 E25];
E(3,1:5)= [E13 E23 E33 E34 E35];
E(4,1:5)= [E14 E24 E34 E44 E45];
E(5,1:5)= [E15 E25 E35 E45 E55];
i=1
ds=(S(i+2,1)-S(i+1,1));
%s(i+1)
m0=m0+polyval(polyint(l0),ds);
mC=mC+polyval(polyint(l1*cosd(theta(i+1))),ds);
mP=mP+polyval(polyint(polysum(conv(Q(i+1,:),Q(i+1,:)),'+',R(i+1,2).^2)),ds);
mS=mS+polyval(polyint(l1*sind(theta(i+1))),ds);
mX=mX+polyval(polyint(l0*X(i+1,:)),ds);
mY=mY+polyval(polyint(l0*Y(i+1,:)),ds);
mW=mW+polyval(polyint(l0*R(i+1,2)),ds);
m2=m2+polyval(polyint(l2),ds);
s(El_mat,1)-1=i
Mi?
Hayır
i=i+1
Evet
9


9
% Şekil Fonksiyonları
G1G1 = [Ls/3, Ls/6;Ls/6, Ls/3];
F1F1 = [13*Ls/35, 9*Ls/70, (11*Ls^2)/210, (-13*Ls^2)/420;...
9*Ls/70, 13*Ls/35, (13*Ls^2)/420, (-11*Ls^2)/210;...
(11*Ls^2)/210, (13*Ls^2)/420, (Ls^3)/105, (-1*Ls^3)/140;...
(-13*Ls^2)/420, (-11*Ls^2)/210, (-1*Ls^3)/140, (Ls^3)/105];
dG1G1=[1/Ls, -1/Ls;-1/Ls, 1/Ls];
dG1F1=[1/Ls, -1/Ls, 0, 0;-1/Ls, 1/Ls, 0, 0];
dG1F2=[0, 0, 1/Ls, -1/Ls; 0, 0, -1/Ls, 1/Ls];
dF1F1=[6/(5*Ls), -6/(5*Ls), 1/10, 1/10;...
-6/(5*Ls), 6/(5*Ls), -1/10, -1/10;...
1/10, -1/10 ,2*Ls/15, -Ls/30;...
1/10, -1/10, -Ls/30, 2*Ls/15];
dF1F2=[0, 0, 1/Ls, -1/Ls;...
0, 0, -1/Ls, 1/Ls;...
-1/Ls, 1/Ls, -1/2, -1/2;...
1/Ls, -1/Ls, 1/2, 1/2];
dF2F1=transpose(dF1F2);
dF2F2=[12/Ls^3, -12/Ls^3, 6/Ls^2, 6/Ls^2;...
-12/Ls^3, 12/Ls^3, -6/Ls^2, -6/Ls^2;...
6/Ls^2, -6/Ls^2, 4/Ls, 2/Ls;...
6/Ls^2, -6/Ls^2, 2/Ls, 4/Ls];
%Eleman Sertlik Matrisi
K11=E(1,1)*dG1G1;
K12=-E(1,2)*dG1F2;
K13=-E(1,3)*dG1F2;
K14=2*E(1,5)*dG1F1-E(1,4)*dG1F2;
K22=E(2,2)*dF2F2;
K23=E(2,3)*dF2F2;
K24=E(2,4)*dF2F2-2*E(2,5)*dF2F1;
K33=E(3,3)*dF2F2;
K34=E(3,4)*dF2F2-2*E(3,5)*dF2F1;
K44=(E(4,4)*dF2F2-2*E(4,5)*(dF1F2+dF2F1)+4*E(5,5)*
dF1F1);
% K Matrisini Oluştur
K = TW_FEA_f_genK_mat(nm,K11,K12,K13,K14,...
K22,K23,K24,...
K33,K34,...
K44);
%Eleman Kütle Matrisi
M11=m0*G1G1;
M12=zeros(2,4);
M13=zeros(2,4);
M14=zeros(2,4);
M22=m0*F1F1;
M23=zeros(4,4);
M24=(mC-mY+m0*0)*F1F1;
M33=m0*F1F1;
M34=(mS-mX-m0*0)*F1F1;
M44=(mP+m2+2*mW)*F1F1;
% M Matrisini Oluştur
M = TW_FEA_f_genK_mat(nm,M11,M12,M13,M14,...
M22,M23,M24,...
M33,M34,...
M44);
10


10
% Sınır Koşulları
n=nm+1;
TW_B_data=ones(n,7);
TW_B_data([1 n],:)=TW_B_data([1 n],:)-
Bnd_Table;
U_bc([1:n],1)=TW_B_data(:,1);
U_bc([n+1:2*n],1)=TW_B_data(:,2);
U_bc([2*n+1:3*n],1)=TW_B_data(:,3);
U_bc([3*n+1:4*n],1)=TW_B_data(:,4);
U_bc([4*n+1:5*n],1)=TW_B_data(:,5);
U_bc([5*n+1:6*n],1)=TW_B_data(:,6);
U_bc([6*n+1:7*n],1)=TW_B_data(:,7);
% Yükleme Koşulları
TW_F_data=Force_Table;
%Force Shape Vector
F_bc_shape=1*ones(n,1);
F_bc_shape([1 n],1)=1;
F_bc(1:n,1)=F_bc_shape.*TW_F_data(:,1);
F_bc(n+1:2*n,1)=F_bc_shape.*TW_F_data(:,2);
F_bc(2*n+1:3*n,1)=F_bc_shape.*TW_F_data(:,3);
F_bc(3*n+1:4*n,1)=F_bc_shape.*TW_F_data(:,4);
F_bc(4*n+1:5*n,1)=F_bc_shape.*TW_F_data(:,5);
F_bc(5*n+1:6*n,1)=F_bc_shape.*TW_F_data(:,6);
F_bc(6*n+1:7*n,1)=F_bc_shape.*TW_F_data(:,7);
sk=size(K,1);
ki=0;
skip(sk)=0;
U_bc(i,1)==1
Mi?
Evet
Hayır
skip(i)=i;
ki=ki+1;
Ki(:,ki)=K(:,i);
Mi(:,ki)=M(:,i);
i=sk
Mı?
Hayır
i=i+1
Evet
ki=0;
Fs=0;
U_bc(i,1)==1
Mi?
Evet
Hayır
ki=ki+1;
Ks(ki,:)=Ki(i,:);
Ms(ki,:)=Mi(i,:);
Fs(ki,1)=F_bc(i,1);
i=sk
Mı?
Hayır
i=i+1
Evet
11


11
% Statik Çözüm
Us=(Ks) \ Fs;
ki=0;
skip(1,i)==0
Mı?
Evet
Hayır
U(i,1)=0;
ki=ki+1;
U(i,1)=Us(ki);
i=sk
Mı?
Hayır
i=i+1
Evet
% Kuvvetler
F=K*U;
% Statik Deformasyon
w=U(1:1:1*(nm+1));
u=U(1*(nm+1)+1:1:2*(nm+1));
du=U(2*(nm+1)+1:1:3*(nm+1));
v=U(3*(nm+1)+1:1:4*(nm+1));
dv=U(4*(nm+1)+1:1:5*(nm+1));
f=U(5*(nm+1)+1:1:6*(nm+1));
df=U(6*(nm+1)+1:1:7*(nm+1));
12
% Dinamik Çözüm
[Vs,Ds]=eig(Ks,Ms);
k=0;
Vm=zeros(ss,sk);
Dm=zeros(ss,sk);
U_bc(i)==1;
Mı?
Evet
Hayır
k=k+1;
Vm(:,i)=Vs(:,k);
Dm(:,i)=Ds(:,k);
i=sk
Mı?
Hayır
i=i+1
Evet
V=zeros(sk,sk);
D=zeros(sk,sk);
k=0;
U_bc(i)==1;
Mı?
Evet
Hayır
k=k+1;
V(i,:)=Vm(k,:);
D(i,:)=Dm(k,:);
i=sk
Mı?
Hayır
i=i+1
Evet
V=real(V);
D=real(D);
i=1;
Vr=zeros(sk,sk);
i=1 veya i=2
veya
i=4 veya i=6
Mi?
Evet
Hayır
Vr((i-1)*(nm+1)+1:i*(nm+1),:)
=V((i-1)*(nm+1)+1:i*(nm+1),:);
i=7
Mı?
Hayır
i=i+1
Evet
13


12
13
i=1;
Vr=zeros(sk,sk);
maxV=max(abs(Vr(:,i)));
maxV = 0
Mı?
Hayır
Evet
V(:,i)=V(:,i)/maxV;
i=sk
Mı?
Hayır
i=i+1
Evet
i=1;
Dn(i,1)=i;
Dn(i,2)=real(sqrt(D(i,i)));
i=sk
Mı?
Hayır
i=i+1
Evet
Dn=sortrows(Dn,2);
k=0;
Dn(i,2)=0
Hayır
Evet
i=sk
Mı?
Hayır
i=i+1
Evet
k=k+1;
Wn(k,:)=Dn(i,:);
Wn=Wn';
Wn(3,:)=Wn(2,:)/(2*pi());
Wmod=V(:,Wn(1,:));
14
15


14
15
% Doğal Modlar ve Doğal Frekanslar
Wmod_w=Wmod(1:1:1*(nm+1),:);
Wmod_u=Wmod(1*(nm+1)+1:1:2*(nm+1),:);
Wmod_v=Wmod(3*(nm+1)+1:1:4*(nm+1),:);
Wmod_f=Wmod(5*(nm+1)+1:1:6*(nm+1),:);
[~,col_w] = find(abs(Wmod_w)==1);
[~,col_u] = find(abs(Wmod_u)==1);
[~,col_v] = find(abs(Wmod_v)==1);
[~,col_f] = find(abs(Wmod_f)==1);
Wmod_w=Wmod(:,col_w);
Wmod_u=Wmod(:,col_u);
Wmod_v=Wmod(:,col_v);
Wmod_f=Wmod(:,col_f);
Wn_w=Wn(:,col_w);
Wn_u=Wn(:,col_u);
Wn_v=Wn(:,col_v);
Wn_f=Wn(:,col_f);
P_opt(7)=1 Mi?
Evet
Hayır
Wmod_w
Wmod_u
Wmod_v
Wmod_f
Wn_w
Wn_u
Wn_v
Wn_f
Database
P_opt(1)=1 Mi?
Plot Static_Def
P_opt(2)=1 Mi?
Plot Wmod_w(1)
P_opt(3)=1 Mi?
Plot Wmod_u(1)
P_opt(4)=1 Mi?
Plot Wmod_v(1)
P_opt(5)=1 Mi?
Plot Wmod_f(1)
P_opt(6)=1 Mi?
% Sonuçlar
Wmod_w, Wn_w
Wmod_u, Wn_u
Wmod_v, Wn_v
Wmod_f, Wn_f
w
u, du
v, dv
f, df
Result Path/ File_Name
Disk
Bitiş

EK-B Yazılımın Matlab Kodları

TW_FEA_Main (Ana Hesaplayıcı ve Çözücü)

```
function TW_FEA_Main
% M.Gökhan Günay 2013
% Thin Walled Composite Beam FEA Solver
% Main Code

clear all
evalin('base','clear')
clc

tic

%% Input Data

load('GUI_DATA.mat')

input_flag=1;

switch input_flag

   case 1

% Lenght Data
L=L_G;
% Element Number
nm=ec;
Ls=L/nm;

% Material Properties Of Lamina
E1 = E11_G*1e9;
E2 = E22_G*1e9;
G12= G12_G*1e9;
v12= v12_G;
v21= v21_G;

ro=ro_G;
```

```
% Lyr_mat Data
Lyr_mat(1,:)=angle_G;
Lyr_mat(2,:)=thickness_G/1000;

%El_mat
El_mat=El_mat_G;


   case 2

% Lenght Data
L=1;
% Element Number
nm=20;
Ls=L/nm;

% Material Properties Of Lamina
E1 = 142e9;
E2 = 9.8e9;
G12= 6e9;
% G23= 6e9;
% G13= 4.8e9;
v12= 0.42;
v21= 0.028;

ro=1445;

% Lyr_mat Data
Lyr_mat=[45 -45 ; 0.001 0.001];

%El_mat Def
El_mat=[1 1 1183.62686996490 519.446055257164;...
      1 2 1183.62686996490 532.284055257164;...
      2 2 1207.06486996490 532.284055257164;...
      3 2 1207.06486996490 519.446055257164;...
      4 2 1183.62686996490 519.446055257164;];

end
```

```
%% Calculations

% Total Thickness of Laminate
Total_Th = sum(Lyr_mat(2,:));

%  Definition of Zi for Each Lamina

Lyr_mat(3,1)= Total_Th/2;
for i=1:size(Lyr_mat,2)-1
Lyr_mat(4,i) = Lyr_mat(3,i)-Lyr_mat(2,i);
Lyr_mat(3,i+1) = Lyr_mat(4,i);
end
Lyr_mat(4,i+1)= -Total_Th/2;

assignin('base', 'Lyr_mat', Lyr_mat)

Lyr_mat
Total_Th

%% Calculation of Q
Q11=E1/(1-v12*v21);
Q22=E2/(1-v12*v21);
Q12=(v12*E2)/(1-v12*v21);
% Q44=G23;
% Q55=G13;
Q66=G12;



%% Calculation of Qbar for Each Lamina

Qb11(1,:) =
(Q11.*(cosd(Lyr_mat(1,:))).^4)+2.*((Q12+2.*Q66).*((sind(Lyr_mat(1,:))).^2)).*((cosd(Lyr_mat(1,:))).^2
)+(Q22.*(sind(Lyr_mat(1,:))).^4);

Qb22(1,:) =
(Q11.*(sind(Lyr_mat(1,:))).^4)+2.*((Q12+2.*Q66).*((sind(Lyr_mat(1,:))).^2)).*((cosd(Lyr_mat(1,:))).^2
)+(Q22.*(cosd(Lyr_mat(1,:))).^4);
```

```
Qb12(1,:) = ((Q11+Q22-
4.*Q66).*((sind(Lyr_mat(1,:))).^2).*(cosd(Lyr_mat(1,:))).^2)+(Q12.*(((sind(Lyr_mat(1,:))).^4)+(cosd(L
yr_mat(1,:))).^4));

Qb16(1,:) = sind(Lyr_mat(1,:)).*cosd(Lyr_mat(1,:)).*(cosd(Lyr_mat(1,:)).^2.*Q11-
sind(Lyr_mat(1,:)).^2.*Q22-(cosd(Lyr_mat(1,:)).^2-sind(Lyr_mat(1,:)).^2).*(Q12+2.*Q66));

Qb26(1,:) = sind(Lyr_mat(1,:)).*cosd(Lyr_mat(1,:)).*(sind(Lyr_mat(1,:)).^2.*Q11-
cosd(Lyr_mat(1,:)).^2.*Q22-(sind(Lyr_mat(1,:)).^2-cosd(Lyr_mat(1,:)).^2).*(Q12+2.*Q66));

Qb66(1,:) = (((Q11+Q22-
2.*Q12).*((sind(Lyr_mat(1,:))).^2).*(cosd(Lyr_mat(1,:))).^2)+(Q66.*(((cosd(Lyr_mat(1,:))).^2)-
(sind(Lyr_mat(1,:))).^2).^2));

% Qb55(1,:) = Q55.*cosd(Lyr_mat(1,:)).^2+Q44.*sind(Lyr_mat(1,:)).^2;

n_Q=2;

switch n_Q
   case 1
% 3D Stress-Strain
Qbar(1,1,:) = Qb11(:);
Qbar(2,2,:) = Qb22(:);
Qbar(1,2,:) = Qb12(:);
Qbar(2,1,:) = Qb12(:);
Qbar(1,3,:) = Qb16(:);
Qbar(3,1,:) = Qb16(:);
Qbar(2,3,:) = Qb26(:);
Qbar(3,2,:) = Qb26(:);
Qbar(3,3,:) = Qb66(:);

   case 2
%   Plain Stress Condition
Qbar(1,1,:) = Qb11(:)-(Qb12(:).^2)./Qb22(:);
Qbar(2,2,:) = 0;
Qbar(1,2,:) = 0;
Qbar(2,1,:) = 0;
Qbar(1,3,:) = Qb16(:)-(Qb12(:).*Qb26(:))./Qb22(:);
```

```
Qbar(3,1,:) = Qb16(:)-(Qb12(:).*Qb26(:))./Qb22(:);
Qbar(2,3,:) = 0;
Qbar(3,2,:) = 0;
Qbar(3,3,:) = Qb66(:)-(Qb26(:).^2)./Qb22(:);

   case 3
%   Plain Strain Condition
Qbar(1,1,:) = Qb11(:);
Qbar(2,2,:) = 0;
Qbar(1,2,:) = 0;
Qbar(2,1,:) = 0;
Qbar(1,3,:) = Qb16(:);
Qbar(3,1,:) = Qb16(:);
Qbar(2,3,:) = 0;
Qbar(3,2,:) = 0;
Qbar(3,3,:) = Qb66(:);

end


%% Calculation of ABD

A=zeros(3);
B=zeros(3);
D=zeros(3);
A55=0;
I0=0;
I1=0;
I2=0;

for i=1:1:size(Lyr_mat,2);
   A=A+(Qbar(:,:,(i)).*(Lyr_mat(3,i)-Lyr_mat(4,i)));
   B=B+((Qbar(:,:,(i)).*((Lyr_mat(3,i))^2-(Lyr_mat(4,i))^2))/2);
   D=D+((Qbar(:,:,(i)).*((Lyr_mat(3,i))^3-(Lyr_mat(4,i))^3))/3);

%     A55=A55+(Qb55(1,(i)).*(Lyr_mat(3,i)-Lyr_mat(4,i)));

   I0=I0+ro*(Lyr_mat(3,i)-Lyr_mat(4,i));
   I1=I1+ro*((Lyr_mat(3,i))^2-(Lyr_mat(4,i))^2)/2;
```

```
   I2=I2+ro*((Lyr_mat(3,i))^3-(Lyr_mat(4,i))^3)/3;
end

%   ABD Matrice Elements
A11=A(1,1);
A16=A(1,3);
A66=A(3,3);

B11=B(1,1);
B16=B(1,3);
B66=B(3,3);

D11=D(1,1);
D16=D(1,3);
D66=D(3,3);

ABD = zeros(6,6);
ABD(1:3,1:3)=A; ABD(1:3,4:6)=B;
ABD(4:6,1:3)=B; ABD(4:6,4:6)=D;

assignin('base' ,'ABD', ABD);

%% Find Center of Gravity

TLW=0;      %Total Line Weight
TAX=0;      %Total A.Moment X dir
TAY=0;      %Total A.Moment Y dir

for i=1:size(El_mat,1)-1;
   P1=[El_mat(i,3),El_mat(i,4)];
   P2=[El_mat(i+1,3),El_mat(i+1,4)];
   mid_X=(P1(1,1)+P2(1,1))/2;          %mid point X
   mid_Y=(P1(1,2)+P2(1,2))/2;          %mid point Y
   El_mat(i+1,7) = distP1P2(P1,P2);    %dist. between points
   El_mat(i+1,8) = atang(P1,P2);       %line angle
   TLW=TLW+El_mat(i+1,7);              %Total Line Weight
   TAX=TAX+mid_X*El_mat(i+1,7);        %Total A.Moment X dim
   TAY=TAY+mid_Y*El_mat(i+1,7);        %Total A.Moment Y dim
end
```

```
CG(1,1)=TAX/TLW;    %Center of Gravity X
CG(1,2)=TAY/TLW;    %Center of Gravity Y

%Move CG to origin
El_mat(:,3)=El_mat(:,3)-CG(1,1);   %move X dir
El_mat(:,4)=El_mat(:,4)-CG(1,2);   %move Y dir

   % Zero Fix
for i=1:1:size(El_mat);

   if abs(El_mat(i,3))<1e-12
      El_mat(i,3)=0;
   end

   if abs(El_mat(i,4))<1e-12
      El_mat(i,4)=0;
   end
end

%plot at origin
figure('Name','TWB Cross Section and Center of Gravity');
line(El_mat(:,3),El_mat(:,4),'color','r','LineWidth',3);
hold on
plot(0,0,'+','MarkerEdgecolor','b','MarkerSize',100)
%plot(El_mat(:,3),El_mat(:,4),'*');

plot(El_mat(1,3),El_mat(1,4),'o');
plot(El_mat(2,3),El_mat(2,4),'+');

xlabel('(mm)')
ylabel('(mm)')
axis equal
hold off

clear P1 P2 mid_X mid_Y TAX TAY CG

%% Calculation of X Y Q R W S Theta
size_El_mat = size(El_mat,1);
```

```
% Calculation of X Y Q R S
S=zeros(size_El_mat+1,1);
X=zeros(size_El_mat,2);
Y=zeros(size_El_mat,2);
Q=zeros(size_El_mat,2);
R=zeros(size_El_mat,2);
W=zeros(size_El_mat,1);
Ai=zeros(size_El_mat,1);

theta(:,1) = El_mat(:,8); %line degree constant

for i=1:size_El_mat-1

   x1=El_mat(i,3)/1000;              %constant
   y1=El_mat(i,4)/1000;              %constant

   X(i+1,:)=[ cosd(theta(i+1)) x1 ];      %polynomal !!!
   Y(i+1,:)=[ sind(theta(i+1)) y1 ];      %polynomal !!!

   R(i+1,:)=polysum(+X(i+1,:).*cosd(theta(i+1)+90),'+',Y(i+1,:).*sind(theta(i+1)+90));   %polynomal !!!

   Q(i+1,:)=polysum(+X(i+1,:).*sind(theta(i+1)+90),'-',Y(i+1,:).*cosd(theta(i+1)+90));    %polynomal !!!

   S(i+2,:)=S(i+1,:)+El_mat(i+1,7)/1000;     %constant

   Ai(i+1,1)=R(i+1,2).*(S(i+2,1)-S(i+1,1));  %constant

end

% Calculation of Shear Flow
Asum=sum(Ai,1);
Ssum=S(size(El_mat,1)+1,1);
F_t=Asum./Ssum;

% Open Cross Section Check

if El_mat(1,3)>=(El_mat(end,3)-abs(El_mat(end,3)/100)) &&
El_mat(1,3)<=(El_mat(end,3)+abs(El_mat(end,3)/100)) &&...
```

```
   El_mat(1,4)>=(El_mat(end,4)-abs(El_mat(end,4)/100)) &&
El_mat(1,4)<=(El_mat(end,4)+abs(El_mat(end,4)/100));

    disp('-- close cross section --')
    disp(' ')

else

    F_t=0;
    disp('!!! open cross section !!!')
    disp(' ')

end

% Calculation of Warping Function
Fs=zeros(size_El_mat,2);
Fss=zeros(size_El_mat,1);

for i=1:size_El_mat-1;
    ds=(S(i+1,1)-S(i,1));
    ds2=(S(i+2,1)-S(i+1,1));
    Fs(i+1,:)=polysum(polyval(Fs(i,:),ds),'+',[1 0].*(R(i+1,2)-F_t));
    W(i+1,1)=polyval(Fs(i+1,:),ds2);
    Fss(i+1,1)=polyval(polyint(Fs(i+1,:)),ds2);
end

Fssum=sum(Fss,1);
W(:,1)=W(:,1)-(Fssum/Ssum);

% assignin('base','Fss',Fss)
% assignin('base','Fs',Fs)
% assignin('base','S',S)
% assignin('base','W',W)
% assignin('base','A_El_mat',El_mat)

%% Calculation of Stiffness Matrice

E11=0; E12=0; E13=0; E14=0; E15=0;
```

```
E22=0; E23=0; E24=0; E25=0;

E33=0; E34=0; E35=0;

E44=0; E45=0;

E55=0;


for i=1:size_El_mat-1

   ds=(S(i+2,1)-S(i+1,1));

   %s(i+1)
   E11=E11+polyval(polyint(A11),ds);
   E12=E12+polyval(polyint(polysum(A11*X(i+1,:),'+',B11*sind(theta(i+1)))),ds);
   E13=E13+polyval(polyint(polysum(A11*Y(i+1,:),'-',B11*cosd(theta(i+1)))),ds);
   E14=E14+polyval(polyint(polysum(A11*W(i+1,1),'-',B11*Q(i+1,:))),ds);
   E15=E15+polyval(polyint((A16*F_t/2)+B16),ds);


E22=E22+polyval(polyint(polysum(A11*conv(X(i+1,:),X(i+1,:)),'+',2*B11*sind(theta(i+1))*X(i+1,:),'+',
D11*sind(theta(i+1))^2)),ds);

E23=E23+polyval(polyint(polysum(A11*conv(X(i+1,:),Y(i+1,:)),'+',B11*polysum(sind(theta(i+1))*Y(i+
1,:),'-',cosd(theta(i+1))*X(i+1,:)),'-',D11*sind(theta(i+1))*cosd(theta(i+1)))),ds);

E24=E24+polyval(polyint(polysum(A11*conv(X(i+1,:),W(i+1,:)),'+',B11*polysum(W(i+1,:)*sind(theta(i
+1)),'-',conv(Q(i+1,:),X(i+1,:))),'-',D11*sind(theta(i+1))*Q(i+1,:))),ds);

E25=E25+polyval(polyint(polysum(A16*F_t/2*X(i+1,:),'+',B16*polysum(X(i+1,:),'+',F_t/2*sind(theta(i+
1))),'+',D16*sind(theta(i+1)))),ds);

   E33=E33+polyval(polyint(polysum(A11*conv(Y(i+1,:),Y(i+1,:)),'-
',2*B11*cosd(theta(i+1))*Y(i+1,:),'+',D11*cosd(theta(i+1))^2)),ds);
   E34=E34+polyval(polyint(polysum(A11*conv(Y(i+1,:),W(i+1,:)),'-
',B11*polysum(W(i+1,:)*cosd(theta(i+1)),'+',conv(Q(i+1,:),Y(i+1,:))),'+',D11*cosd(theta(i+1))*Q(i+1,:)))
,ds);
```

```
    E35=E35+polyval(polyint(polysum(A16*F_t/2*Y(i+1,:),'+',B16*polysum(Y(i+1,:),'-
',F_t/2*cosd(theta(i+1))),'-',D16*cosd(theta(i+1)))),ds);

    E44=E44+polyval(polyint(polysum(A11*conv(W(i+1,:),W(i+1,:)),'-
',2*B11*conv(W(i+1,:),Q(i+1,:)),'+',D11*conv(Q(i+1,:),Q(i+1,:)))),ds);
    E45=E45+polyval(polyint(polysum(A16*F_t/2*W(i+1,:),'+',B16*polysum(W(i+1,:),'-
',F_t/2*Q(i+1,:)),'-',D16*Q(i+1,:))),ds);

    E55=E55+polyval(polyint(polysum(A66*(F_t^2)/4,'+',B66*F_t,'+',D66)),ds);



end

    E = zeros(5,5);
    E(1,1:5)= [E11 E12 E13 E14 E15];
    E(2,1:5)= [E12 E22 E23 E24 E25];
    E(3,1:5)= [E13 E23 E33 E34 E35];
    E(4,1:5)= [E14 E24 E34 E44 E45];
    E(5,1:5)= [E15 E25 E35 E45 E55];

    % assignin('base' ,'E', E);

    % E=evalin('base','E1');

%% Calculation of Inertia Coefficents
m0=0;
mC=0;
mP=0;
mS=0;
mX=0;
mY=0;
mW=0;
m2=0;

for i=1:size_El_mat-1

    ds=(S(i+2,1)-S(i+1,1));

    %s(i+1)
```

```
    m0=m0+polyval(polyint(I0),ds);
    mC=mC+polyval(polyint(I1*cosd(theta(i+1))),ds);
    mP=mP+polyval(polyint(polysum(conv(Q(i+1,:),Q(i+1,:)),'+',R(i+1,2).^2)),ds);
    mS=mS+polyval(polyint(I1*sind(theta(i+1))),ds);
    mX=mX+polyval(polyint(I0*X(i+1,:)),ds);
    mY=mY+polyval(polyint(I0*Y(i+1,:)),ds);
    mW=mW+polyval(polyint(I0*R(i+1,2)),ds);
    m2=m2+polyval(polyint(I2),ds);

end


%% Calculation of Element Stiffness and Mass Matrice

% Cross of Shape Functions
G1G1 = [Ls/3,   Ls/6;...
        Ls/6,   Ls/3];

F1F1 = [13*Ls/35,          9*Ls/70,        (11*Ls^2)/210,     (-13*Ls^2)/420;...
         9*Ls/70,          13*Ls/35,       (13*Ls^2)/420,     (-11*Ls^2)/210;...
       (11*Ls^2)/210,    (13*Ls^2)/420,       (Ls^3)/105,      (-1*Ls^3)/140;...
       (-13*Ls^2)/420,  (-11*Ls^2)/210,     (-1*Ls^3)/140,         (Ls^3)/105];

dG1G1=[1/Ls,   -1/Ls;...
      -1/Ls,    1/Ls];

dG1F1=[1/Ls,   -1/Ls,   0,   0;...
      -1/Ls,    1/Ls,   0,   0];

dG1F2=[0,   0,   1/Ls,  -1/Ls;...
       0,   0,  -1/Ls,   1/Ls];

dF1F1=[6/(5*Ls),  -6/(5*Ls),    1/10,     1/10;...
       -6/(5*Ls),  6/(5*Ls),   -1/10,    -1/10;...
        1/10,   -1/10   ,2*Ls/15,   -Ls/30;...
        1/10,   -1/10,    -Ls/30,  2*Ls/15];

dF1F2=[0,      0,      1/Ls,  -1/Ls;...
       0,      0,     -1/Ls,   1/Ls;...
```

```
    -1/Ls,   1/Ls,  -1/2,   -1/2;...
     1/Ls,  -1/Ls,   1/2,    1/2];

dF2F1=transpose(dF1F2);

dF2F2=[12/Ls^3,     -12/Ls^3,     6/Ls^2,     6/Ls^2;...
    -12/Ls^3,       12/Ls^3,    -6/Ls^2,    -6/Ls^2;...
      6/Ls^2,       -6/Ls^2,     4/Ls,       2/Ls;...
      6/Ls^2,       -6/Ls^2,     2/Ls,       4/Ls];

%Element Stiffness Matrix
K11=E(1,1)*dG1G1;
K12=-E(1,2)*dG1F2;
K13=-E(1,3)*dG1F2;
K14=2*E(1,5)*dG1F1-E(1,4)*dG1F2;

K22=E(2,2)*dF2F2;
K23=E(2,3)*dF2F2;
K24=E(2,4)*dF2F2-2*E(2,5)*dF2F1;

K33=E(3,3)*dF2F2;
K34=E(3,4)*dF2F2-2*E(3,5)*dF2F1;

K44=(E(4,4)*dF2F2-2*E(4,5)*(dF1F2+dF2F1)+4*E(5,5)*dF1F1);


% Generate K Matrice
K = TW_FEA_f_genK_mat(nm,K11,K12,K13,K14,...
                      K22,K23,K24,...
                      K33,K34,...
                      K44);

assignin('base','K',K);

%Element Mass Matrice
M11=m0*G1G1;
M12=zeros(2,4);
M13=zeros(2,4);
M14=zeros(2,4);
```

```
M22=m0*F1F1;
M23=zeros(4,4);
M24=(mC-mY+m0*0)*F1F1;

M33=m0*F1F1;
M34=(mS-mX-m0*0)*F1F1;

M44=(mP+m2+2*mW)*F1F1;

% Generate M Matrice
M = TW_FEA_f_genK_mat(nm,M11,M12,M13,M14,...
                      M22,M23,M24,...
                      M33,M34,...
                      M44);

assignin('base','M',M);


%% Boundry Conditions & Applied Forces and Moments

%BND

n=nm+1;

TW_B_data=ones(n,7);
TW_B_data([1 n],:)=TW_B_data([1 n],:)-bnd_G;


U_bc([1:n],1)=TW_B_data(:,1);
U_bc([n+1:2*n],1)=TW_B_data(:,2);
U_bc([2*n+1:3*n],1)=TW_B_data(:,3);
U_bc([3*n+1:4*n],1)=TW_B_data(:,4);
U_bc([4*n+1:5*n],1)=TW_B_data(:,5);
U_bc([5*n+1:6*n],1)=TW_B_data(:,6);
U_bc([6*n+1:7*n],1)=TW_B_data(:,7);

assignin('base','U_bc',U_bc);
```

```
%FORCE

TW_F_data=force_G;

%Force Shape Vector
F_bc_shape=1*ones(n,1);
F_bc_shape([1 n],1)=1;

F_bc(1:n,1)=F_bc_shape.*TW_F_data(:,1);
F_bc(n+1:2*n,1)=F_bc_shape.*TW_F_data(:,2);
F_bc(2*n+1:3*n,1)=F_bc_shape.*TW_F_data(:,3);
F_bc(3*n+1:4*n,1)=F_bc_shape.*TW_F_data(:,4);
F_bc(4*n+1:5*n,1)=F_bc_shape.*TW_F_data(:,5);
F_bc(5*n+1:6*n,1)=F_bc_shape.*TW_F_data(:,6);
F_bc(6*n+1:7*n,1)=F_bc_shape.*TW_F_data(:,7);

assignin('base','F_bc',F_bc);

%% Solution

sk=size(K,1);
ki=0;
skip(sk)=0;

for i=1:1:sk;
   if U_bc(i,1)==1;
      ki=ki+1;
      Ki(:,ki)=K(:,i);
      Mi(:,ki)=M(:,i);
   else
      skip(i)=i;
   end
end

ki=0;
Fs=0;

for i=1:1:sk;
```

```
   if U_bc(i,1)==1;
      ki=ki+1;
      Ks(ki,:)=Ki(i,:);
      Ms(ki,:)=Mi(i,:);
      Fs(ki,1)=F_bc(i,1);
   end
end

assignin('base','Ks',Ks);
assignin('base','Ms',Ms);
assignin('base','Fs',Fs);

% Find Displacements
Us=(Ks)\Fs;

ki=0;
for i=1:1:sk;
   if skip(1,i)==0;
   ki=ki+1;
   U(i,1)=Us(ki);
   else
   U(i,1)=0;
   end
end

assignin('base','U',U);

% Find Forces
F=K*U;

assignin('base','F',F);

% Save Function Variables to File
save TW_FEA_main_vars.mat

evalin('base','clear')

%% Vibration Solution
if P_opt(1,7) == 1
```

```
   show_vib='on';
else
   show_vib='off';
end

[Wmod_u,Wmod_v,Wmod_w,Wmod_f,Wn_u,Wn_v,Wn_w,Wn_f,Wn,Wmod]=TW_FEA_f_vib(Ks,Ms,
show_vib);


%% Plot

%Check Plot Options

   % Plot Static Results
   if P_opt(1,1) == 1
   TW_FEA_f_plot(U,'Deflection',1,'on')
   end

   % Plot Dynamic Results
   if P_opt(1,2) == 1
   TW_FEA_f_plot(Wmod_w(:,2),'Mod Shape (w)',0.1,'off')
   assignin('base','Wn',Wn)
   assignin('base','Wmod',Wmod);

   assignin('base','Wmod_w',Wmod_w);
   assignin('base','Wn_w',Wn_w);
   end

   if P_opt(1,3) == 1
   TW_FEA_f_plot(Wmod_u(:,2),'Mod Shape (u)',0.1,'off')
   assignin('base','Wn',Wn)
   assignin('base','Wmod',Wmod);

   assignin('base','Wmod_u',Wmod_u);
   assignin('base','Wn_u',Wn_u);
   end

   if P_opt(1,4) == 1
   TW_FEA_f_plot(Wmod_v(:,2),'Mod Shape (v)',0.1,'off')
```

```
    assignin('base','Wn',Wn)
    assignin('base','Wmod',Wmod);

    assignin('base','Wmod_v',Wmod_v);
    assignin('base','Wn_v',Wn_v);
    end

    if P_opt(1,5) == 1 && isempty(Wn_f)==0
    TW_FEA_f_plot(Wmod_f(:,2),'Mod Shape (f)',1,'off')
    assignin('base','Wn',Wn)
    assignin('base','Wmod',Wmod);

    assignin('base','Wmod_f',Wmod_f);
    assignin('base','Wn_f',Wn_f);
    end


%% Results

% Seperate Displacements
w=U(1:1:1*(nm+1));
u=U(1*(nm+1)+1:1:2*(nm+1));
du=U(2*(nm+1)+1:1:3*(nm+1));
v=U(3*(nm+1)+1:1:4*(nm+1));
dv=U(4*(nm+1)+1:1:5*(nm+1));
f=U(5*(nm+1)+1:1:6*(nm+1));
df=U(6*(nm+1)+1:1:7*(nm+1));

assignin('base' ,'AA_work_num', work_num);
assignin('base' ,'w', w);
assignin('base' ,'u', u);
assignin('base' ,'v', v);
assignin('base' ,'f', f);
assignin('base' ,'du', du);
assignin('base' ,'dv', dv);
assignin('base' ,'df', df);

toc
```

```
% Save Results
if P_opt(1,6) == 1
TW_FEA_f_save2file(w,u,du,v,dv,f,df,Wn_w,Wn_u,Wn_v,Wn_f,work_num);
end

end
```

TW_FEA_GUI (Kullanıcı Ara Yüzü Kodu)

```
function varargout = GUI_TW_FEA(varargin)
% GUI_TW_FEA MATLAB code for GUI_TW_FEA.fig
%      GUI_TW_FEA, by itself, creates a new GUI_TW_FEA or raises the existing
%      singleton*.
%
%      H = GUI_TW_FEA returns the handle to a new GUI_TW_FEA or the handle to
%      the existing singleton*.
%
%      GUI_TW_FEA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI_TW_FEA.M with the given input arguments.
%
%      GUI_TW_FEA('Property','Value',...) creates a new GUI_TW_FEA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_TW_FEA_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_TW_FEA_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_TW_FEA

% Last Modified by GUIDE v2.5 19-Jun-2013 18:25:35

%Create Plot Option Vector

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```

```
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_TW_FEA_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_TW_FEA_OutputFcn, ...
                   'gui_LayoutFcn',  [ ] , ...
                   'gui_Callback',   [ ]);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_TW_FEA is made visible.
function GUI_TW_FEA_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_TW_FEA (see VARARGIN)

% Choose default command line output for GUI_TW_FEA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_TW_FEA wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global filename

filename=[ ];

ec=str2num(get(handles.edit_el_count,'String'));
```

```
data_table=zeros(ec+1,7);
set(handles.uitable_force,'data',data_table)
set(handles.text_el_count_status,'string',num2str(ec+1))

data_table=zeros(2,7);
data_table(1,:)=1;
set(handles.uitable_bnd,'data',data_table)

data_table=zeros(1,7);
set(handles.dist_table,'data',data_table)

P_opt=[0 0 0 0 0 0 0];
work_num=0;
save('GUI_DATA','P_opt','work_num')

clc
evalin('base','clear')



% --- Outputs from this function are returned to the command line.
function varargout = GUI_TW_FEA_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%% GMG

% Read Data From GUI and Write To File

function read_write_GUI(hObject, eventdata, handles)

global filename pathname El_mat_G
```

```
set(handles.text_status,'String','Wait !!!');

E11_G=str2num(get(handles.edit_E11,'String'));
E22_G=str2num(get(handles.edit_E22,'String'));
G12_G=str2num(get(handles.edit_G12,'String'));
v12_G=str2num(get(handles.edit_v12,'String'));
v21_G=str2num(get(handles.edit_v21,'String'));
ro_G =str2num(get(handles.edit_ro, 'String'));

ec=str2num(get(handles.edit_el_count,'String'));
L_G=str2num(get(handles.edit_length,'String'));

angle_G=(get(handles.edit_angle,'String'));
eval(strcat('angle_G=[',angle_G,'];'));
thickness_G=(get(handles.edit_thickness,'String'));
eval(strcat('thickness_G=[',thickness_G,'];'));

bnd_G=get(handles.uitable_bnd,'data');
force_G=get(handles.uitable_force,'data');
dist_table_G=get(handles.dist_table,'data');

set(handles.text_el_count_status,'string',num2str(ec+1))
set(handles.text_status,'String','Data Stored');

CCinv = get(handles.checkbox_CCinv,'Value');

save('GUI_DATA','E11_G','E22_G','G12_G','v12_G','v21_G','ro_G','ec','L_G','angle_G','dist_table_G',...
    'thickness_G','bnd_G','force_G','filename','pathname','El_mat_G','CCinv','-append');

% GMG end

% --- Executes on button press in pushbutton_def.
function pushbutton_def_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_def (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text_status,'String','Wait');
```

```
set(handles.edit_E11,'String','142');
set(handles.edit_E22,'String','9.8');
set(handles.edit_G12,'String','6');
set(handles.edit_v12,'String','0.42');
set(handles.edit_v21,'String','0.02');
set(handles.edit_ro,'String','1445');

set(handles.edit_el_count,'String','20');
set(handles.edit_length,'String','1');

set(handles.edit_angle,'String','0 0 0 0 0 45 45 45 45 45');
set(handles.edit_thickness,'String','0.2');

data_table=zeros(2,7);
data_table(1,:)=1;
set(handles.uitable_bnd,'data',data_table)

data_table(1,:)=0;
data_table(21,:)=[2400 0 0 0 0 0 0];
set(handles.uitable_force,'data',data_table);

data_table=zeros(1,7);
set(handles.dist_table,'data',data_table)

read_write_GUI(hObject, eventdata, handles)

% --- Executes on button press in pushbutton_start_FEA.
function pushbutton_start_FEA_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_start_FEA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global filename pathname El_mat_G

if isempty(filename)==1
   errordlg('Cross Section is not selected. Press "Cross Section" and Sellect a Cross Section File
','error');return
end
```

```
El_mat_G=TW_FEA_DXF(filename,pathname);

read_write_GUI(hObject, eventdata, handles)

load('GUI_DATA.mat','work_num')
work_num=work_num+1;
save('GUI_DATA','work_num','-append')

TW_FEA_Main


% --- Executes on button press in pushbutton_cross_section.
function pushbutton_cross_section_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_cross_section (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global filename pathname

[filename, pathname] = uigetfile({'*.dxf'},'Open File','Multiselect','off'); % choose file to open

if filename==0
   filename=[ ];
   errordlg('No File Selected','error');return
end

set(handles.text_CS_status,'String',filename);
set(handles.text_CS_path,'String',pathname);

read_write_GUI(hObject, eventdata, handles)


function edit_E11_Callback(hObject, eventdata, handles)
% hObject    handle to edit_E11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_E11 as text
%        str2double(get(hObject,'String')) returns contents of edit_E11 as a double
```

```
read_write_GUI(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function edit_E11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_E11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function edit_E22_Callback(hObject, eventdata, handles)
% hObject    handle to edit_E22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_E22 as text
%        str2double(get(hObject,'String')) returns contents of edit_E22 as a double

read_write_GUI(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function edit_E22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_E22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
```

```
end



function edit_G12_Callback(hObject, eventdata, handles)
% hObject    handle to edit_G12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_G12 as text
%        str2double(get(hObject,'String')) returns contents of edit_G12 as a double

read_write_GUI(hObject, eventdata, handles)



% --- Executes during object creation, after setting all properties.
function edit_G12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_G12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit_v12_Callback(hObject, eventdata, handles)
% hObject    handle to edit_v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_v12 as text
%        str2double(get(hObject,'String')) returns contents of edit_v12 as a double

read_write_GUI(hObject, eventdata, handles)
```

```
% --- Executes during object creation, after setting all properties.
function edit_v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function edit_v21_Callback(hObject, eventdata, handles)
% hObject    handle to edit_v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_v21 as text
%        str2double(get(hObject,'String')) returns contents of edit_v21 as a double

read_write_GUI(hObject, eventdata, handles)



% --- Executes during object creation, after setting all properties.
function edit_v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function edit_ro_Callback(hObject, eventdata, handles)
% hObject    handle to edit_ro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_ro as text
%        str2double(get(hObject,'String')) returns contents of edit_ro as a double

read_write_GUI(hObject, eventdata, handles)



% --- Executes during object creation, after setting all properties.
function edit_ro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_ro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function edit_el_count_Callback(hObject, eventdata, handles)
% hObject    handle to edit_el_count (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_el_count as text
%        str2double(get(hObject,'String')) returns contents of edit_el_count as a double

ec=str2num(get(handles.edit_el_count,'String'));
set(handles.text_el_count_status,'string',num2str(ec+1))

data_0=zeros(ec+1,7);
set(handles.uitable_force,'data',data_0)
set(handles.text_el_count_status,'string',num2str(ec+1))
```

```
data_0=zeros(2,7);
data_0(1,:)=1;
set(handles.uitable_bnd,'data',data_0)

data_table=zeros(1,7);
set(handles.dist_table,'data',data_table)

read_write_GUI(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function edit_el_count_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_el_count (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function edit_length_Callback(hObject, eventdata, handles)
% hObject    handle to edit_length (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_length as text
%        str2double(get(hObject,'String')) returns contents of edit_length as a double

read_write_GUI(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function edit_length_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_length (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function edit_thickness_Callback(hObject, eventdata, handles)
% hObject    handle to edit_thickness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_thickness as text
%        str2double(get(hObject,'String')) returns contents of edit_thickness as a double

read_write_GUI(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function edit_thickness_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_thickness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function edit_angle_Callback(hObject, eventdata, handles)
% hObject    handle to edit_angle (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_angle as text
%        str2double(get(hObject,'String')) returns contents of edit_angle as a double

read_write_GUI(hObject, eventdata, handles)


% --- Executes during object creation, after setting all properties.
function edit_angle_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_angle (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in checkbox_CCinv.
function checkbox_CCinv_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_CCinv (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_CCinv

read_write_GUI(hObject, eventdata, handles)


% --- Executes when entered data in editable cell(s) in uitable_bnd.
function uitable_bnd_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable_bnd (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%           Indices: row and column indices of the cell(s) edited
%           PreviousData: previous data for the cell(s) edited
%           EditData: string(s) entered by the user
```

```
%	NewData: EditData or its converted form set on the Data property. Empty if Data was not
changed
%	Error: error string when failed to convert EditData to appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)

read_write_GUI(hObject, eventdata, handles)


% --- Executes when entered data in editable cell(s) in uitable_force.
function uitable_force_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable_force (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%	Indices: row and column indices of the cell(s) edited
%	PreviousData: previous data for the cell(s) edited
%	EditData: string(s) entered by the user
%	NewData: EditData or its converted form set on the Data property. Empty if Data was not
changed
%	Error: error string when failed to convert EditData to appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)

read_write_GUI(hObject, eventdata, handles)


% --- Executes on button press in Save_Data_Button.
function Save_Data_Button_Callback(hObject, eventdata, handles)
% hObject    handle to Save_Data_Button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.text_status,'String','Wait !!!');

[FileName,PathName] = uiputfile('*.mat','Save Data As');

Data_E11=str2num(get(handles.edit_E11,'String'));
Data_E22=str2num(get(handles.edit_E22,'String'));
Data_G12=str2num(get(handles.edit_G12,'String'));
Data_v12=str2num(get(handles.edit_v12,'String'));
Data_v21=str2num(get(handles.edit_v21,'String'));
Data_ro=str2num(get(handles.edit_ro, 'String'));
```

```
Data_ec=str2num(get(handles.edit_el_count,'String'));
Data_L=str2num(get(handles.edit_length,'String'));

Data_ang=(get(handles.edit_angle,'String'));
Data_thc=(get(handles.edit_thickness,'String'));

save(strcat(PathName,FileName),'Data_E11','Data_E22','Data_G12','Data_v12','Data_v21','Data_ro','Data
_ec','Data_L','Data_ang','Data_thc')

set(handles.text_status,'String','Data Saved');



% --- Executes on button press in Load_Data_Button.
function Load_Data_Button_Callback(hObject, eventdata, handles)
% hObject    handle to Load_Data_Button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%choose data file
global data_filename data_pathname

[data_filename, data_pathname] = uigetfile({'*.mat'},'Open File','Multiselect','off'); % choose file to open

if data_filename==0
   data_filename=[ ];
   errordlg('No File Selected','error');return
end

%read data
load(strcat(data_pathname,data_filename));

%set new data
set(handles.text_status,'String','Wait');

set(handles.edit_E11,'String',Data_E11);
set(handles.edit_E22,'String',Data_E22);
set(handles.edit_G12,'String',Data_G12);
set(handles.edit_v12,'String',Data_v12);
```

```
set(handles.edit_v21,'String',Data_v21);
set(handles.edit_ro,'String',Data_ro);

set(handles.edit_el_count,'String',Data_ec);
set(handles.edit_length,'String',Data_L);

set(handles.edit_angle,'String',Data_ang);
set(handles.edit_thickness,'String',Data_thc);

%refine tables
data_table=zeros(Data_ec+1,7);
set(handles.uitable_force,'data',data_table)
set(handles.text_el_count_status,'string',num2str(Data_ec+1))

data_table=zeros(2,7);
data_table(1,:)=1;
set(handles.uitable_bnd,'data',data_table)

data_table=zeros(1,7);
set(handles.dist_table,'data',data_table)

set(handles.text_status,'String','Data Loaded');

%write GUI data
read_write_GUI(hObject, eventdata, handles)


% --- Executes when entered data in editable cell(s) in dist_table.
function dist_table_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to dist_table (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%	Indices: row and column indices of the cell(s) edited
%	PreviousData: previous data for the cell(s) edited
%	EditData: string(s) entered by the user
%	NewData: EditData or its converted form set on the Data property. Empty if Data was not
changed
%	Error: error string when failed to convert EditData to appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)
```

```
dist_table=get(handles.dist_table,'data');
nm=str2num(get(handles.edit_el_count,'String'));
n=nm+1;

F_dist(1:n,1)=dist_table(1,1)/nm;
F_dist(1:n,2)=dist_table(1,2)/nm;
F_dist(1:n,3)=dist_table(1,3)/nm;
F_dist(1:n,4)=dist_table(1,4)/nm;
F_dist(1:n,5)=dist_table(1,5)/nm;
F_dist(1:n,6)=dist_table(1,6)/nm;
F_dist(1:n,7)=dist_table(1,7)/nm;

set(handles.uitable_force,'data',F_dist);

%write GUI data
read_write_GUI(hObject, eventdata, handles)


% --- Executes on button press in plot_opt_button.
function plot_opt_button_Callback(hObject, eventdata, handles)
% hObject    handle to plot_opt_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

GUI_TW_FEA_Options


% --- Executes on button press in close_plot_button.
function close_plot_button_Callback(hObject, eventdata, handles)
% hObject    handle to close_plot_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

all_figs = findobj(0, 'type', 'figure');
delete(all_figs);
```

GUI_FEA_Options (Kullanıcı Arayüzü Ayarlar Menüsü)

```
function varargout = GUI_TW_FEA_Options(varargin)
% GUI_TW_FEA_OPTIONS MATLAB code for GUI_TW_FEA_Options.fig
%      GUI_TW_FEA_OPTIONS, by itself, creates a new GUI_TW_FEA_OPTIONS or raises the
existing
%      singleton*.
%
%      H = GUI_TW_FEA_OPTIONS returns the handle to a new GUI_TW_FEA_OPTIONS or the
handle to
%      the existing singleton*.
%
%      GUI_TW_FEA_OPTIONS('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI_TW_FEA_OPTIONS.M with the given input arguments.
%
%      GUI_TW_FEA_OPTIONS('Property','Value',...) creates a new GUI_TW_FEA_OPTIONS or raises
the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_TW_FEA_Options_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_TW_FEA_Options_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_TW_FEA_Options

% Last Modified by GUIDE v2.5 20-Jun-2013 00:51:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_TW_FEA_Options_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_TW_FEA_Options_OutputFcn, ...
                   'gui_LayoutFcn',  [ ] , ...
                   'gui_Callback',   [ ]);
```

```
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_TW_FEA_Options is made visible.
function GUI_TW_FEA_Options_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_TW_FEA_Options (see VARARGIN)

% Choose default command line output for GUI_TW_FEA_Options
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_TW_FEA_Options wait for user response (see UIRESUME)
% uiwait(handles.figure1);

load('GUI_DATA.mat','work_num')
set(handles.text_work_num,'String',num2str(work_num));


% --- Outputs from this function are returned to the command line.
function varargout = GUI_TW_FEA_Options_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton_OK.
function pushbutton_OK_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_OK (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

P_opt(1,1)=get(handles.checkbox_static_def,'Value');
P_opt(1,2)=get(handles.checkbox_Wn_w,'Value');
P_opt(1,3)=get(handles.checkbox_Wn_u,'Value');
P_opt(1,4)=get(handles.checkbox_Wn_v,'Value');
P_opt(1,5)=get(handles.checkbox_Wn_f,'Value');
P_opt(1,6)=get(handles.checkbox_save2file,'Value');
P_opt(1,7)=get(handles.checkbox_show_vib,'Value');

if P_opt(1,6) == 1
[FileRes,PathRes] = uiputfile('*.xlsx','Save Results As');
save('GUI_DATA','FileRes','PathRes','-append')
end

save('GUI_DATA','P_opt','-append')

close

% --- Executes on button press in pushbutton_cancel.
function pushbutton_cancel_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_cancel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

close


% --- Executes on button press in checkbox_static_def.
function checkbox_static_def_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to checkbox_static_def (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_static_def


% --- Executes on button press in checkbox_Wn_w.
function checkbox_Wn_w_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_Wn_w (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_Wn_w


% --- Executes on button press in checkbox_Wn_u.
function checkbox_Wn_u_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_Wn_u (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_Wn_u


% --- Executes on button press in checkbox_Wn_v.
function checkbox_Wn_v_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_Wn_v (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_Wn_v


% --- Executes on button press in checkbox_Wn_f.
function checkbox_Wn_f_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_Wn_f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox_Wn_f


% --- Executes on button press in checkbox_save2file.
function checkbox_save2file_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_save2file (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_save2file


% --- Executes on button press in checkbox_show_vib.
function checkbox_show_vib_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_show_vib (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_show_vib


% --- Executes on button press in checkbox3.
function checkbox3_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox3


% --- Executes on button press in checkbox4.
function checkbox4_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox4
```

```
% --- Executes on button press in checkbox5.
function checkbox5_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox5



% --- Executes on button press in pushbutton_Reset.
function pushbutton_Reset_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_Reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

work_num=0;
save('GUI_DATA','work_num','-append')
set(handles.text_work_num,'String',num2str(work_num));
```

TW_FEA_DXF (DXF Dosya Okuyucu)

```
function [El_mat]=TW_FEA_DXF(filename,pathname)

%% Read file

   fId = fopen(strcat(pathname,filename));
   c_ValAsoc = textscan(fId,'%d%s','Delimiter','\n');
   fclose(fId);
   % Code Group Matrix
   m_GrCode = c_ValAsoc{1};
   % Associated value String Cell
   c_ValAsoc = c_ValAsoc{2};
   %[m_GrCode,c_ValAsoc] = c_ValAsoc{:};

   load('GUI_DATA','CCinv');

%% DXF Entities
```

```
m_PosCero = find(m_GrCode==0);
%Is searched by (0,SECTION),(2,ENTITIES)
indInSecEnt = strmatch('ENTITIES',c_ValAsoc(m_PosCero(1:end-1)+1),'exact');
%(0,ENDSEC)
m_indFinSecEnt = strmatch('ENDSEC',c_ValAsoc(m_PosCero(indInSecEnt:end)),'exact');
% Entities Position
m_PosCero = m_PosCero(indInSecEnt:indInSecEnt-1+m_indFinSecEnt(1));
% Variable initiation
c_Line = cell(1,2);
c_Poly = cell(1,2);
c_Cir = cell(1,2);
c_Arc = cell(1,2);
c_Poi = cell(1,2);
%
iLine = 1;
iPoly = 1;
iCir = 1;
iArc = 1;
iPoi = 1;
isay = 1;
% Loop on the Entities
for iEnt = 1:length(m_PosCero)-2
   m_GrCodeEnt = m_GrCode(m_PosCero(iEnt+1):m_PosCero(iEnt+2)-1);
   c_ValAsocEnt = c_ValAsoc(m_PosCero(iEnt+1):m_PosCero(iEnt+2)-1);
   nomEnt = c_ValAsocEnt{1};  %c_ValAsocEnt{m_PosCero(iEnt+1)}
   %In the entitie's name is assumed uppercase
   switch nomEnt
      case 'LINE'
         % (Xi,Yi,Zi,Xj,Yj,Zj) start and end points
         c_Line{iLine,1} = [isay,str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
            str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...
            str2double(f_ValGrCode(30,m_GrCodeEnt,c_ValAsocEnt)),...
            str2double(f_ValGrCode(11,m_GrCodeEnt,c_ValAsocEnt)),...
            str2double(f_ValGrCode(21,m_GrCodeEnt,c_ValAsocEnt)),...
            str2double(f_ValGrCode(31,m_GrCodeEnt,c_ValAsocEnt))];
         % Layer
         c_Line(iLine,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);
         % Color
         %if no exist is ByLayer (256)
```

```
    %c_Line{iLine,3} = str2double(f_ValGrCode(62,m_GrCodeEnt,c_ValAsocEnt));
    % XData
    %c_Line(iLine,4) = f_XData(GroupCode,'XDataName',m_GrCodeEnt,c_ValAsocEnt);
    % Add properties
    %
    iLine = iLine+1;
    isay=isay+1;
  case 'CIRCLE'
    % (X Center,Y Center,Radius)
    c_Cir{iCir,1} = [isay,str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(40,m_GrCodeEnt,c_ValAsocEnt))];
    % Layer
    c_Cir(iCir,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);
    % Add properties
    %
    iCir = iCir+1;
    isay=isay+1;
  case 'ARC'
    % (X Center,Y Center,Radius,Start angle,End angle)
    c_Arc{iArc,1} = [isay,str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(40,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(50,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(51,m_GrCodeEnt,c_ValAsocEnt))];
    % Layer
    c_Arc(iArc,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);
    % Add properties
    %
    iArc = iArc+1;
    isay=isay+1;
  case 'POINT'
    % (X,Y,Z) Position
    c_Poi{iPoi,1} = [isay,str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...
       str2double(f_ValGrCode(30,m_GrCodeEnt,c_ValAsocEnt))];
    % Layer
    c_Poi(iPoi,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);
    % Add properties
```

```
            %
            iPoi = iPoi+1;
            isay=isay+1;
          %case Add Entities
       end
    end

clear c_ValAsoc c_ValAsocEnt fId filename iArc iCir iEnt iLine iPoi iPoly indInSecEnt m_GrCode
m_GrCodeEnt m_PosCero m_indFinSecEnt nomEnt pathname

%% Accurate Data form Cells
cn_Arc(1,6)=0;
cn_Cir(1,4)=0;
cn_Line(1,7)=0;
cn_Poi(1,4)=0;

for i=1:1:size(c_Arc,1);
   cn_Arc(i,:)=c_Arc{i,1};
end

for i=1:1:size(c_Cir,1);
   cn_Cir(i,:)=c_Cir{i,1};
end

for i=1:1:size(c_Line,1);
   cn_Line(i,:)=c_Line{i,1};
end

for i=1:1:size(c_Poi,1);
   cn_Poi(i,:)=c_Poi{i,1};

end
if isempty(cn_Line)==0
   for i=1:1:size(c_Line,1);
      cn_Line(i,8)=distP1P2(cn_Line(i,2:3),cn_Line(i,5:6));
   end
end

clear ans i isay c_Arc c_Cir c_Line c_Poi c_Poly
```

```
%% Generate Points

El_mat=entity2point(cn_Poi,cn_Line,cn_Arc,cn_Cir);

r=1;
El_mat_order=El_mat;
El_mat=El_mat_order(1,:);

for i=1:size(El_mat_order,1)-1;
   if El_mat_order(i,3)==El_mat_order(i+1,3) && El_mat_order(i,4)==El_mat_order(i+1,4);
      %skip
   else
      El_mat=vertcat(El_mat,El_mat_order(i+1,:));
      r=r+1;
   end
end

if CCinv==1
   El_mat=flipud(El_mat);
end

assignin('base','A_El_mat',El_mat)

end



%% entity2point

function [El_mat]=entity2point(cn_Poi,cn_Line,cn_Arc,cn_Cir)

ent_case=1;
if isempty(cn_Cir)==0
ent_case=2;
end

switch ent_case
```

```
    case 1


%Generate Points
if isempty(cn_Poi)==0
k=1;

p_Poi=zeros(size(cn_Poi,1),4);

for i=1:size(cn_Poi,1)

    p_Poi(k,1)=cn_Poi(i,1);
    p_Poi(k,2)=1;
    p_Poi(k,3)=cn_Poi(i,2);
    p_Poi(k,4)=cn_Poi(i,3);

    k=k+1;

end

else
    p_Poi=[ ];

end


%Generate Line Points
if isempty(cn_Line)==0
k=1;

p_Line=zeros(2*size(cn_Line,1),4);

for i=1:size(cn_Line,1)

    p_Line(k,1)=cn_Line(i,1);
    p_Line(k,2)=1;
    p_Line(k,3)=cn_Line(i,2);
    p_Line(k,4)=cn_Line(i,3);
    p_Line(k+1,1)=cn_Line(i,1);
```

```
    p_Line(k+1,2)=2;
    p_Line(k+1,3)=cn_Line(i,5);
    p_Line(k+1,4)=cn_Line(i,6);

    k=k+2;

end

else
    p_Line=[ ];

end


%Generate Arc Points
if isempty(cn_Arc)==0

%segment lenght
s_len=max(cn_Arc(:,4))/10;

k=0;
for i=1:size(cn_Arc,1)
    if cn_Arc(i,6)>cn_Arc(i,5)                    %arc angle
        cn_Arc(i,8)=abs((cn_Arc(i,5)-cn_Arc(i,6)));
    else
        cn_Arc(i,8)=360-abs((cn_Arc(i,5)-cn_Arc(i,6)));
    end
    cn_Arc(i,7)=2*pi*cn_Arc(i,4)*cn_Arc(i,8)/360;       %arc leght
    cn_Arc(i,9)=abs(round(cn_Arc(i,7)/s_len));          %segment count
    angle0=cn_Arc(i,8)/cn_Arc(i,9);                     %increament angle
    angleP=cn_Arc(i,6)+angle0;                          %start angle


    for j=1:cn_Arc(i,9)+1
        k=k+1;
        p_Arc(k,1)=cn_Arc(i,1);
        p_Arc(k,2)=j;   %index
        angleP=mod((angleP-angle0),360);
        p_Arc(k,3)=cn_Arc(i,4)*cosd(angleP)+cn_Arc(i,2);
```

```
        p_Arc(k,4)=cn_Arc(i,4)*sind(angleP)+cn_Arc(i,3);
    end

end

else
    p_Arc=[ ];

end

% Generate El_mat
El_mat=vertcat(p_Poi,p_Line,p_Arc);
El_mat=sortrows(El_mat,1);



    case 2

% Generate Circle Points
if isempty(cn_Cir)==0
k=0;

    %Segment Count
    seg_Cir=36;
    angle0=360/seg_Cir;
    angleP=90+angle0;

i=1;

    for j=1:seg_Cir+1
        k=k+1;
        p_Cir(k,1)=1;
        p_Cir(k,2)=j;
        angleP=mod((angleP-angle0),360);
        p_Cir(k,3)=cn_Cir(i,4)*cosd(angleP)+cn_Cir(i,2);
        p_Cir(k,4)=cn_Cir(i,4)*sind(angleP)+cn_Cir(i,3);
    end

else
    p_Cir=[ ];
```

```
end

% Generate El_mat
El_mat=p_Cir;
El_mat=sortrows(El_mat,1);

end

clear angle0 angleP ans c_Arc c_Cir c_Line c_Poi c_Poly i isay j k s_len

El_mat

% Zero Check
for j=1:size(El_mat,1)

   if El_mat(j,3)<=1e-12 && El_mat(j,3)>=-1e-12
      El_mat(j,3)=0;
   end

   if El_mat(j,4)<=1e-12 && El_mat(j,4)>=-1e-12
      El_mat(j,4)=0;
   end
end

iEl=1;
k=0;

for j=1:size(El_mat,1)

   if El_mat(j,1)==iEl
      k=k+1;
      El_1(k,:)=El_mat(j,:);
   end

end


El_mat_order=El_1;
```

```
for i=1:max(El_mat(:,1))

k=0;
l=0;
El_1=[ ];
El_2=[ ];

for j=1:size(El_mat,1)

   if El_mat(j,1)==iEl
      k=k+1;
      El_1(k,:)=El_mat(j,:);
   elseif El_mat(j,1)==iEl+1
      l=l+1;
      El_2(l,:)=El_mat(j,:);
   end

end

   maxEl=size(El_1,1);

   if isempty(El_2)==1
      return
   else

if El_2(1,3)>=(El_1(maxEl,3)-abs(El_1(maxEl,3)/50)) &&
El_2(1,3)<=(El_1(maxEl,3)+abs(El_1(maxEl,3)/50)) &&...
  El_2(1,4)>=(El_1(maxEl,4)-abs(El_1(maxEl,4)/50)) &&
El_2(1,4)<=(El_1(maxEl,4)+abs(El_1(maxEl,4)/50));

   El_mat_order=vertcat(El_mat_order,El_2);
   El_mat([1:size(El_mat_order,1)],:)=El_mat_order;

else

   El_mat_order=vertcat(El_mat_order,flipud(El_2));
```

```
   El_mat([1:size(El_mat_order,1)],:)=El_mat_order;

end
   end

iEl=iEl+1;

end

end
```

TW_FEA_f_genK_mat (K ve M Matrisi Düzenleyici)

```
function [K]=TW_FEA_f_genK_mat(nm,K11,K12,K13,K14,...
                               K22,K23,K24,...
                               K33,K34,...
                               K44)

K11=genKij_22(K11,nm);
K12=genKij_24(K12,nm);
K13=genKij_24(K13,nm);
K14=genKij_24(K14,nm);

K22=genKij_44(K22,nm);
K23=genKij_44(K23,nm);
K24=genKij_44(K24,nm);

K33=genKij_44(K33,nm);
K34=genKij_44(K34,nm);

K44=genKij_44(K44,nm);

% Built Up K Matrice
H1=horzcat(K11,K12,K13,K14);
H2=horzcat(K12',K22,K23,K24);
H3=horzcat(K13',K23,K33,K34);
H4=horzcat(K14',K24,K34,K44);
```

```
K=vertcat(H1,H2,H3,H4);

end

function [Kij_nm]=genKij_22(Kij,nm)
%Calcualte Kij_22

Kij_nm=zeros(nm+1);

for i=1:1:(nm);
Kij_nm([i 1+i],[i 1+i])=Kij_nm([i 1+i],[i 1+i])+Kij([1 2],[1 2]);
end

end

function [Kij_nm]=genKij_24(Kij,nm)
%Calcualte Kij_24
K=zeros(nm+1,nm+1);

K1=K;
T([1 2],[1 2])=Kij([1 2],[1 2]);
for i=1:1:(nm)
   K1([i 1+i],[i 1+i])=K1([i 1+i],[i 1+i])+T([1 2],[1 2]);
end

K2=K;
T([1 2],[1 2])=Kij([1 2],[3 4]);
for i=1:1:(nm)
   K2([i 1+i],[i 1+i])=K2([i 1+i],[i 1+i])+T([1 2],[1 2]);
end

Kij_nm=horzcat(K1,K2);

end


function [Kij_nm]=genKij_44(Kij,nm)
%Calcualte Kij_24
K=zeros(nm+1,nm+1);
```

```
K1=K;
T([1 2],[1 2])=Kij([1 2],[1 2]);
for i=1:1:(nm)
   K1([i 1+i],[i 1+i])=K1([i 1+i],[i 1+i])+T([1 2],[1 2]);
end

K2=K;
T([1 2],[1 2])=Kij([1 2],[3 4]);
for i=1:1:(nm)
   K2([i 1+i],[i 1+i])=K2([i 1+i],[i 1+i])+T([1 2],[1 2]);
end

K3=K;
T([1 2],[1 2])=Kij([3 4],[1 2]);
for i=1:1:(nm)
   K3([i 1+i],[i 1+i])=K3([i 1+i],[i 1+i])+T([1 2],[1 2]);
end

K4=K;
T([1 2],[1 2])=Kij([3 4],[3 4]);
for i=1:1:(nm)
   K4([i 1+i],[i 1+i])=K4([i 1+i],[i 1+i])+T([1 2],[1 2]);
end

Kij_nm=vertcat(horzcat(K1,K2),horzcat(K3,K4));

end
```

TW_FEA_f_plot (Grafik Düzenleyici)

```
function TW_FEA_f_plot(U,headline,scale,export_def)

% Load Data
load('TW_FEA_main_vars.mat','El_mat','W','L','Ls','nm')

U=U*scale;

% Seperate Displacements
```

```
w=U(1:1:1*(nm+1));
u=U(1*(nm+1)+1:1:2*(nm+1));
du=U(2*(nm+1)+1:1:3*(nm+1));
v=U(3*(nm+1)+1:1:4*(nm+1));
dv=U(4*(nm+1)+1:1:5*(nm+1));
f=U(5*(nm+1)+1:1:6*(nm+1));
df=U(6*(nm+1)+1:1:7*(nm+1));

%Find Derivatives
Li=0:Ls:L;


%w
w_p = polyfit(Li',w,4);
%w_pv=polyval(w_p,Li');

dw_p=polyder(w_p);
dw_pv=polyval(dw_p,Li');


%ddu
du_p = polyfit(Li',du,4);
%du_pv=polyval(du_p,Li');

ddu_p=polyder(du_p);
ddu_pv=polyval(ddu_p,Li');


%ddv
dv_p = polyfit(Li',dv,4);
%dv_pv=polyval(dv_p,Li');

ddv_p=polyder(dv_p);
ddv_pv=polyval(ddv_p,Li');


%ddf
df_p = polyfit(Li',df,4);
%df_pv=polyval(df_p,Li');
```

```
ddf_p=polyder(df_p);
ddf_pv=polyval(ddf_p,Li');


if strcmp(export_def,'on') == 1

assignin('base' ,'w', w);
assignin('base' ,'u', u);
assignin('base' ,'v', v);
assignin('base' ,'f', f);
% assignin('base' ,'f_pv', f_pv);
assignin('base' ,'du', du);
assignin('base' ,'dv', dv);
assignin('base' ,'df', df);
% assignin('base' ,'df_pv', df_pv);

end


%% Plot Pole Line

Li=0:Ls:L;

figure('Name',headline);
hold on

line(Li+w',u',v',...
   'LineWidth',2,'color','k',...
   'Marker','o','MarkerSize',2,...
   'MarkerEdgeColor','g',...
   'MarkerEdgeColor','g');
view(0, 0)

%% Plot 3D
Z_p=zeros(size(El_mat,1),nm+1);
X_p=zeros(size(El_mat,1),nm+1);
Y_p=zeros(size(El_mat,1),nm+1);
C=zeros(size(El_mat,1),nm+1);
```

```
for i=1:1:size(El_mat,1);
    Z_p(i,:)=w-du.*El_mat(i,3)/1000-dv.*El_mat(i,4)/1000-df*W(i,1)+Li';
    X_p(i,:)=u+f.*El_mat(i,4)/1000+El_mat(i,3)/1000;
    Y_p(i,:)=v-f.*El_mat(i,3)/1000+El_mat(i,4)/1000;

    C(i,:)=abs(w'+u'+v');
end

C_max=max(max(C));
C=abs(C./C_max);

axis_l=0.5;

surf(Z_p,X_p,Y_p,C);
line(Z_p(:,end),X_p(:,end),Y_p(:,end));

xlabel('(m)')
ylabel('(m)')
zlabel('(m)')
colormap jet

view(45,45)

axis equal

zlim([-axis_l axis_l])
ylim([-axis_l axis_l])
xlim([0 1.1*L])

hold off

end


%% Refine Derivatives

function [ref_out]=refine2zero(ref_in)
```

```
ref_out=(ref_in-ref_in(1))*ref_in(end)/(ref_in(end)-ref_in(1));

end
```

TW_FEA_f_vib (Serbest Titreşim Çözücüsü)

```
function
[Wmod_u,Wmod_v,Wmod_w,Wmod_f,Wn_u,Wn_v,Wn_w,Wn_f,Wn,Wmod]=TW_FEA_f_vib(Ks,Ms,
export_var)

% Load Data
load('TW_FEA_main_vars.mat','U_bc','sk','nm')

% Find Eigenvalues
[Vs,Ds]=eig(Ks,Ms);

% assignin('base','Vs',Vs);
% assignin('base','Ds',Ds);

ss=size(Ks,1);

% Resize With Zeros
Vm=zeros(ss,sk);
Dm=zeros(ss,sk);
k=0;
for i=1:sk
   if U_bc(i)==1
      k=k+1;
      Vm(:,i)=Vs(:,k);
      Dm(:,i)=Ds(:,k);
   end
end

V=zeros(sk,sk);
D=zeros(sk,sk);
k=0;
for i=1:sk
   if U_bc(i)==1
      k=k+1;
```

```
        V(i,:)=Vm(k,:);
        D(i,:)=Dm(k,:);
    end
end

V=real(V);
D=real(D);

% Normalize V
Vr=zeros(sk,sk);
for i=1:7
    if i==1 || i==2 || i==4 || i==6
        Vr((i-1)*(nm+1)+1:i*(nm+1),:)=V((i-1)*(nm+1)+1:i*(nm+1),:);
    end
end
for i=1:sk
        maxV=max(abs(Vr(:,i)));
    if maxV ~= 0
        V(:,i)=V(:,i)/maxV;
    end
end

% Find and Sort Natural Freq.
Dn(sk,2)=0;

for i=1:sk
    Dn(i,1)=i;
    Dn(i,2)=real(sqrt(D(i,i)));
end
Dn=sortrows(Dn,2);

k=0;
for i=1:sk
    if Dn(i,2)~=0
        k=k+1;
        Wn(k,:)=Dn(i,:);
    end
end
```

```
Wn=Wn';
Wn(3,:)=Wn(2,:)/(2*pi());

Wmod=V(:,Wn(1,:));



% Wn_mod Seperation and Reorder
Wmod_w=Wmod(1:1:1*(nm+1),:);
Wmod_u=Wmod(1*(nm+1)+1:1:2*(nm+1),:);
Wmod_v=Wmod(3*(nm+1)+1:1:4*(nm+1),:);
Wmod_f=Wmod(5*(nm+1)+1:1:6*(nm+1),:);

[~,col_w] = find(abs(Wmod_w)==1);
[~,col_u] = find(abs(Wmod_u)==1);
[~,col_v] = find(abs(Wmod_v)==1);
[~,col_f] = find(abs(Wmod_f)==1);

Wmod_w=Wmod(:,col_w);
Wmod_u=Wmod(:,col_u);
Wmod_v=Wmod(:,col_v);
Wmod_f=Wmod(:,col_f);

Wn_w=Wn(:,col_w);
Wn_u=Wn(:,col_u);
Wn_v=Wn(:,col_v);
Wn_f=Wn(:,col_f);

if strcmp(export_var,'on') == 1

   assignin('base','Wn',Wn)
   assignin('base','Wmod',Wmod);

   assignin('base','Wmod_w',Wmod_w);
   assignin('base','Wmod_u',Wmod_u);
   assignin('base','Wmod_v',Wmod_v);
   assignin('base','Wmod_f',Wmod_f);

   assignin('base','Wn_w',Wn_w);
```

```
    assignin('base','Wn_u',Wn_u);
    assignin('base','Wn_v',Wn_v);
    assignin('base','Wn_f',Wn_f);

end

end
```

TW_FEA_f_save2file (XLXS Dosya Kaydedicisi)

```
function TW_FEA_f_save2file(w,u,du,v,dv,f,df,Wn_w,Wn_u,Wn_v,Wn_f,work_num)

load('GUI_DATA.mat','FileRes','PathRes')
load('TW_FEA_main_vars.mat','Lyr_mat')

Ang='';
Thc='';

for i=1:size(Lyr_mat,2)
    Ang=strcat(Ang,'/',num2str(Lyr_mat(1,i)));
    Thc=strcat(Thc,'/',num2str(Lyr_mat(2,i)*1000));
end

Wmax=max(w);
Wmin=min(w);

Umax=max(u);
Umin=min(u);

dUmax=max(du);
dUmin=min(du);

Vmax=max(v);
Vmin=min(v);

dVmax=max(dv);
dVmin=min(dv);

Fmax=max(f);
```

```
Fmin=min(f);

dFmax=max(df);
dFmin=min(df);

if isempty(Wn_f) == 1
   Wn_f(3,1)=0;
end


% Write Headlines
Cell_Headlines={'Lay-up','Thickness','','w-max','w-min','u-max','u-min','du-max','du-min','v-max','v-
min','dv-max','dv-min','f-max','f-min','df-max','df-min','','Wn_w','Wn_v','Wn_u','Wn_f'};
sheet = 1;
xlRange = 'B2';
xlswrite(strcat(PathRes,FileRes),Cell_Headlines,sheet,xlRange)

% Write Results
Results_def={Ang,Thc,'',Wmax,Wmin,Umax,Umin,dUmax,dUmin,Vmax,Vmin,dVmax,dVmin,Fmax,F
min,dFmax,dFmin,'',Wn_w(3,1),Wn_v(3,1),Wn_u(3,1),Wn_f(3,1)};
sheet = 1;
xlRange = strcat('B',num2str(work_num+2));
xlswrite(strcat(PathRes,FileRes),Results_def,sheet,xlRange)

end
```

POLYSUM (Polinom Toplayıcı)

```
function
[psum]=polysum(poly1,sign1,poly2,sign2,poly3,sign3,poly4,sign4,poly5,sign5,poly6,sign6,poly7)

   switch nargin

   case 3
      psum = polysum_int(poly1,sign1,poly2);

   case 5
      psum = polysum_int(polysum_int(poly1,sign1,poly2),sign2,poly3);
```

```
    case 7
        psum = polysum_int(polysum_int(polysum_int(poly1,sign1,poly2),sign2,poly3),sign3,poly4);

    case 9
        psum =
polysum_int(polysum_int(polysum_int(polysum_int(poly1,sign1,poly2),sign2,poly3),sign3,poly4),sign4,
poly5);

    case 11
        psum =
polysum_int(polysum_int(polysum_int(polysum_int(polysum_int(poly1,sign1,poly2),sign2,poly3),sign3,
poly4),sign4,poly5),sign5,poly6);

    case 13
        psum =
polysum_int(polysum_int(polysum_int(polysum_int(polysum_int(polysum_int(poly1,sign1,poly2),sign2,
poly3),sign3,poly4),sign4,poly5),sign5,poly6),sign6,poly7);

    otherwise

        error('Wrong input, Check your input order')

    end

function [psum]=polysum_int(poly1,sign,poly2)

    if isa(poly1,'char')==1 || isa(poly2,'char')==1
        error('Wrong input, Check your input order')
    end

    if sign=='+'
        psum=[zeros(1, size(poly2,2)-size(poly1,2)) poly1]+[zeros(1, size(poly1,2)-size(poly2,2)) poly2];
    elseif sign=='-'
        psum=[zeros(1, size(poly2,2)-size(poly1,2)) poly1]-[zeros(1, size(poly1,2)-size(poly2,2)) poly2];
    else
        error('Poly Sum/Min Error... Check Sign')
    end
end
end
```

ATANG (Ters Tanjant Hesaplayıcı)

```
function [atang]=atang(P1,P2)
atang=atand(abs((P1(1,2)-P2(1,2))/(P1(1,1)-P2(1,1))));
if P1(1,1)<=P2(1,1) && P1(1,2)<=P2(1,2)
   atang=atang;
elseif P1(1,1)<P2(1,1) && P1(1,2)>P2(1,2)
   atang=360-atang;
elseif P1(1,1)>=P2(1,1) && P1(1,2)>=P2(1,2)
   atang=180+atang;
elseif P1(1,1)>P2(1,1) && P1(1,2)<P2(1,2)
   atang=180-atang;
end
atang=mod(atang,360);
end
```

DistP1P2 (İki Nokta Arası Uzaklık Hesaplayıcı)

```
function [Dist]=distP1P2(P1,P2)
   Dist=sqrt((P1(1,1)-P2(1,1))^2+(P1(1,2)-P2(1,2))^2);
end
```

f_VarGrCode (DXF Eleman Seçici) [28]

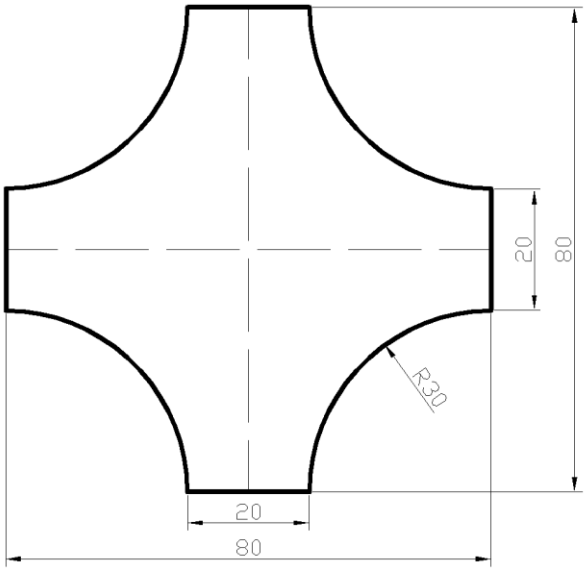
```
function c_Val = f_ValGrCode(grCode,m_GrCode,c_ValAsoc)
   c_Val = c_ValAsoc(m_GrCode==grCode);
end
```

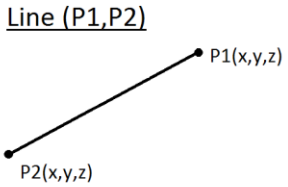
f_XData (DXF Eleman Okuyucu) [28]

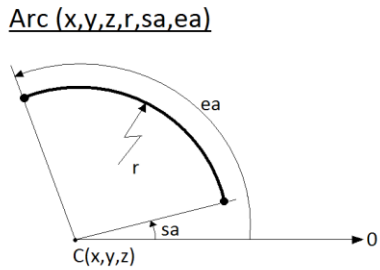
```
function c_XData = f_XData(grCode,XDatNom,m_GrCode,c_ValAsoc)
   m_PosXData = find(m_GrCode==1001);
   if ~isempty(m_PosXData)
      indInXData = m_PosXData(strmatch(upper(XDatNom),c_ValAsoc(m_PosXData),'exact'));
      m_indFinXData = find(m_GrCode(indInXData+2:end)==1002)+indInXData+1;
      m_IndXData = indInXData+2:m_indFinXData(1)-1;
      c_XData = f_ValGrCode(grCode,m_GrCode(m_IndXData),c_ValAsoc(m_IndXData));
   else
      c_XData = {[ ]};
   end
end
```

EK-C Örnek Kesit ve DXF Dosyası

Aşağıda bir kesite ait teknik resim ve bu teknik resmi tanımlayan DXF dosyası örnek olarak verilmiştir. DXF dosyasının tamamı burada verilmemiştir, sadece geometrik tanımlamaların geçtiği ‘entities’ bölümü verilmiştir.







*Parantez içindeki açıklamalar sonradan eklenmiştir.

ENTITIES	0.0	0.0000000000000426
0	11 (P2-x)*	20 (center-y)*
LINE(P1,P2)*	30.00000000000004	0.0000000000001254
5	21 (P2-y)*	30 (center-z)*
29A	0.0000000000001217	0.0
102	31 (P2-z)*	40
{ACAD_REACTORS	0.0	30.0 (Radius-r)*
330	0	100
2D8	ARC (x,y,z,r,sa,ea)*	AcDbArc
330	5	50 (start angle-sa)*
318	2A2	0.0
102	102	51 (end angle-ea)*
}	{ACAD_REACTORS	90.00000000000005
330	330	0
1F	300	LINE
100	102	5
AcDbEntity	}	2A3
8	330	330
gmg	1F	1F
100	100	100
AcDbLine	AcDbEntity	AcDbEntity
10 (P1-x)*	8	8
49.99999999999997	gmg	gmg
20 (P1-y)*	100	100
0.0000000000001192	AcDbCircle	AcDbLine
30 (P1-z)*	10 (center-x)*	10

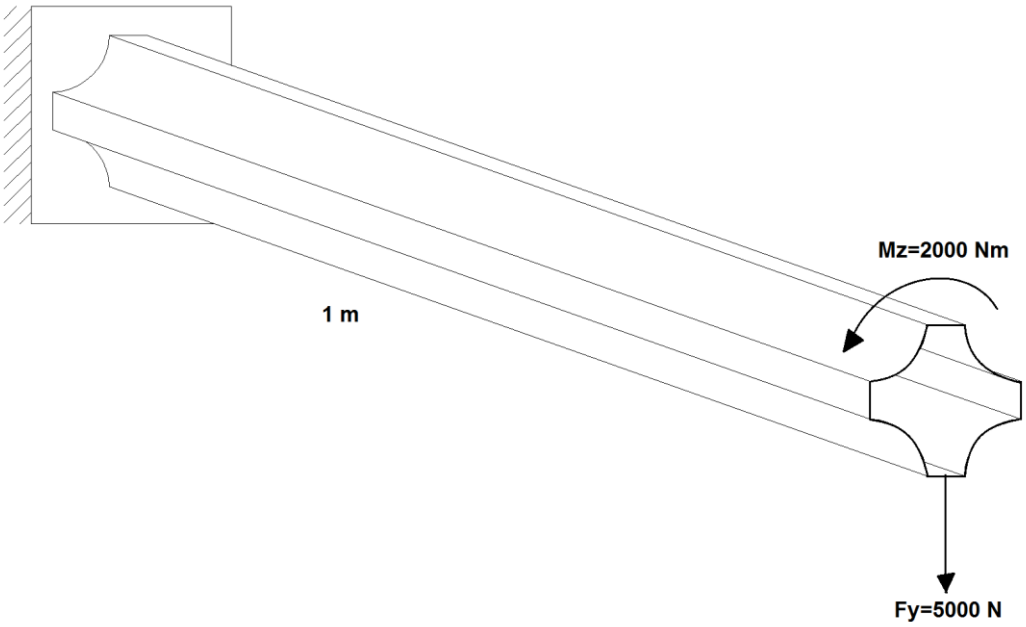
-0.0000000000048317
 20
30.00000000000012
 30
0.0
 11
0.0000000000001027
 21
50.00000000000001
 31
0.0
  0
ARC
  5
2A4
330
1F
100
AcDbEntity
  8
gmg
100
AcDbCircle
 10
0.0000000000000426
 20
80.00000000000001
 30
0.0
 40
30.0
100
AcDbArc
 50
269.9999999999999
 51
0.0
  0
LINE
  5
2A5
102
{ACAD_REACTORS
330
2D8
102
}
330
1F
100
AcDbEntity
  8
gmg
100
AcDbLine
 10
30.00000000000004
 20
80.00000000000101
 30
0.0
 11
50.00000000000003
 21
80.0
 31
0.0
  0
ARC
  5
2A6
102
{ACAD_REACTORS
330
2D8
102
}
330
1F
100
AcDbEntity
  8
gmg
100
AcDbCircle
 10
80.00000000000003
 20
79.99999999999999
 30
0.0
 40
30.0
100
AcDbArc
 50
179.9999999999999
 51
270.0
  0
LINE
  5
2A7
102
{ACAD_REACTORS
330
2C0
102
}
330
1F
100
AcDbEntity
  8
gmg
100
AcDbLine
 10
80.00000000000014
 20
49.99999999999999
 30
0.0
 11
79.99999999999987
 21
30.00000000000002
 31
0.0
  0
ARC
  5
2A8
102
{ACAD_REACTORS
330
2C0
330
2D8
330
2E8
330
300
102
}
330
1F
100
AcDbEntity
  8
gmg
100
AcDbCircle
 10
80.00000000000001
 20
0.0000000000000142
 30
0.0
 40
30.0
100
AcDbArc
 50
89.99999999999998
 51
180.0
  0
END

Malzeme Özellikleri ve Yükleme Koşulları:

Malzeme Özellikleri:

Lay-up	h	E_{11} (GPa)	E_{22} (GPa)	G_{12} (GPa)	v_{12}
[0/45]	2mm	142	9,80	6,00	0,42

Yükleme Koşulları:

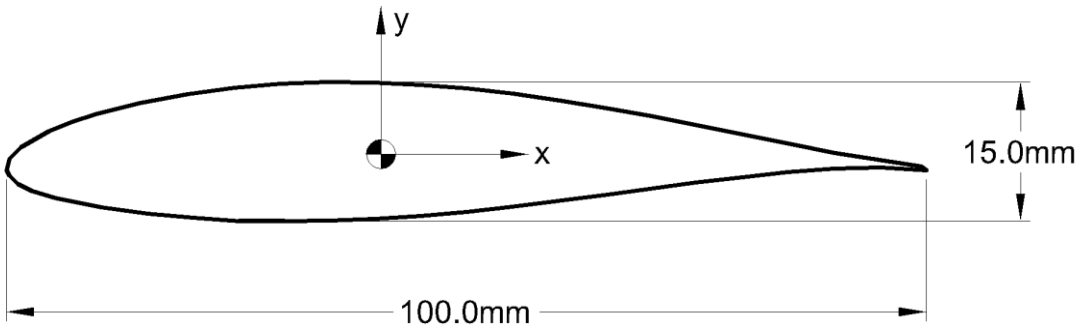


Sonuçlar:

w (m)	u (m)	v (m)	f (rad)
7,82e-4	1,9e-18	-0,0653	-0,5806

EK-D Kesit Boyutları

1) NACA-63-415 Kesiti



NACA-63-415 Koordinatları (1/100)

	1	2	3	4	5	6	7	8	9	10	11
X	0,0000	0,0030	0,0053	0,0099	0,0220	0,0466	0,0715	0,0965	0,1467	0,1971	0,2475
Y	0,0000	0,0129	0,0159	0,0207	0,0296	0,0426	0,0526	0,0608	0,0735	0,0828	0,0894

	12	13	14	15	16	17	18	19	20	21	22
X	0,2980	0,3485	0,3991	0,4496	0,5000	0,5504	0,6007	0,6509	0,7011	0,7511	0,8010
Y	0,0936	0,0956	0,0953	0,0929	0,0887	0,0830	0,0760	0,0678	0,0588	0,0491	0,0390

	23	24	25	26	27	28	29	30	31	32	33
X	0,8509	0,9006	0,9503	1,0000	0,0000	0,0070	0,0098	0,0151	0,0280	0,0534	0,0785
Y	0,0289	0,0188	0,0093	0,0000	0,0000	-0,0109	-0,0131	-0,0165	-0,0222	-0,0300	-0,0357

	34	35	36	37	38	39	40	41	42	43	44
X	0,1035	0,1533	0,2030	0,2525	0,3020	0,3515	0,4010	0,4505	0,5000	0,5496	0,5993
Y	-0,0401	-0,0466	-0,0510	-0,0536	-0,0547	-0,0544	-0,0524	-0,0491	-0,0446	-0,0392	-0,0331

	45	46	47	48	49	50	51	52
X	0,6491	0,6989	0,7489	0,7999	0,8492	0,8994	0,9497	1,0000
Y	-0,0266	-0,0199	-0,0133	-0,0072	-0,0019	0,0018	0,0033	0,0000

2) Kompozit Bisiklet Çerçevesi Alt Kirişi

