

PLAKLARIN DARBE YÜKÜ ALTINDA ANALİZİ

Uğur KOÇ

**YÜKSEK LİSANS TEZİ
İNŞAAT MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MAYIS 2008

ANKARA

PLAKLARIN DARBE YÜKÜ ALTINDA ANALİZİ

Uğur KOÇ

YÜKSEK LİSANS TEZİ
İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

MAYIS 2008

ANKARA

Uğur KOÇ tarafından hazırlanan “PLAKLARIN DARBE YÜKÜ ALTINDA ANALİZİ” adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Tekin GÜLTOP

Tez Danışmanı, İnşaat Müh. Anabilim Dalı

.....

Bu çalışma, jürimiz tarafından oy birliği ile İnşaat Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Mehmet Emin TUNA

Mimarlık Anabilim Dalı, G.Ü.

.....

Prof. Dr. Tekin GÜLTOP

İnşaat Mühendisliği Anabilim Dalı, G.Ü.

.....

Doç. Dr. Kurtuluş SOYLUK

İnşaat Mühendisliği Anabilim Dalı, G.Ü.

.....

Tarih : 16/05/2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nermin ERTAN

Fen Bilimleri Enstitüsü Müdür

.....

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Uğur KOÇ

PLAKLARIN DARBE YÜKLERİ ALTINDA ANALİZİ**(Yüksek Lisans Tezi)****Uğur KOÇ****GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ****Mayıs 2008****ÖZET**

Plaklar genel olarak; bir boyutu diğer iki boyutuna göre daha küçük olan elemanlar olarak tanımlanırlar. Bu çalışmada tekil yük, üniform yayılı yük ve darbe yükleri altında plak deformasyonları sonlu farklar yöntemi ile hesaplanmıştır. Plakların tekil yük ve yayılı yük altında analizlerinde ikinci derece doğruluklu merkezi sonlu farklar metodu kullanılmış ve bu yöntemle elde edilen denklem sistemleri Matlab programının yardımıyla hesaplanmıştır. Geliştirilen programla deplasman değerleri hesaplanmıştır. Bu sonuçlar, analitik sonuçlarla kıyaslanmış ve hatanın çok düşük olduğu tespit edilmiştir. Darbe yükleri altında analiz için programda farklı bir kısım ayrılmıştır. Bu kısımda plaklar kirişlerden oluşan bir ızgara sistemi gibi kabul edilerek çözüm yapılmış ve deplasman değerleri hesaplanmıştır. Bu çalışma ile guide tabanı kullanılarak kullanıcıya verilerin girişi ve sonuçların izlenmesi için kolaylık sağlanmış, algoritması karışık olan ifadelerin çözümünün kısa sürede ve düşük hata oranıyla hesaplanması hedeflenmiştir.

Bilim Kodu : 911.1.144
Anahtar Kelimeler : Plak, sonlu farklar, darbe yükü, deformasyon
Sayfa Adedi : 80
Tez Yöneticisi : Prof. Dr. Tekin GÜLTOP

THE ANALYSIS OF PLATES UNDER IMPACT LOADING**(M.Sc. Thesis)****Uğur KOÇ****GAZİ ÜNİVERSİTY****INSTITUTE OF SCIENCE AND TECHNOLOGY****May 2008****ABSTRACT**

Plates generally described as the elements whose one dimension is smaller than the other two dimensions. In this work, the deformations of plates under single load, uniform distributed load and impact load, are calculated with finite difference method. In this analysis, second degree central finite difference method is used and the equations which are gained by this method are calculated with the help of Matlab program. By the improved program, the deformations of plates are estimated. These results are compared with the analytical results and determined that the errors are negligible. For the analysis under impact loading, there is another part in the program. In this part, plates are accepted as grids composed of beams and the deformation values are calculated. With this work, it is aimed to make the entry of inputs and the impression of results easy for the users. And Also it is aimed to calculate the complex algorithms with a negligible errors.

Science Code	: 911.1.144
Key Words	: Plate, finite difference, impact loading, deformation
Page Number	: 80
Adviser	: Prof. Dr. Tekin GÜLTOP

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Prof. Dr. Tekin GÜLTOP'a, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan babam Nevzat KOÇ, annem Sevim KOÇ, ablam Pınar ÇALAPALA, eniştem Murat ÇALAPALA, yeğenim Kemal Onur ÇALAPALA ve manevi desteklerinin yanında bana her konuda yardımcı olan ablam Dilek KOÇ SAN ve eniştem Bekir Taner SAN'a teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xii
1.GİRİŞ.....	1
2. PLAKLAR TEORİSİ TEMEL DENKLEMLERİ.....	7
2.1 Plak Genel Denklemleri.....	7
2.2 Klasik Yöntemler.....	11
2.2.1 Fourier Serileri.....	11
2.3 Numerik ve Yaklaşık Yöntemler.....	15
2.3.1 Sonlu Farklar Yöntemi.....	15
3. SONLU FARKLAR METODUYLA YAPILAN ÇÖZÜMLER.....	18
3.1 Sonlu Fark Metodunun Programda Uygulanışı.....	18
3.2 Örnekler.....	21
4. DARBE YÜKLERİ.....	29
4.1 Darbe Yüklemesinin Tanımı Özellikleri ve Programda Uygulanışı.....	29
4.2 Örnekler.....	33
5. SONUÇLAR.....	41
KAYNAKLAR.....	42

	Sayfa
EKLER.....	44
EK-1 Program Kodu.....	45
ÖZGEÇMİŞ.....	80

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Bazı mesnet durumlarına göre matematiksel ifadeler.....	9
Çizelge 2.2. Bazı önemli yükleme tipleri.....	12
Çizelge 3.1. Sonlu farklar metodu matris eşitliği.....	18
Çizelge 3.2. Plak deplasman değerleri (cm).....	22
Çizelge 3.3. Plak gerilmeleri (kgf/cm^2).....	23
Çizelge 3.4. Plak gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	23
Çizelge 3.5. Plak deplasman değerleri (cm).....	25
Çizelge 3.6. Plak gerilmeleri (kgf/cm^2).....	25
Çizelge 3.7. Plak gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	26
Çizelge 3.8. Plak deplasman değerleri (cm).....	27
Çizelge 3.9. Plak gerilmeleri (kgf/cm^2).....	27
Çizelge 3.10. Plak gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	27
Çizelge 4.1. Darbe yüklemesi altında deplasman değerleri(cm).....	35
Çizelge 4.2. Darbe yüklemesi altında gerilme değerleri(kgf/cm^2).....	35
Çizelge 4.3. Darbe yüklemesi altında gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	36
Çizelge 4.4. Darbe yüklemesi altında deplasman değerleri (cm).....	38
Çizelge 4.5. Darbe yüklemesi altında gerilme değerleri(kgf/cm^2).....	39
Çizelge 4.6. Darbe yüklemesi altında gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	39
Çizelge 4.7. Tekil yük altında gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	40
Çizelge 4.8. Darbe yükü altında gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$).....	40

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Plak ve dış kuvvetin oluşturduğu gerilme ve momentler.....	6
Şekil 2.2. M_x ve M_y momentleri etkisi altında bulunan plak.....	7
Şekil 2.3. İki kenarı basit mesnetli dikdörtgen plak.....	13
Şekil 3.1. Program analiz ekranı.....	17
Şekil 3.2. Sonlu farklar ağ sistemindeki nokta katsayıları.....	18
Şekil 3.3. Plak deplasmanları (cm).....	19
Şekil 3.4. Plak deplasmanları(cm).....	21
Şekil 3.5. Plak Gerilmeleri (kgf/cm^2).....	22
Şekil 3.6. Plak gerilme enerjisi($\text{kgf/cm}^2/\text{cm}^3$).....	22
Şekil 3.7. Plak deplasmanları (cm).....	24
Şekil 3.8. Plak gerilmeleri (kgf/cm^2).....	24
Şekil 3.9. Plak gerilme enerjisi ($\text{kgf/cm}^2/\text{cm}^3$).....	25
Şekil 3.10. Plak deplasmanları (cm).....	26
Şekil 4.1. Plak ve aynı ölçülerdeki kiriş çerçeve üzerindeki statik yük.....	29
Şekil 4.2. Programda darbe yükü için oluşturulan arayüz.....	32
Şekil 4.3. Plak deplasmanları (cm).....	32
Şekil 4.4. Darbe yüklemesinde girilen değerler.....	33
Şekil 4.5. Darbe yüklemesi altında deplasman grafiği(cm).....	34
Şekil 4.6. Darbe yüklemesi altında gerilme grafiği (kgf/cm^2).....	34
Şekil 4.7. Darbe yüklemesi altında gerilme enerjisi grafiği ($\text{kgf/cm}^2/\text{cm}^3$).....	35
Şekil 4.8. Darbe yüklemesinde girilen değerler.....	36
Şekil 4.9. Darbe yüklemesi altında deplasman grafiği (cm).....	37

Şekil	Sayfa
Şekil 4.10. Darbe yüklemesi altında gerilme grafiği (kgf/cm ²).....	37
Şekil 4.11. Darbe yüklemesi altında gerilme enerjisi (kgf/cm ² /cm ³).....	38

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
M_x	Mesnetlerde oluşan X yönündeki momentler
M_y	Mesnetlerde oluşan Y yönündeki momentler
K_x	Plakta oluşan X yönündeki eğrilik
K_y	Plakta oluşan Y yönündeki eğrilik
w	Plağın düşey doğrultusundaki deplasman
ν	Poisson oranı
D	Plak eğilme rijitliği
t	Plak kalınlığı
E	Young modülü (Elastisite katsayısı)
Q	Plağın üzerindeki yük
P	Plağın üzerindeki darbe yükü
m	Plağa çarpan cismin kütlesi
g	Yerçekim ivmesi
h	Plağa çarpan cismin düşmeden önceki yüksekliği
U_m	Çarpan cismin enerjisi
U_e	Plağın gerilme enerjisi
v_0	Çarpan cismin ilk hızı
L	Kiriş uzunluğu
$[k]$	Rijitlik matrisi

1.GİRİŞ

Plaklar genel olarak, bir boyutu diğer iki boyutuna göre daha küçük olan düzlemsel eleman olarak tanımlanır. İnşaat Mühendisliğinde birçok alanda kullanıldığı için önemli bir yapı elemanıdır. Yapı döşemeleri, barajların su kapakları, köprüler, çelik plaklar bunlardan sadece birkaçıdır. Geometrik olarak, düz veya eğik çizgilerle sınırlandırılırlar. Statik açıdan plaklar, serbest, basit mesnetli, ankastre ve elastik sınırlandırılmalı ve elastik destekleri içeren sabit sınır koşullarına sahiptir. Statik veya dinamik yükler plaklar tarafından çoğunlukla plak yüzeyine dik olarak etki eder [1]. Plaklar gösterdikleri davranış ve yapısal işlevlerine göre dörde ayrılırlar;

Rijit Plaklar : Plak denilince akla ilk gelen plak çeşididir. Eğilme, burulma momentlerini ve kesme kuvvetlerini alır.

Membran Plaklar : Plak kalınlığı çok ince olan plaklardır. Orta yüzeyde eksenel kuvvet ve yüzey düzleminde kesme kuvvetlerini almasına karşın eğilme rijitliği çok küçüktür.

Eğilebilir Plaklar : Rijit ve membran plakların özelliklerini taşıyan plaklardır.

Kalın Plaklar : Kalınlığı diğer plak çeşitlerine göre büyük olan, problemlerde 3. boyutunda hesaba katılması gereken plaklardır.

Bu çalışmada, eğilme etkisi gösteren Kirchhoff (ince plaklar) dikkate alınacaktır. Plak teorisinde plaklar için genel bazı kabuller vardır. Plakların analizinde küçük deformasyonlar teorisine göre kullanılan kabuller şunlardır;

- Kalınlık (h), diğer boyutların yanında çok küçüktür. $h \ll L$
- Deplasman (w), kalınlıktan(h) daha küçüktür. $w \ll h$
- Plak geometrisi başlangıçta düz durumdadır ve tarafsız düzlemine dik yüzeyler, deformasyondan sonra da dik kalırlar.
- Tarafsız düzleme dik gerilmeler ve kesme deformasyonları ihmal edilmektedir.

Plakların analizinde klasik yöntemler, sayısal ve yaklaşık yöntemler olmak üzere iki çeşit yöntem kullanılmaktadır. Bu çalışmada plakların tekil yük ve yayılı yük altında analizleri için ikinci derece doğruluklu merkezi sonlu farklar yöntemi kullanılarak çözülmüştür. Darbe yükleri altında analiz için ise plaklar, kirişlerden oluşan eşdeğer ızgara sistemi olarak modellenerek hesap yapılacaktır. Sonlu farklar yöntemi, çeşitli diferansiyel denklemlerin kapalı çözümleri elde edilemediğinde, kullanılan sayısal ve yaklaşık yöntemlerdendir [2]. İnce plakların dördüncü dereceden kısmi türevli, diferansiyel denklemi için sonlu farklar yöntemi uygulandığında; seçilen ağın düğüm noktalarında diferansiyel büyüklüklere göre denklemdaki türevler bulunmakta ve problem için sonlu fark operatörü elde edilmektedir [3]. Problemlerin çözümü bu yolla yapılmaktadır. Bu çalışmada, analizler için program yapılmış ve programlama dili olarak Matlab kullanılmıştır. Programlama dili olarak Matlab seçilmesinin sebepleri ise;

- Plak analizinde kullanılan yöntemde ayrılan parça ile orantılı olarak büyük matrislerin çıkması ve bu algoritmanın ancak bu program ile çözülebilecek olması,
- Yüksek seviyeli bir teknik programlama dili olması,
- Veri analizi ve sayısal hesaplamalar için uygun bir program olması,
- Grafik arayüzler tasarlamak için araçlar bulunması,
- Verilerin GUIDE tabanında görselleştirilebilmesi ve rahatlıkla izlenebilmesidir.

Böylece kullanıcı için verilerin kolaylıkla girişine imkan tanınmış ve sonlu farklar yöntemi ile oluşturulması gereken manuel olarak hesaplanamayacak büyüklükte ve karışıklıkta olan matris sistemlerinin çözümü sağlanmıştır.

Elastik cisme ani bir kuvvet uygulandığı zaman, cisim boyunca belirli bir hızla hareket eden bir gerilme dalgası oluşur. Buna darbe yükü denir. Darbe, yapısal ve malzeme tepkisini ve hasar bilgilerini içeren karışık bir konudur. Bununla beraber darbenin gerilme analizleri, bütün problemi anlamının temelidir. Bu çalışmada darbe yükleri altında analizde plaklar eşdeğer kiriş sistemlerinden oluşan ızgara şeklinde modellenmiştir. Deformasyonlar, bu model dikkate alınarak hesaplanmıştır.

Literatür Araştırması

Plaklar ve darbe yüklerine ilişkin olarak genelde teorik çalışmalar yapılmış, deneysel olarak yapılan çalışmalarda plakların darbe yükleri altında analizleri laboratuvar ortamında deneysel olarak farklı yönleriyle incelenmiştir. Teorik çalışmalarda plakların genel tanımları, denklemleri, gerilme, moment ve diğer kuvvetler arasındaki ilişkiler incelenmiştir. Literatürde bulunan teorik ve pratik çalışmalardan bazıları aşağıda kısaca özetlenmiştir.

Ergün ve Kumbasar (2003) tarafından yapılan çalışmada ince plaklar için sonlu fark ifadeleri incelenmiş ve klasik sonlu fark ifadelerinin çıkartılışından tamamen bağımsız olarak geliştirilmiş bir sonlu fark yöntemi ortaya konulmuştur. Öncelikle sonlu farklar yöntemi incelenmiş, sonlu fark denkleminin elde edilmesi gösterilmiştir. Daha sonra tekil yük, çizgisel yayılı yük ve bölgesel yayılı yük altında deplasman ifadeleri çıkartılmıştır. Sınır koşulları üzerinde de inceleme yapıldıktan sonra örneklerle bilgiler pekiştirilmiştir. Çalışma sonucunda geliştirilen sonlu fark yönteminin diğer sonlu fark yöntemlerinden daha yüksek hassasiyet ve performans gösterdiği formülasyonunun kolay, işlevsel ve çok yönlü olduğu belirtilmiştir. Yine geliştirilen yöntemle, serbest uçlu sınır koşullarında sonlu fark operatörünün hesaplanarak deplasman ifadelerinin bulunabileceği ve ankastre mesnet sınır koşulunda oluşan hatanın Betti teoremindeki mesnet tepkileri işinin hesaplanamamasından kaynaklandığı belirtilmiştir. Çalışmanın sonuçlarından bir diğeri ise ağ sisteminin sıklaştırılması ile hata payının birbiriyle çok ilgili olmadığının gösterilmesidir. Böylece yöntemin elle çözüm için uygun olduğu, klasik sonlu farklar yöntemindeki gibi hata payının düşük olması için çok sayıda bilinmeyen oluşması ve bilgisayarla çözümün zorunlu olmadığı gösterilmiştir. Bunun için bulunan yöntemin plak teorisinde yardımcı yöntem olarak kullanılabileceği noktasına dikkat çekilmiştir [3].

Smetankina, Shupikov, Sotrikhin ve Yareschenko (2005) tarafından gerçekleştirilen çalışmada eliptik bir plağın darbe yüklemesine verdiği dinamik tepki, teorik ve deneysel olarak incelenmiştir. Darbe problem durumu açıklanmış ve formülasyonları

elde edildikten sonra çözüm yöntemi anlatılmıştır. Deneysel kurulum ve metotlar açıklandıktan sonra sayısal sonuçlarla grafikler oluşturulmuş ve böylece çalışma sonlandırılmıştır. Çalışma sonucunda eliptik bir plağın darbe yükü altında dinamik tepkisi için analitik bir yöntem geliştirilmiş ve darbe probleminin çözümü trigonometrik serilerle elde edilmiştir. Bununla birlikte plağın dinamik davranışlarının incelenme tekniği önerilmiş ve eliptik plağın merkez ve odakta bulunan darbe yükleri altında gerilme hesapları örnekle gösterilmiştir [4].

Franz (2001) tarafından gerçekleştirilen çalışmada darbe yüklerine maruz ortotropik kompozit plakların mekanik etkisinin fotoelastik incelemesi yapılmıştır. Deneysel çalışmada pünomatik darbe test cihazı ve yüksek ısı çözünürlüğü için ultrakısa-dalga ışık kaynakları kullanılmıştır. Kompozit malzemelerin ve darbe yükünün karışıklıkları sebebiyle, gerilme uygunluğu dizaynı için plağın darbe yükünden hemen sonraki mekanik davranışı bulunmaya çalışılmıştır. Deneyin sonucunda darbe yükünden oluşan dalga yayılımları resimlerle ifade edilmiş, Young modülü ile dalga boylarının ilgisi belirtilmiştir. Bununla birlikte plak üzerinde oluşan gerilmelerin azaltılışının farklı bir başlık dizaynı veya fiber malzeme kullanılarak mümkün olabileceğinden bahsedilmiştir [5].

Olsson, Donadon ve Falzon (2005) tarafından gerçekleştirilen çalışmada darbe yükü farklı bir açıdan ele alınarak plaklarda küçük kütle ve yüksek hız darbe yüklemesinde tabakalaştırma konusu incelenmiştir. Çok kısa, kısa ve uzun darbe yüklemeleri süresince plaktaki tabakalaşma etkilerin incelenmesi yapılmış bu incelemelerin nerelerde nasıl kullanılabileceği açıklanmıştır. Sonlu elemanlar çözümü ve bunun deneysel karşılaştırması yapılmıştır. Çalışmanın sonucunda teorik çalışmanın deneysel sonuçlarla karşılaştırılması grafik üzerinden yapılarak hatanın düşük olduğu tespit edilmiş ve düşük kütle/yüksek hız darbe yüklemesi altında bu tabakalaşma teorisinin kullanılabileceği belirtilmiştir [6].

Yoon, Heo, Song ve Jung (2004) tarafından yapılan çalışmada yanal darbe yükü altında çoklu-nokta sınırlamalı denklemleri kullanarak ızgara sisteminin dinamik darbe analizleri yapılmıştır. Çalışmanın amacı ızgara yapısının dinamik çökme

davranışını önceden kestirebilen bir metodoloji önermektir. Bunu yapabilmek için 5x5 hücreli kısmi ızgara numunesinin test bilgileri ve Abaqus programının sonlu eleman çözümü ile bulunan burkulma davranışı kullanılmıştır. Sonlu eleman çözümünde gerçek test sonuçlarına ulaşmak için uygun sınır koşulları ve darbe yükleme durumları kullanılmıştır. Geometrik bilgi, malzeme özellikleri, darbenin ivme, hız ve kuvveti, darbe etkisinden sonraki gerilme ve basınç değerleri sayısal olarak elde edilmiş ve bu sonuçların sonlu eleman çözümü ile karşılaştırılması yapılmıştır. Söz konusu çalışma neticesinde, önerilen çoklu nokta sınırlamalı modelin sonlu eleman çözümüne göre daha gerçekçi sonuçlar verdiği, deneysel sonuçlarla daha çok uyum gösterdiği belirtilmiştir. Yine bu modelin rijitliğinin gerçek yapıya daha yakın olduğu gözlenmiş, grafiklerle gösterilmiştir [7].

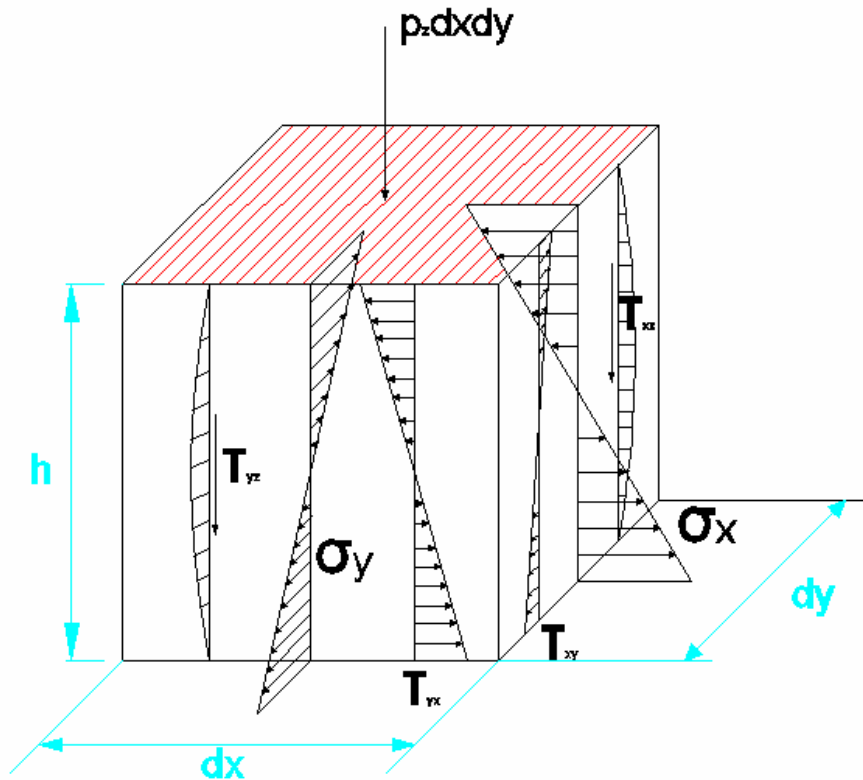
Abbas, Gupta ve Alam (2004) tarafından gerçekleştirilen çalışmada kiriş ve plakların darbe yükü altında nonlinear tepkileri incelenmiştir. Deneyler laboratuvar ortamında düşen çekiç yüklemesi ile dairesel betonarme plaklar üzerinde yapılmıştır. Plakın darbe yükü altında çatlama gerilmeleri tespit edilmiş ve sonuçlar hem fotoğraflanmış hem de grafiksel olarak ifade edilmiştir. Deneyin sonucunda çatlama gerilmesi doğru bir şekilde incelenmiş olduğu ve çıkan sonuçların literatürdeki araştırmalarla benzer olduğu belirtilmektedir [8].

Aslan, Karakuzu ve Okutan (2003) tarafından gerçekleştirilen çalışmada lamine kompozit plakların düşük hızlı darbe yüklemesi altında tepkileri incelenmiştir. Çalışılan lamineler, fiber ve epoksi reçinelerini içeren E-camlardan üretilmiştir. Sonlu eleman kodlamasının kullanıldığı 3DIMPACT programı ile yapılan sayısal simülasyonlar (benzeşimler) ile deneysel sonuçlar kıyaslanmıştır. Darbe yüklemesinde düşen cismin plaktan ayrıldığı sırada belirli bir zaman içerisinde darbe kuvvetinin sıfıra indiği gözlenmiştir. Çalışmanın sonucunda düşük hızlı darbe yüklemesine maruz kalan cismin mekanik davranışının düzlem ölçülerine bağlı olduğu, en büyük darbe kuvvetinin orta büyüklükteki plakta gözlemlendiği belirtilmiştir [9].

2. PLAKLAR TEORİSİ TEMEL DENKLEMLERİ

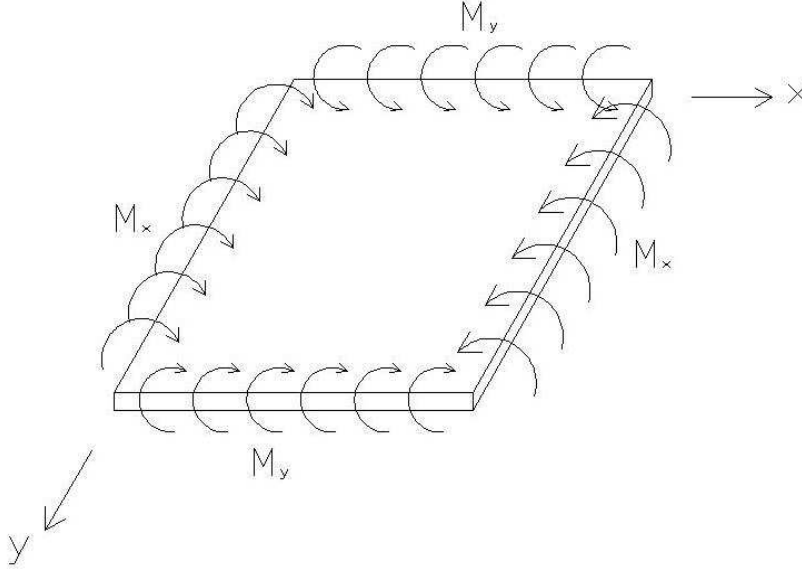
2.1. Plak Genel Denklemleri

Plaklar kesme kuvvetleri, eğilme momentleri ve burulma momentlerini taşırlar. Plakların analizlerinde genelde küçük deformasyonlar teorisine göre kabuller yapılır. Örnek bir plak ve dış kuvvetin oluşturduğu gerilme ve momentler Şekil 2.1’de gösterilmiştir. Bu çalışmada kesme kuvvetlerinin ihmal edildiği ve eğilme davranışının etkin olduğu ince plak teorisi(Kirchhoff plak teorisi) kullanılacaktır.



Şekil 2.1. Plak ve dış kuvvetin oluşturduğu gerilme ve momentler

Dikdörtgen plaklar için genel bağıntıların bulunması için yapılması gereken Şekil 2.2’de gösterilen M_x ve M_y momentleri altındaki bir plak için eğrilik ve moment arasındaki ilişkiyi bulmaktır.



Şekil 2.2. M_x ve M_y momentleri etkisi altında bulunan plak

Plagın sadece M_x ve sadece M_y momentleri etkisi altında bulunması halinde eğrilik denklemleri Eş. 2.1’de gösterilmektedir.

$$\begin{aligned}
 \text{Sadece } M_x &\Rightarrow \left\{ \begin{aligned} K_x &= -\frac{\partial^2 w}{\partial x^2} = \frac{12M_x}{Et^3} \\ K_y &= -\frac{\partial^2 w}{\partial y^2} = -\frac{\nu 12M_x}{Et^3} \end{aligned} \right\} \\
 \text{Sadece } M_y &\Rightarrow \left\{ \begin{aligned} K_x &= -\frac{\partial^2 w}{\partial x^2} = -\frac{\nu 12M_y}{Et^3} \\ K_y &= -\frac{\partial^2 w}{\partial y^2} = \frac{12M_y}{Et^3} \end{aligned} \right\} \quad (2.1)
 \end{aligned}$$

Plak malzemesinin lineer elastik olduğu kabul edildiği için süperpozisyon ilkesi geçerlidir. Süperpozisyon uygulanınca Eş. 2.2 elde edilir.

$$\begin{aligned}
K_x &= -\frac{\partial^2 w}{\partial x^2} = \frac{12M_x}{Et^3} - \frac{\nu 12M_y}{Et^3} = \frac{12(M_x - \nu M_y)}{Et^3} \\
K_y &= -\frac{\partial^2 w}{\partial y^2} = \frac{12M_y}{Et^3} - \frac{\nu 12M_x}{Et^3} = \frac{12(M_y - \nu M_x)}{Et^3}
\end{aligned}
\tag{2.2}$$

K_x eşitliğinden M_x değeri çekilir ve K_y eşitliğinde yerine konulursa Eş. 2.3'deki gibi M_x ve M_y değerleri bulunur.

$$\begin{aligned}
M_x &= -D \times \left\{ \frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right\} \\
M_y &= -D \times \left\{ \frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right\}
\end{aligned}
\tag{2.3}$$

Plak eğilme rijitliğinin formülasyonu Eş. 2.4'de gösterilmektedir.

$$D = \frac{Et^3}{12(1-\nu^2)} \tag{2.4}$$

Eğilme için plak genel denklemini bu formüllerin de yardımıyla Eş. 2.5'deki gibi elde edilir.

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{p_z(x, y)}{D} \tag{2.5}$$

Dördüncü dereceden diferansiyel denklem olan plak genel denklemini çözmek için, her sınır noktaları için 2 sınır durumuna ihtiyaç vardır. Eğilme, burulma momentleri ve kesme kuvvetleri ve sınır durumlarını formüle etmek için, yer değiştirme bileşenleri dikkate alınmalıdır. Sınır ifadeleri aşağıda verilen 4 duruma göre sınıflandırılmaktadır. Çizelge 2.1'de bazı mesnet durumlarına göre matematiksel ifadeler verilmiştir [1].

- i. Geometrik sınır durumları
- ii. Statik sınır durumları (serbest uçlar)
- iii. Karma sınır durumları
- iv. Elastik mesnet ve sınırlamalar

Çizelge 2.1. Bazı mesnet durumlarına göre matematiksel ifadeler

x=a DAKİ MESNET TİPİ	MATEMATİKSEL İFADE
Basit Mesnet	$(w)_{x=a}=0; (m_x)_{x=a}=\left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right)_{x=a}=0$
Sabit Kenar	$(w)_{x=a}=0; \left(\frac{\partial w}{\partial x}\right)_{x=a}=0$
Serbest Kenar	$(m_x)_{x=a} = \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right)_{x=a} = 0$ $(v_x)_{x=a} = \left[\frac{\partial^3 w}{\partial x^3} + (2-\nu) \frac{\partial^3 w}{\partial x \partial y^2}\right]_{x=a} = 0$
Kısmen sabit Kenar	$(w_x)_{x=a} = 0$ $\left(\frac{\partial w}{\partial x}\right)_{x=a} = (p)^{-1} D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right)_{x=a}$
Elastik Mesnet	$(m_x)_{x=a} = \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right)_{x=a} = 0$ $(w_x)_{x=a} = p^{-1} D \left[\frac{\partial^3 w}{\partial x^3} + (2-\nu) \frac{\partial^3 w}{\partial x \partial y^2}\right]_{x=a}$
Elastik Mesnet ve Sınırlamalar	$(w_x)_{x=a} = p^{-1} D \left[\frac{\partial^3 w}{\partial x^3} + (2-\nu) \frac{\partial^3 w}{\partial x \partial y^2}\right]_{x=a}$ $\left(\frac{\partial w}{\partial x}\right)_{x=a} = (p)^{-1} D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2}\right)_{x=a}$

Plak genel denklemlerinin çözümünde “klasik yöntem” ve “sayısal ve yaklaşık yöntem” olmak üzere genellikle iki yöntem kullanılmaktadır.

2.2 Klasik Yöntemler

Plak problemlerinde, nispeten basit sınır ve yük koşulları ve basit geometriye sahip plakların çözümünde kullanılır. Fourier serileri, Navier çözümlemesi ve Levy yöntemi olmak üzere başlıca 3 klasik yöntem vardır.

2.2.1. Fourier Serileri

Fourier serileri mekanik problemlerin çözümünde geniş bir kullanım alanına sahip olduğu gibi plak teorisinde de kullanılmaktadır. Elastisite teorisinde, birçok problemin analitik çözümleri için Fourier serileri vazgeçilmezdir. Fourier serileri tek ve çift Fourier serileri olmak üzere ikiye ayrılır.

Tek Fourier Serileri

Sin ve Cos terimlerini içeren rastgele seçilmiş $y=f(x)$ fonksiyonunu düşünürsek, eğer bu fonksiyon periyodik bir fonksiyonsa ve sonsuz terimli bir seriyi ifade ediyorsa Tek Fourier serisi tam açılımı Eş. 2.6'daki gibidir.

$$f(x) = \frac{1}{2} A_0 + \sum_{n=1}^{\infty} A_{n1} \cos nwx + \sum_{n=1}^{\infty} B_{n1} \sin nwx \quad (2.6)$$

Formüldeki “w” doğal açısal frekansı ifade etmektedir ve $2\pi/T$ ’ ye eşittir [12].

Çift Fourier Serileri

Plakların statik ve dinamik analizlerinde, sinüs serileriyle ifade edilen 2 değişkenli $f(x,y)$ fonksiyonu kullanılır. Fonksiyon x ve y doğrultusunda yarım açılımlı sinüs serilerinin çarpımından oluşmuştur ve Eş. 2.7’de ifade edilmiştir.

$$f(x,y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} P_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \quad (2.7)$$

Çift Fourier serilerinde kullanılan açılım katsayıları bazı durumlar için çizelge 2.2’de verilmiştir [12].

Navier Çözümlemesi

Navier, bütün kenarlarından basit mesnetlenmiş plakların eğilme denklemlerini çift trigonometrik serilerle ifade etmiştir. Navier yönteminde deplasmanlar Eş. 2.8’de gösterildiği gibidir.

$$w(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} W_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \quad (2.8)$$

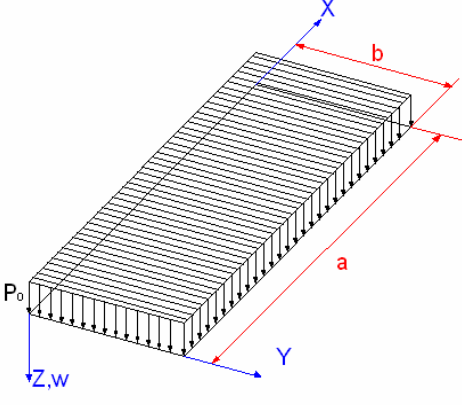
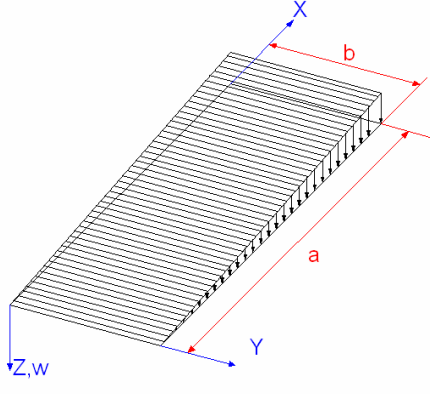
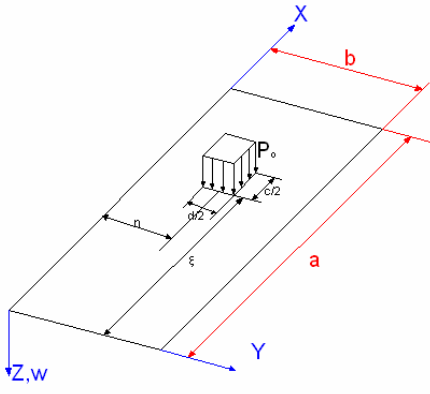
Plak üzerindeki yükün ifadesi çift sinüs serileriyle Eş. 2.9 ‘daki gibi olur.

$$p_z(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} P_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}, (m, n = 1, 2, 3, \dots) \quad (2.9)$$

Deplasman ve yük ifadeleri Eş. 2.5’deki plak genel denkleminde yerine koyulursa Eş. 2.10’daki cebirsel ifade elde edilir.

$$W_{mn} = \left(\frac{m^4 \pi^4}{a^4} + \frac{2m^2 n^2 \pi^4}{a^2 b^2} + \frac{n^4 \pi^4}{b^4} \right) \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} = \frac{1}{D} P_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \quad (2.10)$$

Çizelge 2.2. Bazı önemli yükleme tipleri

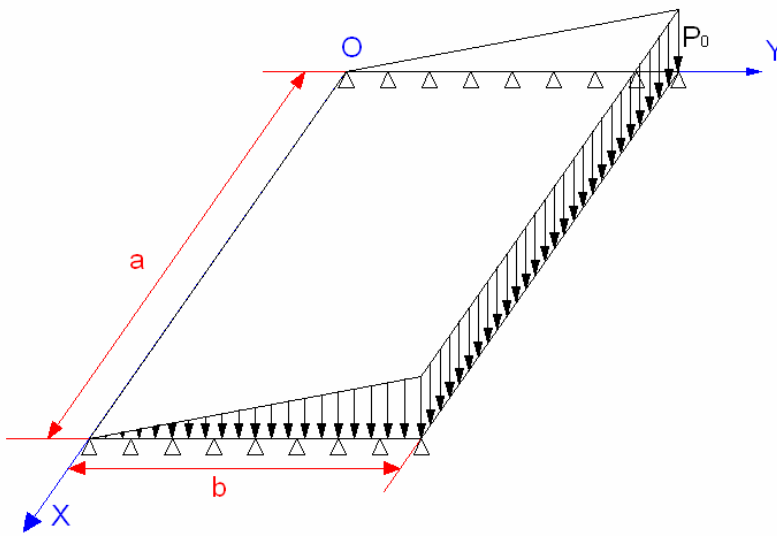
No.	YÜK	AÇILIM KATSAYILARI
	$p_z(x, y) = \sum_m \sum_n P_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b}$	P_{mn}
1		$P_{mn} = \frac{16p_0}{\pi^2 mn}$ $(m, n = 1, 3, 5, \dots)$
2		$P_{mn} = -\frac{8p_0 \cos m\pi}{\pi^2 mn}$ $(m, n = 1, 3, 5, \dots)$
3		$P_{mn} = \frac{16p_0}{\pi^2 mn} \sin \frac{m\pi \xi}{a} \sin \frac{n\pi \eta}{b}$ $x \sin \frac{m\pi c}{2a} \sin \frac{n\pi d}{2b}$ $(m, n = 1, 3, 5, \dots)$

Plaktaki momentler de Eş.2.11' deki gibi ifade edilir [12].

$$\begin{aligned}
 m_x &= \pi^2 D \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \left[\left(\frac{m}{a} \right)^2 + \nu \left(\frac{n}{b} \right)^2 \right] W_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \\
 m_y &= \pi^2 D \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \left[\left(\frac{n}{b} \right)^2 + \nu \left(\frac{m}{a} \right)^2 \right] W_{mn} \sin \frac{m\pi x}{a} \sin \frac{n\pi y}{b} \\
 m_{xy} &= -\pi^2 D (1-\nu) \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{mn}{ab} W_{mn} \cos \frac{m\pi x}{a} \cos \frac{n\pi y}{b}
 \end{aligned} \tag{2.11}$$

Levy Yöntemi

Levy yöntemi iki kenarından basit mesnetlenmiş dikdörtgen plaklar için bir analiz yöntemidir (Şekil 2.3).



Şekil 2.3. İki kenarı basit mesnetli dikdörtgen plak

Levy yönteminin uygulanabilmesi için yüklerin x eksenine boyunca bütün kesitlerde sabit olması ve $x=0$ ve $x=a$ kenarlarında plak, basit mesnetlenmiş olması gerekmektedir. Deplasman ifadesi Eş. 2.12'de seri şeklinde verilmektedir.

$$w_H(x, y) = \sum_{m=1}^{\infty} Y_m(y) \sin \frac{m\pi x}{a} \quad (2.12)$$

Plak genel denkleminin de kullanılmasıyla beraber Eş. 2.13 elde edilir [12].

$$w(x, y) = \sum_{m=1}^{\infty} W_m \sin \frac{m\pi x}{a} + \sum_{m=1}^{\infty} \left(A_m \cosh \frac{m\pi y}{a} + B_m \frac{m\pi y}{a} \sinh \frac{m\pi y}{b} \right) \sin \frac{m\pi x}{a} \quad (2.13)$$

2.3 Sayısal ve Yaklaşık Yöntemler

Plak geometrisinin basit, yükleme ve sınır koşullarının karmaşık olmadığı sistemlerde klasik yöntemin kullanılması daha uygun olmaktadır. Plak sistemlerinin daha zorlaştığı durumlarda sayısal ve yaklaşık yöntemlerle çözüm yapmak daha uygun olacaktır.

Szillard (1974) şu konuyu belirtmektedir [1].

“Dijital bilgisayarların keşfinden sonra sayısal yöntem çok daha önem kazanmış ve yüksek hızlı elektronik bilgisayarlarda, sayısal problem çözümlerinde algoritma kullanılmaya başlanmıştır. Bu algoritmalar bilgisayarlara, bilgisayar programları ile tanıtılmış ve böylece problem çözümlerinde bir tür hata daha ortaya çıkmıştır; makine hatası. Bilgisayar kullanımında sonuçların kontrol edilebilmesi için bağımsız bir yoldan tekrar çözülmesi gerekmektedir. Bu da insan tarafından çözüldüğü takdirde bir de insan hata faktörü etkili olacaktır”.

2.3.1 Sonlu Farklar Yöntemi

Sonlu fark yöntemi, elemanların parçalara ayrılarak uygulandığı bir yöntemdir. Bir kirişi h aralıklarıyla k, l, m, n, o gibi noktalara ayırarak hesap yapıldığı düşünülebilir. “ m ” noktasındaki fonksiyon z_i olarak gösterilsin. Fonksiyonun türevleri Eş. 2.14’de gösterildiği gibi olacaktır.

$$\begin{aligned}
\left(\frac{dz}{dx}\right)_i &= \left(\frac{z_n - z_m}{h} + \frac{z_m - z_l}{h}\right) \div 2 = \frac{z_n - z_l}{2h} \\
\left(\frac{d^2z}{dx^2}\right)_i &= \left(\frac{\frac{z_n - z_m}{h} - \frac{z_m - z_l}{h}}{h}\right) = \frac{z_n - 2z_m + z_l}{h^2} \\
\left(\frac{d^3z}{dx^3}\right)_i &= \left(\frac{z_o - 2z_n + 2z_l - z_k}{2h^3}\right) \\
\left(\frac{d^4z}{dx^4}\right)_i &= \left(\frac{z_o - 4z_n + 6z_m - 4z_l + z_k}{h^4}\right)
\end{aligned} \tag{2.14}$$

Kirişin gerçek diferansiyel denklemi Eş. 2.15'deki gibidir.

$$\frac{\partial^4 z}{\partial x^4} = \frac{q}{EI} \tag{2.15}$$

Eş. 2.14, Eş. 2.15' de yerine konarak ve kesme hatası da hesaba katılarak Eş. 2.16 elde edilir. Bu denklem 2.derece doğruluklu sayısal eşitliktir.

$$\frac{z_k - 4z_l + 6z_m - 4z_n + z_o}{\Delta x^4} = \frac{q}{EI} + \theta(\Delta x^2) \tag{2.16}$$

Kesme hatasının üzerindeki derece aynı zamanda doğruluk derecesinin de göstergesidir. Kullanılan eşitlik 2. derece doğruluklu merkezi sonlu farklar eşitliğidir. Merkezi sonlu farklar açılımı olduğu gibi ileri ve geri sonlu farklar açılımları da vardır. Fakat gerçeğe en yakın hesaplamalar merkezi sonlu farklar açılımı ile elde edilmektedir. Plaklarda nokta ayırımı iki yönlü olduğu için kirişlerden daha karışık algoritmalar oluşturulacaktır. Bu çalışmada plakların analizleri yapılırken 2. derece doğruluklu merkezi sonlu farklar yöntemi uygulanmıştır.

3. SONLU FARKLAR METODUYLA YAPILAN ÇÖZÜMLER

3.1 Sonlu Fark Metodunun Programda Uygulanışı

Sonlu farklar yönteminde plaklar, sonlu farklar ağı denilen parçalara ayrılmaktadır. Bu noktaların deformasyonları hesaplanarak plak yüzeyinin deplasman ifadeleri bulunur. Bu çalışmada; işlevsel, bilgisayar kullanımına uygun ve hata payının az olması sebebi ile sonlu farklar yöntemi kullanılmıştır.

Programlama dili olarak “Matlab”, yazılım tabanı olarak “Guide” kullanılmıştır. Program iki kısımdan oluşmaktadır. Birinci kısımda plakların 2.derece doğruluklu merkezi sonlu farklar metodu ile tekil yük ve yayılı yük altında analizleri yapılabilmektedir (Şekil 3.1). Şekil 3.1’de gösterilen ekranda plağa ait malzeme ve analiz özellikleri olan Young modülü (Elastisite katsayısı), Poisson oranı, plak uzunluğu, plak kalınlığı, ayrılan parça sayısı, parça uzunluğu, tekil yük veya yayılı yükün büyüklüğü ve etkidiği nokta verilerinin girişi yapılmaktadır. Plak üzerinde tekil yük varsa yayılı yük değeri, yayılı yük durumu hesap edilecekse tekil yük değeri “0” girilmelidir. Program kullanıcının 2 tip birim sistemi kullanmasına olanak sağlamaktadır.

Hesaplamalarda plak genel denklemi kullanılmaktadır. Eşitliğin sağ tarafındaki plak eğilme rijitliği Eş. 3.1’den elde edilmektedir.

$$D = \frac{Et^3}{12(1-\nu^2)} \quad (3.1)$$

Burada Young modülü, plak kalınlığı ve Poisson oranı kullanıcı tarafından girilmektedir. Yine eşitliğin sağ tarafındaki yükün büyüklüğü kullanıcı tarafından girilen veriler arasındadır.

plakanalizi

SONLU FARKLAR YÖNTEMİ İLE PLAK ANALİZİ

PLAK DEPLASMANLARINI GÖSTER

Plak Bilgileri

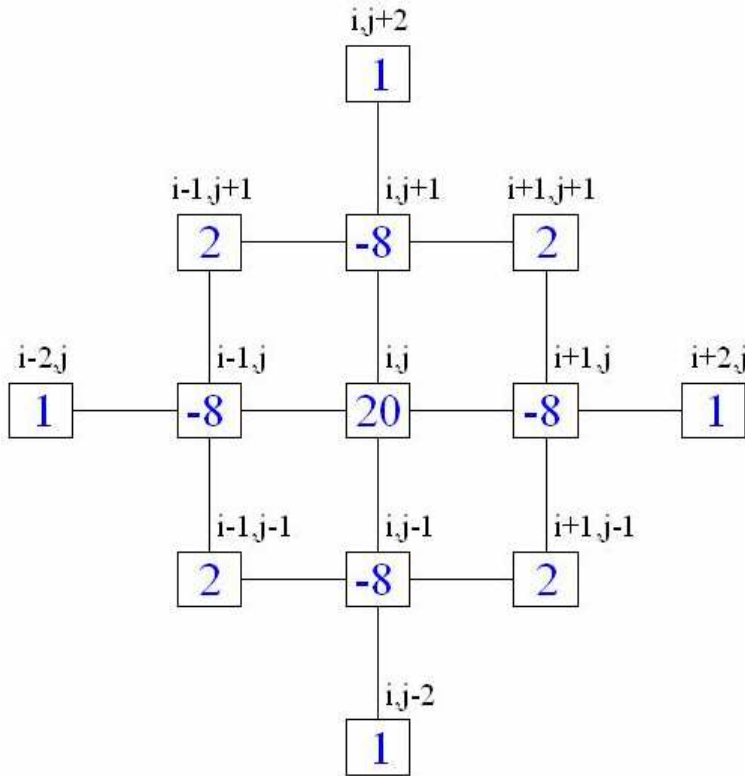
Plakın üzerindeki yayılı yükü girin	kg/cm ²	MPa
Plakın üzerindeki tekil yükü girin	kg	N
Tekil yükün etki ettiği noktayı girin		
Elastisite Modülü	kg/cm ²	MPa
Poisson Oranı		
Plak Kalınlığı	cm	mm
Aynılması istenilen parça sayısı (X yönü)		
Aynılması istenilen parça sayısı (Y yönü)		
Parça Uzunluğu	cm	mm

24-Dec-2007

Darbe Yüklü Altında Analiz

Şekil 3.1. Program analiz ekranı

Eşitliğin sol tarafı için ise sonlu farklar ağ sisteminden faydalanılmaktadır. Burada kullanılan matris sistemlerindeki katsayılar, deplasman ifadesinin x ve y' ye göre türevlerinden gelmektedir. Ağ sistemindeki katsayılar Şekil 3.2'de gösterildiği gibidir.



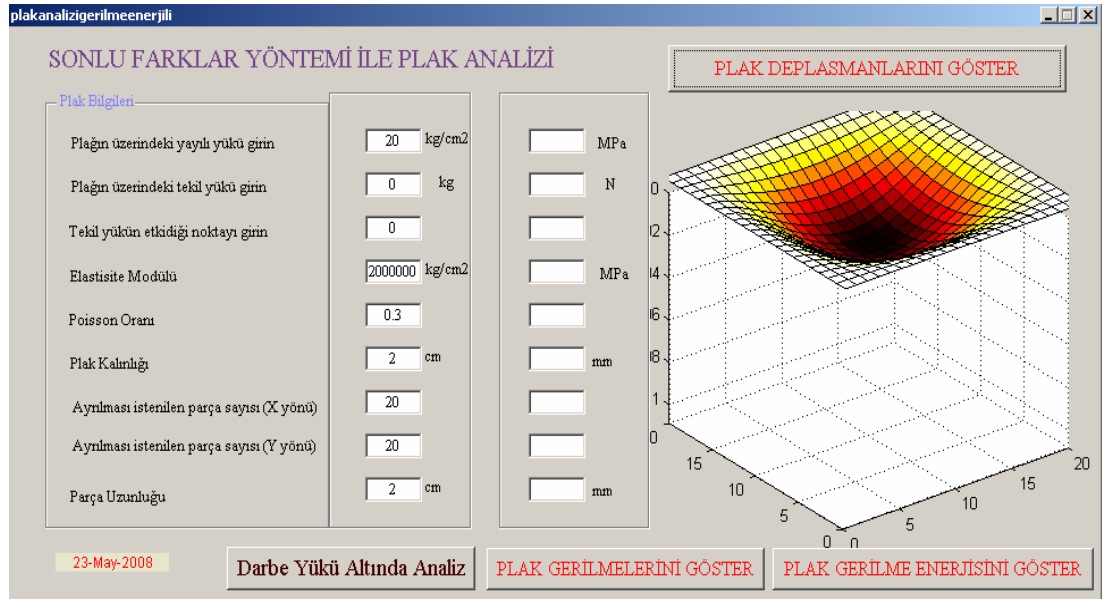
Şekil 3.2. Sonlu farklar ağ sistemindeki nokta katsayıları

Her nokta için bu katsayılar algoritma meydana getirildikten sonra diğer kısımlara sıfır yerleştirilerek matris oluşturulur (Çizelge 3.1).

Çizelge 3.1. Sonlu farklar metodu matris eşitliği

$$\begin{bmatrix} \dots\dots\dots \\ \dots\dots\dots \\ \dots\dots\dots \\ \dots\dots\dots \end{bmatrix}_{(n \times n)} \begin{Bmatrix} z_{11} \\ \dots \\ \dots \\ z_{nn} \end{Bmatrix}_{(n \times 1)} = \begin{Bmatrix} q\Delta x^4 \div D \\ q\Delta x^4 \div D \\ q\Delta x^4 \div D \\ q\Delta x^4 \div D \end{Bmatrix}_{(n \times 1)}$$

Bu eşitliğin sonucunda deplasman değerleri bulunarak grafik üzerinde gösterimi yapılır (Şekil 3.3).



Şekil 3.3. Plak deplasmanları (cm)

Plakların gerilme değerleri de bulunan deplasman değerlerinin bir fonksiyonu olarak elde edilebilirler. Gerilmelerin, normal gerilmelerin bir fonksiyonu olarak denklemleri Eş. 3.2’de verilmektedir [12]. Tekil ve yayılı yükün altında hesaplanan gerilmeler statik, darbe yükü altında hesaplanan gerilmeler dinamik gerilmelerdir.

$$\begin{aligned}\sigma_x &= \frac{E}{1-\nu^2}(\epsilon_x + \nu\epsilon_y) \\ \sigma_y &= \frac{E}{1-\nu^2}(\epsilon_y + \nu\epsilon_x) \\ \tau_{xy} &= G\gamma_{xy}\end{aligned}\tag{3.2}$$

Eğrilik ve deplasmanla olan ilişkileri de Eş.3.3’de gösterilmektedir [12]

$$\begin{aligned}
\sigma_x &= -\frac{Ez}{1-\nu^2} (K_x + \nu K_y) = -\frac{Ez}{1-\nu^2} \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \\
\sigma_y &= -\frac{Ez}{1-\nu^2} (K_y + \nu K_x) = -\frac{Ez}{1-\nu^2} \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \\
\tau_{xy} &= -\frac{Ez}{1+\nu} K_{xy} = -\frac{Ez}{1+\nu} \frac{\partial^2 w}{\partial x \partial y}
\end{aligned} \tag{3.3}$$

Bununla birlikte plakların gerilme enerjisi de programla hesaplanabilmektedir. Genel plak teorisine göre deformasyonun gerilme enerjisi, kayma ve kalınlık gerilmesinin ihmal edildiği durumlardaki eğilmenin oluşturduğu gerilmedir [13]. Gerilme enerjisi Eş.3.4’de gösterilmektedir.

$$\begin{aligned}
U_e &= \frac{1}{2} \int (\sigma_x \varepsilon_x + \sigma_y \varepsilon_y + \sigma_z \varepsilon_z) dVol \\
&= \frac{1}{2} \int (-w_{xx} \int_{-h/2}^{h/2} \sigma_x z dz - w_{yy} \int_{-h/2}^{h/2} \sigma_y z dz - 2w_{xy} \int_{-h/2}^{h/2} \tau_{xy} z dz) dAlan \\
&= \frac{1}{2} \int [M_x (-w_{xx}) + M_y (-w_{yy}) + 2M_{xy} (-w_{xy})] dAlan
\end{aligned} \tag{3.4}$$

Moment-Eğilme ilişkilerini kullanarak yukarıdaki denklem Eş.3.5’de ki şekli alır.

$$U_e = \frac{D}{2} \int \left[(\nabla^2 w)^2 - 2(1-\mu)(w_{xx}w_{yy} - w_{xy}^2) \right] dAlan \tag{3.5}$$

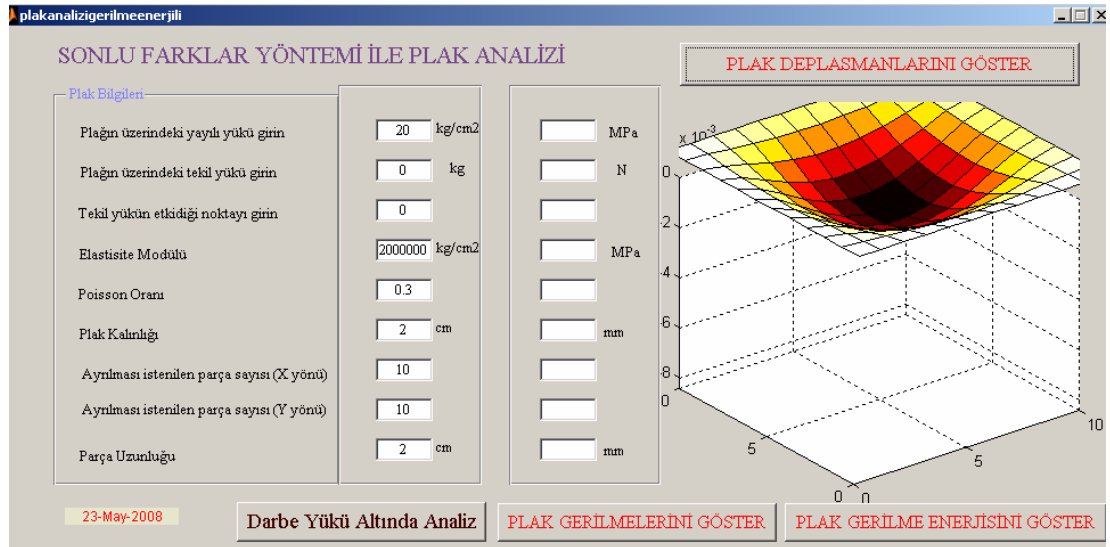
Plagin bütün kenarlarının basit mesnetlenmiş olduğu kabul edilirse eşitliğin sağ tarafındaki ikinci terim sıfırlanır ve denklem Eş. 3.6’daki ifadeyi alır.

$$U_e = \frac{D}{2} \iint (\nabla^2 w)^2 d_x d_y \tag{3.6}$$

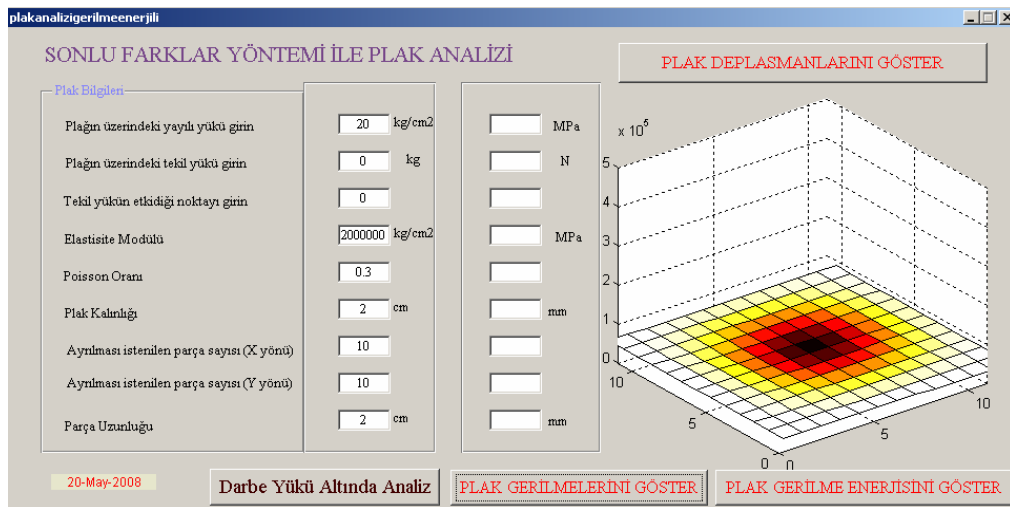
3.2 Örnekler

Örnek 1

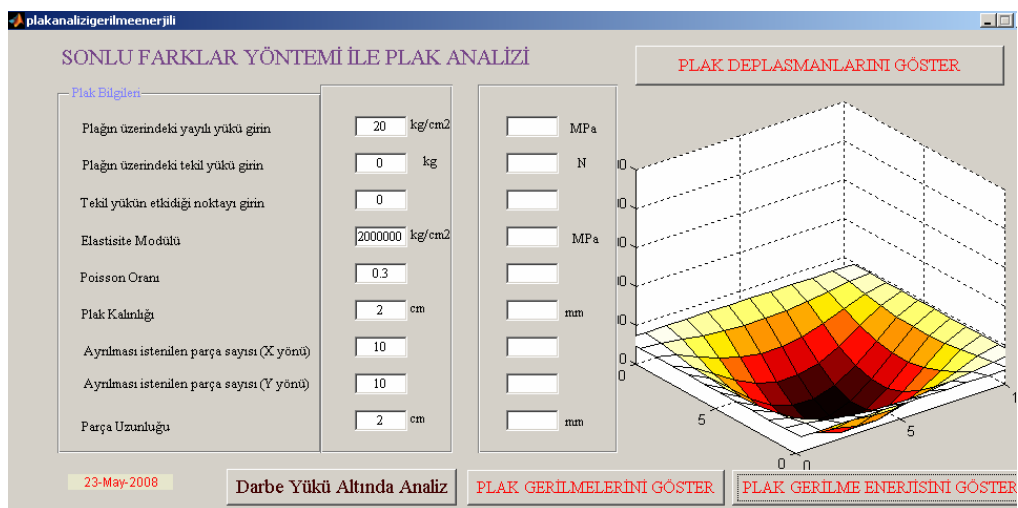
Düzgün yayılı yük etkisinde dört kenarından basit mesnetli kare plakta parça aralıkları $a/10$ olacak şekilde oluşturulan ağ sisteminde $\nu=0.3$ $E=2 \times 10^6$ ve $t=2$ cm için çözümdeki deplasman grafiği Şekil 3.4’de, gerilme grafiği Şekil 3.5’de ve gerilme enerjisi grafiği Şekil 3.6’da matris ile hesaplanan değerler Çizelge 3.2, Çizelge 3.3 ve Çizelge 3.4’de verilmektedir.



Şekil 3.4. Plak deplasmanları (cm)



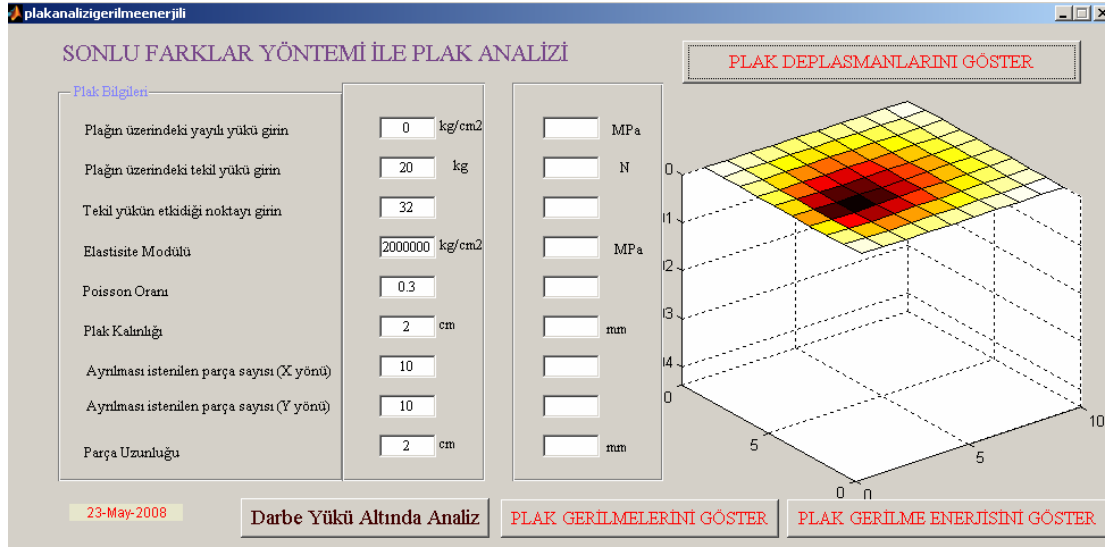
Şekil 3.5. Plak Gerilmeleri (kgf/cm²)



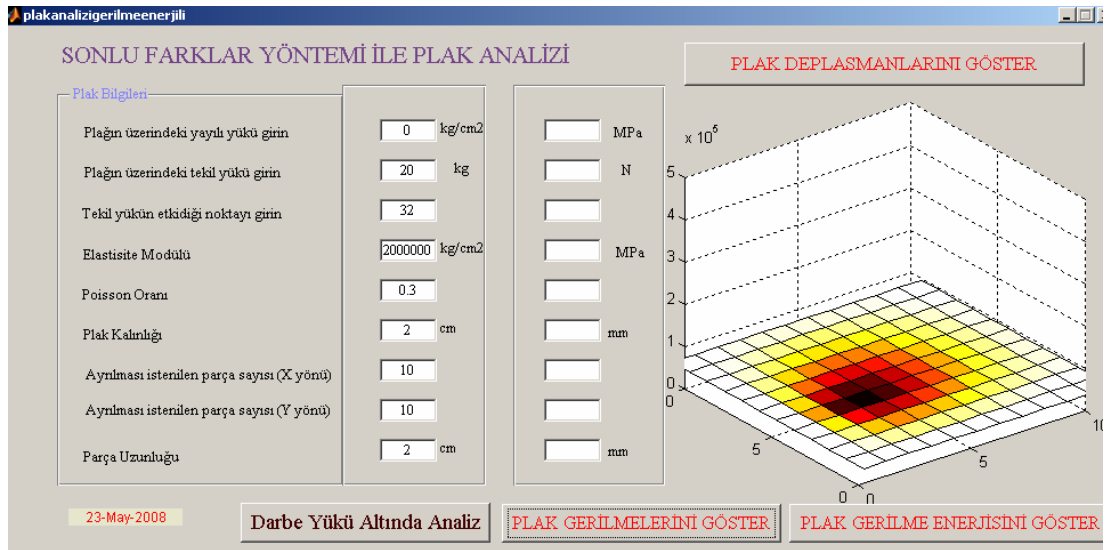
Şekil 3.6. Plak gerilme enerjisi(kgf/cm²/cm³)

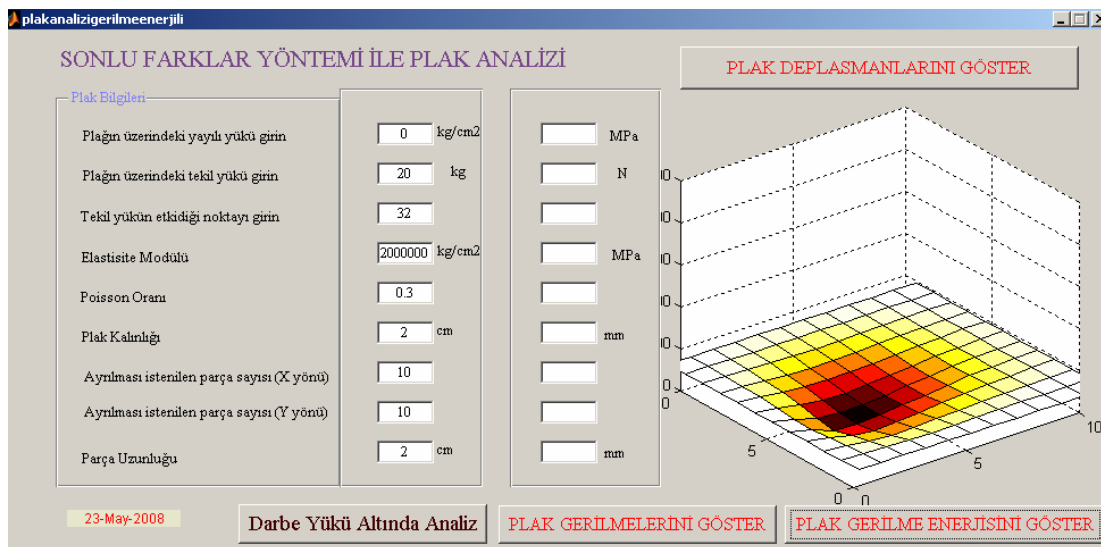
Çizelge 3.2. Plak deplasman değerleri (cm)

[illegible]



Şekil 3.7. Plak deplasmanları (cm)

Şekil 3.8. Plak gerilmeleri (kgf/cm²)



Şekil 3.9. Plak gerilme enerjisi ($\text{kgf/cm}^2/\text{cm}^3$)

Çizelge 3.5. Plak deplasman değerleri (cm)

$$\mathbf{F} =$$

1.0e-003 *

0.0157	0.0423	0.0560	0.0544	0.0443	0.0320	0.0210	0.0123	0.0060	0.0018
0.0554	0.1344	0.1701	0.1626	0.1326	0.0973	0.0654	0.0397	0.0207	0.0076
0.1112	0.2556	0.3044	0.2809	0.2258	0.1659	0.1126	0.0694	0.0370	0.0151
0.1604	0.3624	0.4025	0.3590	0.2859	0.2113	0.1447	0.0894	0.0464	0.0159
0.1230	0.2874	0.3539	0.3397	0.2848	0.2176	0.1521	0.0948	0.0485	0.0153
0.0780	0.1947	0.2628	0.2718	0.2409	0.1911	0.1369	0.0862	0.0439	0.0134
0.0447	0.1200	0.1745	0.1919	0.1782	0.1461	0.1068	0.0678	0.0343	0.0102
0.0229	0.0664	0.1025	0.1182	0.1138	0.0955	0.0707	0.0450	0.0225	0.0064
0.0093	0.0300	0.0491	0.0588	0.0579	0.0493	0.0366	0.0230	0.0111	0.0028
0.0019	0.0080	0.0142	0.0175	0.0175	0.0149	0.0109	0.0066	0.0029	0.0005

Çizelge 3.6. Plak gerilmeleri (kgf/cm²)

yerilme =

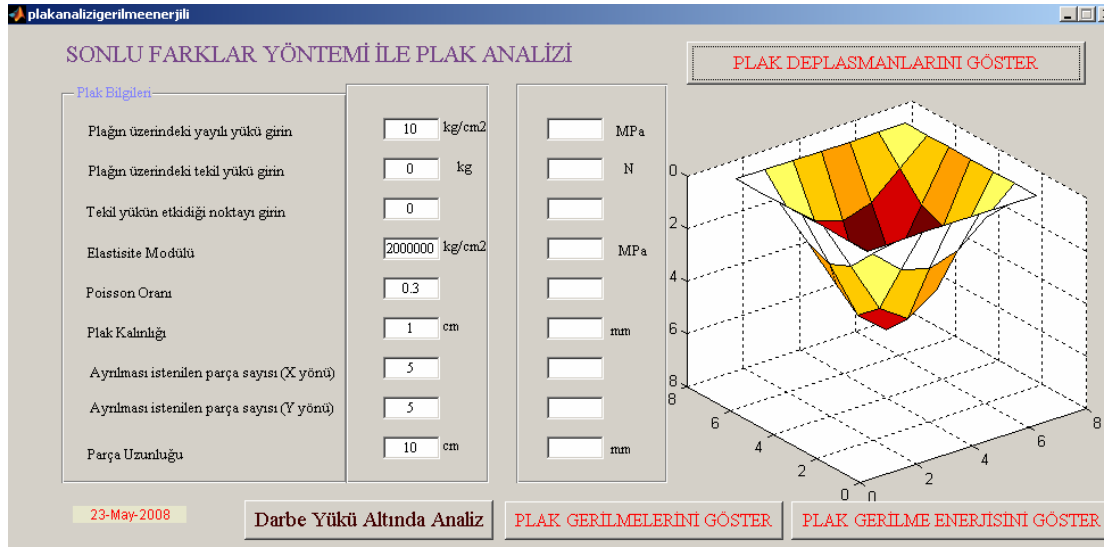
[illegible]

Çizelge 3.7. Plak gerilme enerjisi değerleri (kgf/cm²/cm³)

Gerilme_Enerjisi=											
0	0	0	0	0	0	0	0	0	0	0	0
0	1.4014	3.7707	4.9957	4.8523	3.9472	2.8532	1.8699	1.0954	0.5327	0.1610	0
0	4.9443	11.9879	15.1773	14.5055	11.8284	8.6808	5.8304	3.5412	1.8429	0.6763	0
0	9.9183	22.7999	27.1576	25.0530	20.1415	14.8009	10.0402	6.1872	3.3014	1.3494	0
0	14.3116	32.3281	35.9034	32.0258	25.5044	18.8442	12.9033	7.9773	4.1390	1.4152	0
0	10.9719	25.6402	31.5659	30.3037	25.4062	19.4083	13.5712	8.4603	4.3262	1.3641	0
0	6.9561	17.3648	23.4469	24.2487	21.4878	17.0499	12.2091	7.6936	3.9136	1.1925	0
0	3.9835	10.7023	15.5643	17.1140	15.8925	13.0312	9.5225	6.0504	3.0617	0.9094	0
0	2.0403	5.9193	9.1449	10.5461	10.1477	8.5228	6.3091	4.0136	2.0031	0.5706	0
0	0.8317	2.6783	4.3789	5.2414	5.1671	4.3979	3.2634	2.0522	0.9881	0.2540	0
0	0.1681	0.7132	1.2640	1.5653	1.5638	1.3304	0.9728	0.5883	0.2555	0.0427	0
0	0	0	0	0	0	0	0	0	0	0	0

Örnek 3

Düzgün yayılı yük etkisinde dört kenarından basit mesnetli kare plakta parça aralıkları $a/4$ olacak şekilde oluşturulan ağ sisteminde $\nu=0.3$ $E=2 \times 10^6$ ve $t=1$ cm için çözümdeki grafik Şekil 3.10'da, matris ile hesaplanan değerler Çizelge 3.8, Çizelge 3.9 ve Çizelge 3.10'da verilmektedir.



Şekil 3.10. Plak deplasmanları (cm)

Çizelge 3.8. Plak deplasman değerleri (cm)

M =

0	0	0	0	0	0	0
0	0.0927	0.1852	0.2198	0.1852	0.0927	0
0	0.1852	0.3787	0.4529	0.3787	0.1852	0
0	0.2198	0.4529	0.5429	0.4529	0.2198	0
0	0.1852	0.3787	0.4529	0.3787	0.1852	0
0	0.0927	0.1852	0.2198	0.1852	0.0927	0
0	0	0	0	0	0	0

Çizelge 3.9. Plak gerilmeleri (kgf/cm²)

gerilme =

1.0e+005 *

0	0	0	0	0	0	0
0	1.0182	2.0346	2.4153	2.0346	1.0182	0
0	2.0346	4.1618	4.9772	4.1618	2.0346	0
0	2.4153	4.9772	5.9657	4.9772	2.4153	0
0	2.0346	4.1618	4.9772	4.1618	2.0346	0
0	1.0182	2.0346	2.4153	2.0346	1.0182	0
0	0	0	0	0	0	0

Çizelge 3.10. Plak gerilme enerjisi değerleri (kgf/cm²/cm³)

Gerilme_Enerjisi=

0	0	0	0	0	0	0
0	165.2980	330.3181	392.1268	330.3181	165.2980	0
0	330.3181	675.6627	808.0368	675.6627	330.3181	0
0	392.1268	808.0368	968.5208	808.0368	392.1268	0
0	330.3181	675.6627	808.0368	675.6627	330.3181	0
0	165.2980	330.3181	392.1268	330.3181	165.2980	0
0	0	0	0	0	0	0

Çizelge 3.8’de görüldüğü üzere plakta oluşan en büyük deformasyon plak ortasındadır ve değeri 0.5429 cm.’ dir. Problemin çözümünde SAP programı kullanılırsa plak ortasındaki deformasyon değeri 0.5655 cm., analitik çözüm kullanılırsa 0.5671 cm. çıkmaktadır. Hata oranı 0.0242 dir.

Plakların sonlu farklar yöntemi ile deformasyon, gerilme ve gerilme enerjileri programla hesaplanmaktadır. Tekil yük ve yayılı yük altında hesaplanan gerilmeler statik, darbe yükleri altında hesaplanan gerilme de dinamik gerilmelerdir.

4. DARBE YÜKLERİ

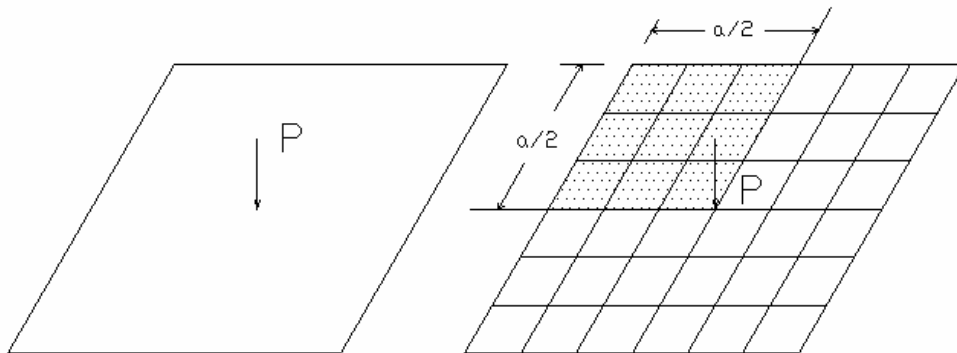
4.1 Darbe Yüklemesinin Tanımı Özellikleri ve Programda Uygulanışı

Bir kuvvet aniden elastik bir kütleye uygulanırsa (çarpma ile), cisim üzerinde belirli bir hız ile hareket eden bir gerilme dalgası oluşur. Buna darbe yükleme denmektedir[10]. Darbe yüklemede bazı kabuller yapılmaktadır. Bunlar;

- 1 - Darbe sırasında enerji kaybı yoktur,
- 2 - Çarpan cismin tekrar sıçramadığı, plağa yapışarak bir parçası gibi kalacaktır. Böylece plağın ataleti çarpan cismin yanında ihmal edilebilir olacaktır[11].

Birinci ve ikinci kabuller arasında dinamik açıdan uygunluk yoktur. Çünkü çarpan cisimlerin çarpmadan sonra birlikte hareket ettiği çarpışma türü “Plastik Çarpışma”dır. Böyle bir çarpışmada enerji kaybının olmadığı kabulü, daha yüksek dinamik gerilmelere neden olacağından, daha emniyetli sonuçlar veren bir kabul olarak düşünülmüştür.

Bu çalışmada plakların darbe yükleri altında analizleri için plaklar kirişlerden oluşan ızgara sistemiymiş gibi modellenmiştir. Şekil 4.1’de görüldüğü üzere öncelikle plakla aynı ölçülere sahip kiriş çerçevesine etkiyen statik yük altında deplasman ifadesi bulunur.



Şekil 4.1. Plak ve aynı ölçülerdeki kiriş çerçevesi üzerindeki statik yük

$$\{d_i\} = [k_{ij}]^{-1} \{P_j\} \quad (4.2)$$

Böylece simetrik yapının köşe noktasında yani kiriş çerçeve sisteminin orta noktasında ki deplasman ifadesi Eş.4.3'deki denklem gibi elde edilir

$$[w] = 5.60518 * 10^{-3} \frac{Pa^2}{EI} \quad (4.3)$$

Bu nokta için bulunan Eş.4.3'de ki denklem plağın orta noktasındaki deplasman değerine eşitlenerek Eşdeğer rijitlik(EI) ifadesi bulunur.

Hesaplamalarda kolaylık sağlamak için kirişler üzerine gelen darbe yükünün oluşturduğu gerilme enerjisi ile aynı miktarda gerilme enerjisi oluşturan statik yük bulunur. Eşdeğer statik yükleme için Eş.4.4'de verilen denklem kullanılır.

$$P_m = \sqrt{\frac{96U_m EI}{L^3}} \quad (4.4)$$

Eşitlikte verilen U_m darbe yüklemesini yapan cismin çarpmadan önceki kinetik enerjisine eşit olan maksimum gerilme enerjisidir. İfadesi Eş.4.5'de verilmektedir.

$$U_m = \frac{1}{2} m v_0^2 \quad (4.5)$$

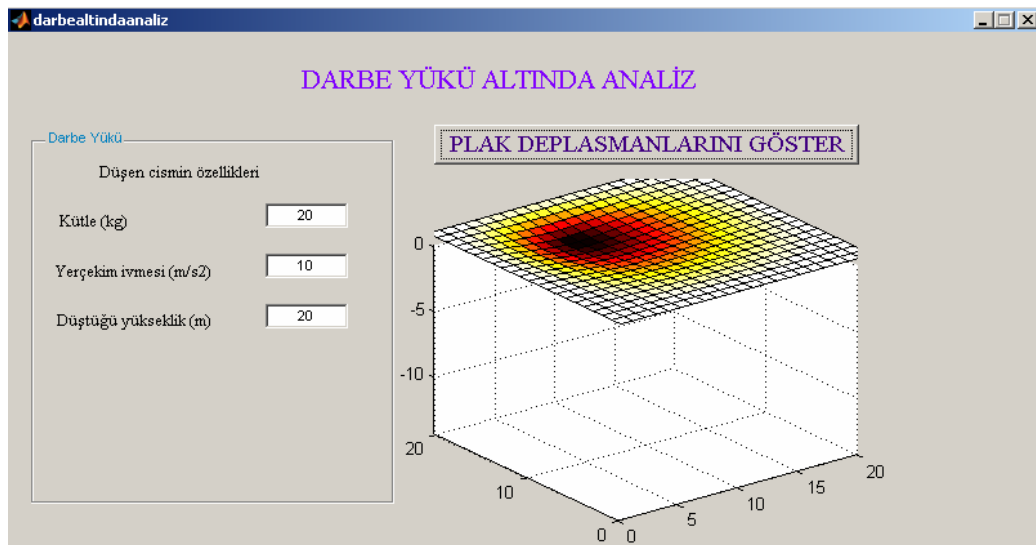
Yukarıda belirtildiği gibi çarpmada enerji kaybı ihmal edilirse enerjinin korunumundan cismin havadaki potansiyel enerjisi çarpmadan önceki kinetik enerjisine eşittir. Böylece U_m ifadesi Eş.4.6'daki ifadeyi alır.

$$U_m = m * g * h \quad (4.6)$$

Eş.4.4, Eş.4.5 ve Eş.4.6'nın kullanılması ile eşdeğer statik yük bulunur ve plaklar üzerinde statik yükleme durumu ile çözüm yapılır ve grafiksel sonuçlar elde edilir. Programla elde edilen değerler darbe yükünün statik yükten daha fazla bir tesir oluşturduğunu ve deplasmanlarının daha büyük değerler aldığını göstermektedir. Darbe yüklemesi için programda oluşturulan grafik arayüzü Şekil 4.2'de çıkan deplasmanlar Şekil 4.3'de verilmektedir.



Şekil 4.2. Programda darbe yükü için oluşturulan arayüz

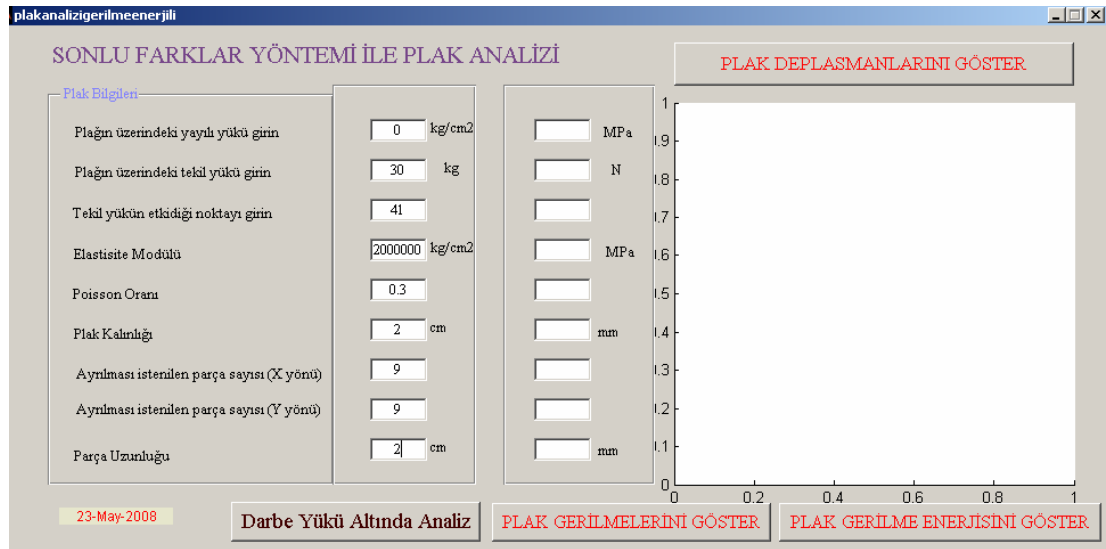


Şekil 4.3. Plak deplasmanları (cm)

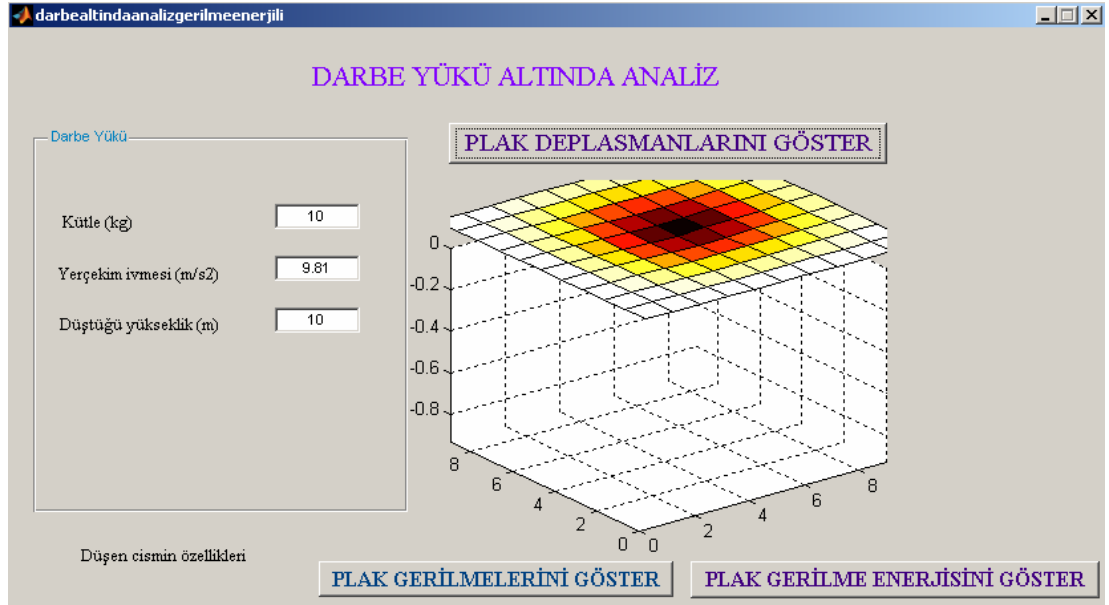
4.2 Örnekler

Örnek 1

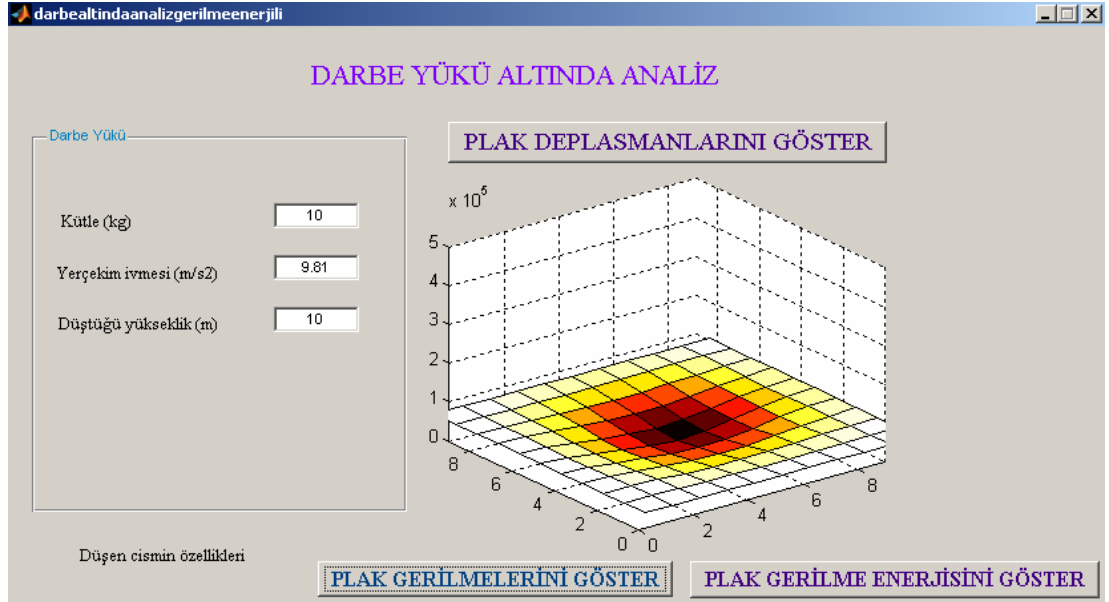
Dört kenarından basit mesnetli kare plağın orta noktasına 10 m. yükseklikten düşen 10 kg.lık bir cismin darbe yüklemesi yapması durumunda parça aralıkları $a/9$ olacak şekilde oluşturulan ağ sisteminde $\nu=0.3$ $E=2 \times 10^6$ ve $t=2$ cm için çözümdeki grafik Şekil 4.4, Şekil 4.5, Şekil 4.6 ve Şekil 4.7’de matris ile hesaplanan değerler Çizelge 4.1, Çizelge 4.2 ve Çizelge 4.3’de verilmektedir.



Şekil 4.4. Darbe yüklemesinde girilen değerler



Şekil 4.5. Darbe yüklemesi altında deplasman grafiği (cm)



Şekil 4.6. Darbe yüklemesi altında gerilme grafiği (kgf/cm²)

Çizelge 4.3. Darbe yüklemesi altında gerilme enerjisi değerleri ($\text{kgf/cm}^2/\text{cm}^3$)

Gerilme_Enerjisi=

1.0e+003 *

0	0	0	0	0	0	0	0	0	0	0
0	0.0160	0.0640	0.1230	0.1715	0.1905	0.1715	0.1230	0.0640	0.0160	0
0	0.0648	0.2136	0.3896	0.5339	0.5914	0.5339	0.3896	0.2136	0.0648	0
0	0.1272	0.3975	0.7153	0.9815	1.0921	0.9815	0.7153	0.3975	0.1272	0
0	0.1815	0.5581	1.0087	1.4043	1.5844	1.4043	1.0087	0.5581	0.1815	0
0	0.2047	0.6290	1.1507	1.6367	1.8934	1.6367	1.1507	0.6290	0.2047	0
0	0.1815	0.5581	1.0087	1.4043	1.5844	1.4043	1.0087	0.5581	0.1815	0
0	0.1272	0.3975	0.7153	0.9815	1.0921	0.9815	0.7153	0.3975	0.1272	0
0	0.0648	0.2136	0.3896	0.5339	0.5914	0.5339	0.3896	0.2136	0.0648	0
0	0.0160	0.0640	0.1230	0.1715	0.1905	0.1715	0.1230	0.0640	0.0160	0
0	0	0	0	0	0	0	0	0	0	0

Örnek 2

Dört kenarından basit mesnetli kare plağın orta noktasına 50 m. yükseklikten düşen 30 kg.lık bir cismin darbe yüklemesi yapması durumunda parça aralıkları $a/15$ olacak şekilde oluşturulan ağ sisteminde $\nu=0.4$ $E=2 \times 10^6$ ve $t=2$ cm için çözümdeki grafik Şekil 4.8, Şekil 4.9, Şekil 4.10 ve Şekil 4.11’de matris ile hesaplanan değerler Çizelge 4.4, Çizelge 4.5 ve Çizelge 4.6’da verilmektedir.

plakanalizi

SONLU FARKLAR YÖNTEMİ İLE PLAK ANALİZİ

PLAK DEPLASMANLARINI GÖSTER

Plak Bilgileri

Plağın üzerindeki yayılı yükü girin: kg/cm^2

Plağın üzerindeki tekil yükü girin: kg

Tekil yükün etki ettiği noktayı girin:

Elastisite Modülü: kg/cm^2

Poisson Oranı:

Plak Kalınlığı: cm

Ayrılması istenilen parça sayısı (X yönü):

Ayrılması istenilen parça sayısı (Y yönü):

Parça Uzunluğu: cm

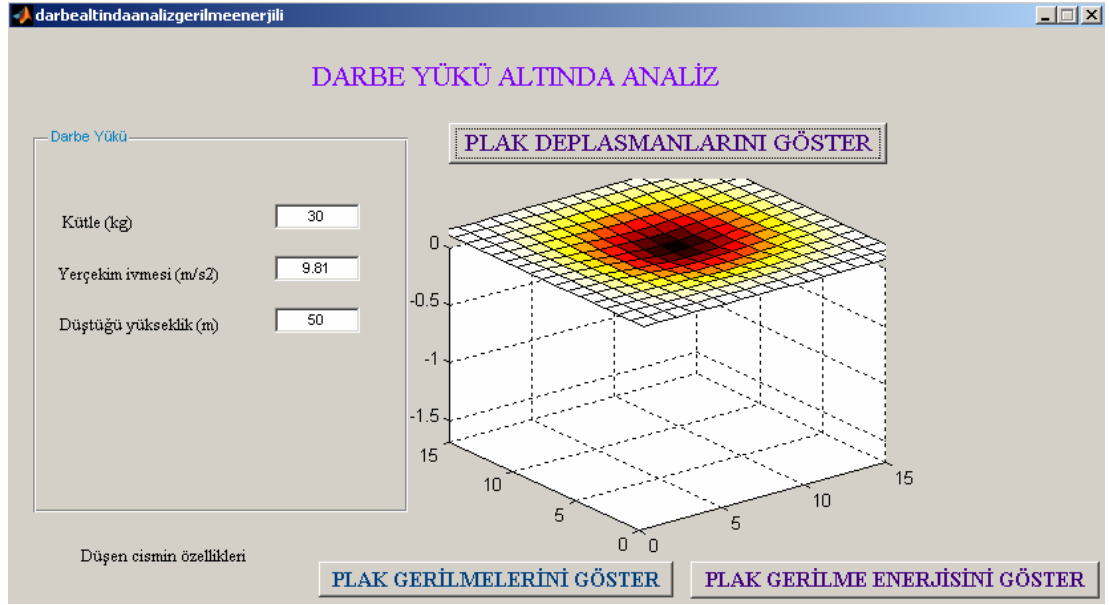
08-May-2008

Darbe Yüklü Altında Analiz

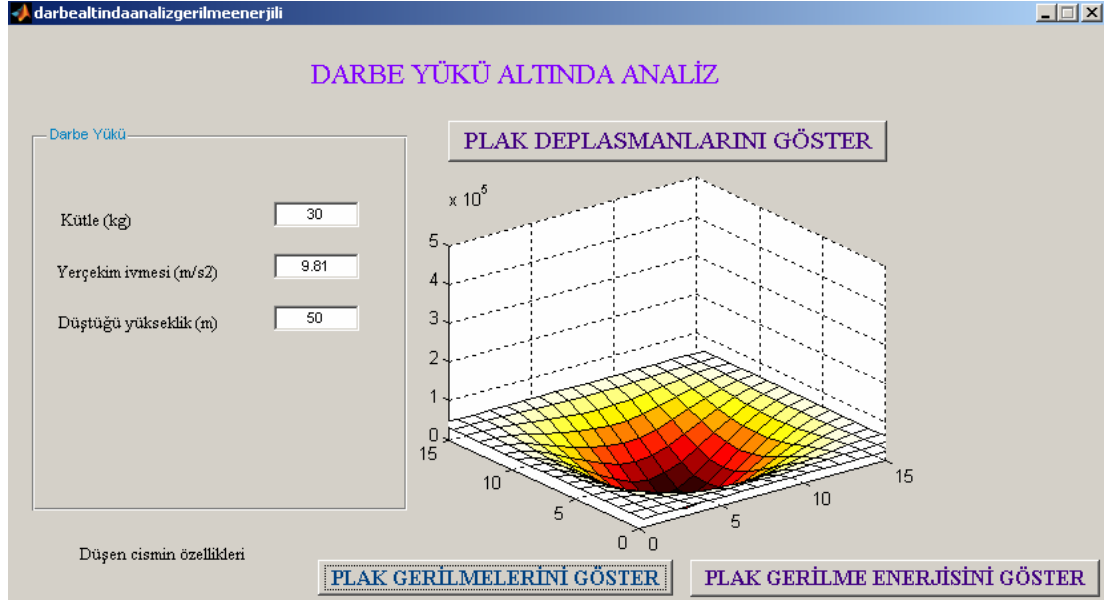
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

0 0.2 0.4 0.6 0.8 1

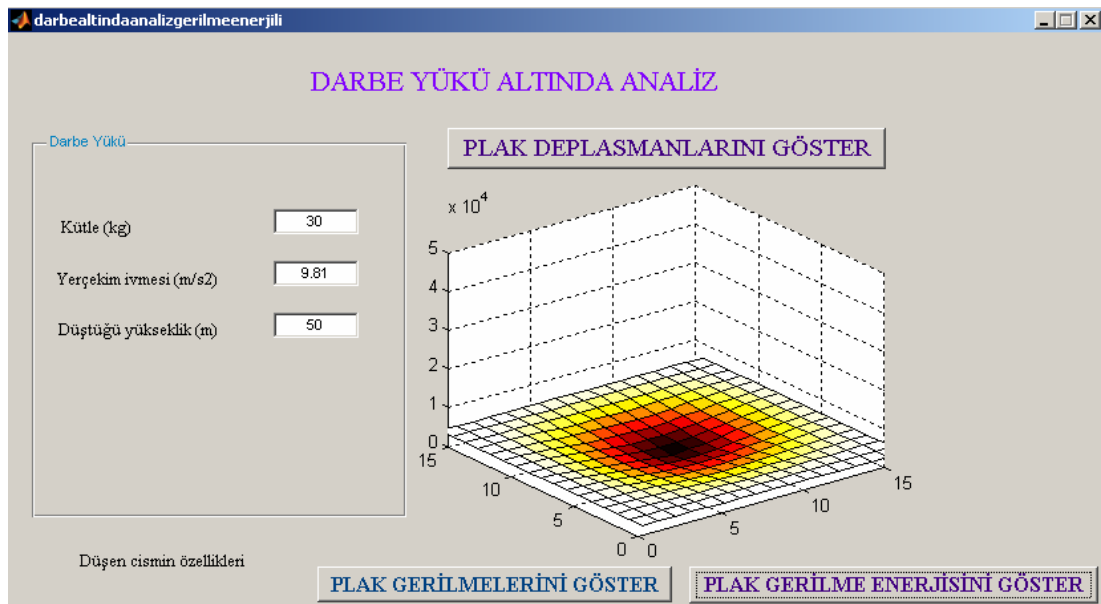
Şekil 4.8. Darbe yüklemesinde girilen değerler



Şekil 4.9. Darbe yüklemesi altında deplasman grafiği (cm)



Şekil 4.10. Darbe yüklemesi altında gerilme grafiği (kgf/cm²)



Şekil 4.11. Darbe yüklemesi altında gerilme enerjisi ($\text{kgf/cm}^2/\text{cm}^3$)

Çizelge 4.4. Darbe yüklemesi altında deplasman değerleri (cm)

$$M =$$
[illegible]

Çizelge 4.5. Darbe yüklemesi altında gerilme değerleri(kgf/cm²)

Gerilme =

1.0e+005 *

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0010	0.0087	0.0216	0.0371	0.0528	0.0663	0.0755	0.0787	0.0755	0.0663	0.0528	0.0371	0.0216	0.0087	0.0010	0	0	0
0	0.0087	0.0371	0.0802	0.1308	0.1815	0.2251	0.2546	0.2651	0.2546	0.2251	0.1815	0.1308	0.0802	0.0371	0.0087	0	0	0
0	0.0217	0.0804	0.1652	0.2633	0.3614	0.4460	0.5037	0.5243	0.5037	0.4460	0.3614	0.2633	0.1652	0.0804	0.0217	0	0	0
0	0.0375	0.1318	0.2645	0.4171	0.5703	0.7036	0.7957	0.8289	0.7957	0.7036	0.5703	0.4171	0.2645	0.1318	0.0375	0	0	0
0	0.0537	0.1840	0.3653	0.5738	0.7852	0.9724	1.1047	1.1534	1.1047	0.9724	0.7852	0.5738	0.3653	0.1840	0.0537	0	0	0
0	0.0678	0.2295	0.4535	0.7127	0.9798	1.2225	1.4003	1.4687	1.4003	1.2225	0.9798	0.7127	0.4535	0.2295	0.0678	0	0	0
0	0.0776	0.2609	0.5149	0.8112	1.1222	1.4145	1.6400	1.7343	1.6400	1.4145	1.1222	0.8112	0.5149	0.2609	0.0776	0	0	0
0	0.0811	0.2722	0.5373	0.8479	1.1778	1.4966	1.7561	1.8800	1.7561	1.4966	1.1778	0.8479	0.5373	0.2722	0.0811	0	0	0
0	0.0776	0.2609	0.5149	0.8112	1.1222	1.4145	1.6400	1.7343	1.6400	1.4145	1.1222	0.8112	0.5149	0.2609	0.0776	0	0	0
0	0.0678	0.2295	0.4535	0.7127	0.9798	1.2225	1.4003	1.4687	1.4003	1.2225	0.9798	0.7127	0.4535	0.2295	0.0678	0	0	0
0	0.0537	0.1840	0.3653	0.5738	0.7852	0.9724	1.1047	1.1534	1.1047	0.9724	0.7852	0.5738	0.3653	0.1840	0.0537	0	0	0
0	0.0375	0.1318	0.2645	0.4171	0.5703	0.7036	0.7957	0.8289	0.7957	0.7036	0.5703	0.4171	0.2645	0.1318	0.0375	0	0	0
0	0.0217	0.0804	0.1652	0.2633	0.3614	0.4460	0.5037	0.5243	0.5037	0.4460	0.3614	0.2633	0.1652	0.0804	0.0217	0	0	0
0	0.0087	0.0371	0.0802	0.1308	0.1815	0.2251	0.2546	0.2651	0.2546	0.2251	0.1815	0.1308	0.0802	0.0371	0.0087	0	0	0
0	0.0010	0.0087	0.0216	0.0371	0.0528	0.0663	0.0755	0.0787	0.0755	0.0663	0.0528	0.0371	0.0216	0.0087	0.0010	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Çizelge 4.6. Darbe yüklemesi altında gerilme enerjisi değerleri (kgf/cm²/cm³)

Gerilme_Enerjisi=

1.0e+005 *

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0010	0.0087	0.0216	0.0371	0.0528	0.0663	0.0755	0.0787	0.0755	0.0663	0.0528	0.0371	0.0216	0.0087	0.0010	0	0	0
0	0.0087	0.0371	0.0802	0.1308	0.1815	0.2251	0.2546	0.2651	0.2546	0.2251	0.1815	0.1308	0.0802	0.0371	0.0087	0	0	0
0	0.0217	0.0804	0.1652	0.2633	0.3614	0.4460	0.5037	0.5243	0.5037	0.4460	0.3614	0.2633	0.1652	0.0804	0.0217	0	0	0
0	0.0375	0.1318	0.2645	0.4171	0.5703	0.7036	0.7957	0.8289	0.7957	0.7036	0.5703	0.4171	0.2645	0.1318	0.0375	0	0	0
0	0.0537	0.1840	0.3653	0.5738	0.7852	0.9724	1.1047	1.1534	1.1047	0.9724	0.7852	0.5738	0.3653	0.1840	0.0537	0	0	0
0	0.0678	0.2295	0.4535	0.7127	0.9798	1.2225	1.4003	1.4687	1.4003	1.2225	0.9798	0.7127	0.4535	0.2295	0.0678	0	0	0
0	0.0776	0.2609	0.5149	0.8112	1.1222	1.4145	1.6400	1.7343	1.6400	1.4145	1.1222	0.8112	0.5149	0.2609	0.0776	0	0	0
0	0.0811	0.2722	0.5373	0.8479	1.1778	1.4966	1.7561	1.8800	1.7561	1.4966	1.1778	0.8479	0.5373	0.2722	0.0811	0	0	0
0	0.0776	0.2609	0.5149	0.8112	1.1222	1.4145	1.6400	1.7343	1.6400	1.4145	1.1222	0.8112	0.5149	0.2609	0.0776	0	0	0
0	0.0678	0.2295	0.4535	0.7127	0.9798	1.2225	1.4003	1.4687	1.4003	1.2225	0.9798	0.7127	0.4535	0.2295	0.0678	0	0	0
0	0.0537	0.1840	0.3653	0.5738	0.7852	0.9724	1.1047	1.1534	1.1047	0.9724	0.7852	0.5738	0.3653	0.1840	0.0537	0	0	0
0	0.0375	0.1318	0.2645	0.4171	0.5703	0.7036	0.7957	0.8289	0.7957	0.7036	0.5703	0.4171	0.2645	0.1318	0.0375	0	0	0
0	0.0217	0.0804	0.1652	0.2633	0.3614	0.4460	0.5037	0.5243	0.5037	0.4460	0.3614	0.2633	0.1652	0.0804	0.0217	0	0	0
0	0.0087	0.0371	0.0802	0.1308	0.1815	0.2251	0.2546	0.2651	0.2546	0.2251	0.1815	0.1308	0.0802	0.0371	0.0087	0	0	0
0	0.0010	0.0087	0.0216	0.0371	0.0528	0.0663	0.0755	0.0787	0.0755	0.0663	0.0528	0.0371	0.0216	0.0087	0.0010	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Örneklerden de elde ettiğimiz sonuçlarla görülmektedir ki darbe yüklemesi altında alınan deplasman değerleri grafiksel olarak bir farklılık göstermesede değerleri bakımından statik yükleme durumundan farklıdır. Yapılacak son örnekte plağın orta noktasına etkiyen tekil yük ve darbe yükün oluşturdukları gerilme enerjilerinin farklılıkları irdelenecektir.

Örnek 3

Dört kenarından basit mesnetli kare plağın orta noktasına uygulanan 20 kg'lık tekil yük ve 20 m. yükseklikten düşen 30 kg.lık bir cismin darbe yüklemesi yapması

5. SONUÇLARIN DEĞERLENDİRİLMESİ

Bu çalışmada plakların tekil, yayılı ve darbe yükler altında analizleri incelenmiş bununla ilgili olarak Matlab programlama dilinde bir program yazılmıştır. Program sonucunda;

- 1) Sonlu farklar yönteminde kullanılması gereken karmaşık ve manüel olarak çözülemeyecek algoritmalar program yardımıyla kolay ve işlevsel bir şekilde oluşturulmuştur.
- 2) Plak analizleri için kullanıcılara veri girişi kolay ve görsel açıdan rahat bir program yapılmış, program sonuçlarının hem matris olarak hesaplanan değerlerinin hem de görsel olarak grafik çıktılarının elde edilmesi sağlanmıştır.
- 3) Analizler için yapılan Matlab yazılımının dosya büyüklüğünün az olması ve hızlı çalışmasından dolayı diğer programlarda arayüz olarak kullanılmasına olanak sağlanmıştır.
- 4) Programda darbe yükleri altında da analiz yapılmış, plakların kiriş sistemlerinden oluşan ızgara sistemiymiş gibi düşünülerek deplasman hesapları yapılmış ve sonuçlar grafiksel olarak elde edilmiştir.
- 5) Darbe yükleri altında analizde plak kiriş sistemi olarak düşünülerek darbe yükü ile statik yük arasında bağıntı kurulmuş, deplasmanla birlikte gerilme ve gerilme enerjisi değerleride hesaplanmış ve grafiklerle gösterilmiştir.
- 6) Program sonuçlarının analitik sonuçlarla karşılaştırması yapılarak hata payının çok düşük olduğu ve bunun da örneklerle teyidi yapılmıştır.

KAYNAKLAR

1. Szilard, R., “Theory and analysis of plates”, Newmark, N.M., Hall, W.J., **Prentice-Hall, Inc.Englewood Cliffs**, New Jersey, 15-27 (1974).
2. Berktaş, İ., “Plak teorisi ve uygulamaları”,**Yıldız Teknik Üniversitesi Müh. Fak. İnşaat Müh. Bölümü.**, İstanbul, 237 (1992).
3. Ergün, A., Kumbasar, N., “İnce plaklar için geliştirilmiş sonlu farklar yöntemi”, **İ.T.Ü Dergisi**, 2(1): 36 (2003).
4. Smetankina, N.V., Shupikov, A.N., Sotrikhin, S.Yu., Yareschenko, V.G., “Dynamic response of an elliptic plate to impact loading:Theory and experiment”, **International Journal of Impact Engineering**, Norway, 1-13 (2005).
5. Franz, T., “Photoelastic study of the mechanic behaviour of orthotropic composite plates subjected to impact”, **Centre for Research in Computational and Applied Mechanics (CERECAM)**, Cape Town, 1-10 (2001).
6. Ollson, R., Donadon, M.V., Falzon, B.G., “Delamination treshold load for dynamic impact on plates”, **International Journal of Impact Engineering**, Norway, 1-18 (2006).
7. Yoon, K., Heo, S., Song, K., Jung, Y., “Dynamic impact analysis of the grid structure using multi-point constraint(MPC) equation under the lateral impact load”, **Advanced Reactor Development Division**, South Korea, 1-8 (2004).
8. Abbas, H., Gupta, N.K., Alam, M., “Nonlinear response of concrete beams and plates under impact loading”, **International Journal of Impact Engineering**, Norway, 1-15 (2004).

9. Aslan, Z., Karakuzu, R., Okutan, B., “The response of laminated composite plates under low-velocity impact loading”, *Department of Mechanical Engineering*, İzmir, 1-9 (2002).
10. Young, W.C., “Roark’s formulas for stress and strain”, *McGraw-hill Inc.*, New York, 710 (2002).
11. Beer, F.P., “Mechanics of materials”, *McGraw-hill Inc.*, New York, 593 (1992).
12. Ugural, A.C., “Stresses in plates and shells”, *McGraw-hill Inc.*, New Jersey Institute of Technology, 93-95 (1999).
13. Varadan, T.K., Bashkar K. “Analysis of plates Theory and Problems”, **Narosa Publishing House**, 29 (1999).

EKLER

EK-1 Program Kodu

```

function varargout = ooo(varargin)

% OOO M-file for ooo.fig
%     OOO, by itself, creates a new OOO or raises the existing
%     singleton*.
%
%     H = OOO returns the handle to a new OOO or the handle to
%     the existing singleton*.
%
%     OOO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in OOO.M with the given input
arguments.
%
%     OOO('Property','Value',...) creates a new OOO or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before ooo_OpeningFunction gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to ooo_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 ooo

% Last Modified by GUIDE v2.5 20-May-2008 14:00:48

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @ooo_OpeningFcn, ...
                  'gui_OutputFcn',  @ooo_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 ooo is made visible.
function ooo_OpeningFcn(hObject, eventdata, handles, varargin)

```

EK-1 (Devam) Program Kodu

```
% 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 ooo (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

m_date=date;
ui_text_date = uicontrol('Units','pixels','Position',[60 25 100
15],...
    'Style','text','BackgroundColor',[.9 .9 .8],...
    'ForegroundColor',[1 0 0],'String',m_date,...
    'FontSize',9,'Fontweight','Demi',...
    'callback','new_project');

% --- Outputs from this function are returned to the command line.
function varargout = ooo_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;

function edit1_Callback(hObject, eventdata, handles)
global q
q = str2double(get(hObject,'string'));

% hObject    handle to edit1 (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 edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
%        as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (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.
```

EK-1 (Devam) Program Kodu

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
global E
E = str2double(get(hObject,'string'));

% hObject    handle to edit7 (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 edit7 as text
% str2double(get(hObject,'String')) returns contents of edit7
% as a double
% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (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 edit8_Callback(hObject, eventdata, handles)
global v
v = str2double(get(hObject,'string'));
if v>0.5
    v=0.5; set(hObject,'string','0.5');
end

% hObject    handle to edit8 (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 edit8 as text
% str2double(get(hObject,'String')) returns contents of edit8
% as a double
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (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.
```

EK-1 (Devam) Program Kodu

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
global t
t = str2double(get(hObject,'string'));

% hObject    handle to edit9 (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 edit9 as text
% str2double(get(hObject,'String')) returns contents of edit9
% as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (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 edit10_Callback(hObject, eventdata, handles)
global m
m = str2double(get(hObject,'string'));

% hObject    handle to edit10 (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 edit10 as text
% str2double(get(hObject,'String')) returns contents of
% edit10 as a double

function edit18_Callback(hObject, eventdata, handles)
global n
n = str2double(get(hObject,'string'));
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

EK-1 (Devam) Program Kodu

```
% Hints: get(hObject,'String') returns contents of edit18 as text
%         str2double(get(hObject,'String')) returns contents of
edit18 as a double

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (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 during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (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 edit11_Callback(hObject, eventdata, handles)
global c
c = str2double(get(hObject,'string'));

% hObject    handle to edit11 (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 edit11 as text
%         str2double(get(hObject,'String')) returns contents of
edit11 as a double

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

EK-1 (Devam) Program Kodu

```
% 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 pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global q;
global E;
global c;
global m;
global t;
global v;
global n;
global w;
global xx;
global xxx;

D = (E*t*t*t)/(12*(1-v*v));
xy = q*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

x=xy/2;
A0 = eye (o,o)*20;
A0(1:o-1:end)=22;
A0(o*p+m:o*o:end)=22;
A0(o*o-o*p-p:o*o:end)=22;
A0(o+2:o+1:o*p)=21;
A0(o*o-o*s-s:o+1:o*o-1)=21;
A0(o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0(o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0(2:o+1:o*o)=-8;
A0(o+1:o+1:o*o)=-8;
A0(3:o+1:o*o-o)=1;
A0(o*2+1:o+1:o*o)=1;
A0(o*m+m:o*m+m:end)=0;
A0(o*m+p:o*m+m:end)=0;
A0(o*i+m:o*m+m:end)=0;
A0(o*s+i:o*m+m:end)=0;
A0(o*p+i:o*m+m:end)=0;
```

EK-1 (Devam) Program Kodu

```

A0(o*p+g:o*m+m:end)=0;

A0(m+1:o+1:o*o-o*m)=-8;
A0(m+2:o+1:o*o-o*m)=2;
A0(o+m+1:o+1:o*o-o*m)=2;
A0(o*m+2*m:o*m+m:end)=0;
A0(o*p+2*m+1:o*m+m:end)=0;

A0(o*m+1:o+1:o*o)=-8;
A0(o*m+2:o+1:o*o)=2;
A0(o*i+1:o+1:o*o)=2;
A0(o*m*2+m:o*m+m:o*o)=0;
A0(o*p*2+o+i:o*m+m:o*o)=0;

A0(2*m+1:o+1:o*m*u)=1;

A0(o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
if xx>0
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*w*c*c/D;
B0(xx-1,1)=4*w*c*c/D;
B0(xx-2,1)=1*w*c*c/D;
B0(xx+1,1)=4*w*c*c/D;
B0(xx+2,1)=1*w*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
surf(-F)
colormap(hot);

axis([0 m 0 n -(50*xxx) 0.0000005])
else
    if xx==0
        B=B;
        y = inv(A0) * B;
        C=(1:1:m*n);
        DD = reshape(C, m, n);
        F=y(DD');
        wklwrite('matA.wk1', F, 1, 1)
        M = wklread('matA.wk1');
        M(m+2,n+2)=0
        surf(-M)
        colormap(hot);
        axis([0 m 0 n -(4*F(n/2,m/2)) 0.0000005])
    end
end

```

EK-1 (Devam) Program Kodu

```
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes1

function edit12_Callback(hObject, eventdata, handles)
global q
q = str2double(get(hObject,'string'));
% hObject    handle to edit12 (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 edit12 as text
%        str2double(get(hObject,'String')) returns contents of
edit12 as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit12 (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 edit13_Callback(hObject, eventdata, handles)
global E
E = str2double(get(hObject,'string'));
% hObject    handle to edit13 (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 edit13 as text
%        str2double(get(hObject,'String')) returns contents of
edit13 as a double
% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

EK-1 (Devam) Program Kodu

```
% 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 edit14_Callback(hObject, eventdata, handles)
global v
v = str2double(get(hObject,'string'));
if v>0.5
    v=0.5; set(hObject,'string','0.5');
end
% hObject      handle to edit14 (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 edit14 as text
%      str2double(get(hObject,'String')) returns contents of
edit14 as a double
% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit14 (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 edit15_Callback(hObject, eventdata, handles)
global t
t = str2double(get(hObject,'string'));
% hObject      handle to edit15 (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 edit15 as text
%      str2double(get(hObject,'String')) returns contents of
edit15 as a double
% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit15 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called
```

EK-1 (Devam) Program Kodu

```
% 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 edit16_Callback(hObject, eventdata, handles)
global m
m = str2double(get(hObject,'string'));

% hObject    handle to edit16 (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 edit16 as text
%       str2double(get(hObject,'String')) returns contents of
edit16 as a double
% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (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 edit17_Callback(hObject, eventdata, handles)
global c
c = str2double(get(hObject,'string'));
% hObject    handle to edit17 (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 edit17 as text
%       str2double(get(hObject,'String')) returns contents of
edit17 as a double
% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (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.
```

EK-1 (Devam) Program Kodu

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)
global n
n = str2double(get(hObject,'string'));
% hObject    handle to edit19 (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 edit19 as text
%        str2double(get(hObject,'String')) returns contents of
edit19 as a double

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (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 edit20_Callback(hObject, eventdata, handles)
global w
w = str2double(get(hObject,'string'));
% hObject    handle to edit20 (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 edit20 as text
%        str2double(get(hObject,'String')) returns contents of
edit20 as a double
% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit20 (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'))

```

EK-1 (Devam) Program Kodu

```

        set(hObject,'BackgroundColor','white');
    end

function edit22_Callback(hObject, eventdata, handles)
global xx
xx = str2double(get(hObject,'string'));
% hObject    handle to edit22 (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 edit22 as text
%        str2double(get(hObject,'String')) returns contents of
edit22 as a double

% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit22 (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 edit21_Callback(hObject, eventdata, handles)
global xx
xx = str2double(get(hObject,'string'));
% hObject    handle to edit21 (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 edit21 as text
%        str2double(get(hObject,'String')) returns contents of
edit21 as a double

% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit21 (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'))

```

EK-1 (Devam) Program Kodu

```

        set(hObject,'BackgroundColor','white');
    end

function edit23_Callback(hObject, eventdata, handles)
global w
w = str2double(get(hObject,'string'));
% hObject    handle to edit23 (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 edit23 as text
%        str2double(get(hObject,'String')) returns contents of
edit23 as a double

% --- Executes during object creation, after setting all properties.
function edit23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit23 (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 pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
openfig('darbealtindaanalizgerilmeenerjili.fig','new','visible')
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function pushbutton2_CreateFcn(hObject, eventdata, handles)

```

EK-1 (Devam) Program Kodu

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

global q;
global E;
global c;
global m;
global t;
global v;
global n;
global w;
global xx;
global xxx;
D = (E*t*t*t)/(12*(1-v*v));
xy = q*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

x=xy/2;
A0 = eye (o,o)*20;
A0 (1:o-1:end)=22;
A0 (o*p+m:o*o:end)=22;
A0 (o*o-o*p-p:o*o:end)=22;
A0 (o+2:o+1:o*p)=21;
A0 (o*o-o*s-s:o+1:o*o-1)=21;
A0 (o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0 (o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0 (2:o+1:o*o)=-8;
A0 (o+1:o+1:o*o)=-8;
A0 (3:o+1:o*o-o)=1;
A0 (o*2+1:o+1:o*o)=1;
A0 (o*m+m:o*m+m:end)=0;
A0 (o*m+p:o*m+m:end)=0;
A0 (o*i+m:o*m+m:end)=0;
A0 (o*s+i:o*m+m:end)=0;
A0 (o*p+i:o*m+m:end)=0;
A0 (o*p+g:o*m+m:end)=0;

A0 (m+1:o+1:o*o-o*m)=-8;
A0 (m+2:o+1:o*o-o*m)=2;
A0 (o+m+1:o+1:o*o-o*m)=2;
A0 (o*m+2*m:o*m+m:end)=0;
```

EK-1 (Devam) Program Kodu

```

A0(o*p+2*m+1:o*m+m:end)=0;

A0(o*m+1:o+1:o*o)=-8;
A0(o*m+2:o+1:o*o)=2;
A0(o*i+1:o+1:o*o)=2;
A0(o*m*2+m:o*m+m:o*o)=0;
A0(o*p*2+o+i:o*m+m:o*o)=0;

A0(2*m+1:o+1:o*m*u)=1;

A0(o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
if xx>0
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*w*c*c/D;
B0(xx-1,1)=4*w*c*c/D;
B0(xx-2,1)=1*w*c*c/D;
B0(xx+1,1)=4*w*c*c/D;
B0(xx+2,1)=1*w*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');

else
    if xx==0
        B=B;
        y = inv(A0) * B;
        C=(1:1:m*n);
        DD = reshape(C, m, n);
        F=y(DD');

    end
end

pi=3.14159265;
pi4=pi*pi*pi*pi;
Dpi4=D*pi4;
Dp=Dpi4/32;
uzz=m*c;
uz=n*c;
terimbir=(3*uzz)/(uz*uz*uz);
terimiki=(3*uz)/(uzz*uzz*uzz);
terimuc=2/(uz*uzz);
terimler=terimbir+terimiki+terimuc;
gk=Dp*terimler;
GM=F*gk;
wklwrite('matA.wk1', GM, 1, 1)
Gerilme_Enerjisi = wklread('matA.wk1');
```

EK-1 (Devam) Program Kodu

```

Gerilme_Enerjisi(m+2,n+2)=0
surf(-Gerilme_Enerjisi)
colormap(hot);

axis([0 m 0 n -(50*xxx) 500])
% hObject      handle to pushbutton6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% hObject      handle to pushbutton8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
global q;
global E;
global c;
global m;
global t;
global v;
global n;
global w;
global xx;
global xxx;

D = (E*t*t*t)/(12*(1-v*v));
xy = q*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

x=xy/2;
A0 = eye (o,o)*20;
A0(1:o-1:end)=22;
A0(o*p+m:o*o:end)=22;
A0(o*o-o*p-p:o*o:end)=22;
A0(o+2:o+1:o*p)=21;
A0(o*o-o*s-s:o+1:o*o-1)=21;
A0(o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0(o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0(2:o+1:o*o)=-8;
A0(o+1:o+1:o*o)=-8;
A0(3:o+1:o*o-o)=1;
A0(o*2+1:o+1:o*o)=1;

```

EK-1 (Devam) Program Kodu

```

A0(o*m+m:o*m+m:end)=0;
A0(o*m+p:o*m+m:end)=0;
A0(o*i+m:o*m+m:end)=0;
A0(o*s+i:o*m+m:end)=0;
A0(o*p+i:o*m+m:end)=0;
A0(o*p+g:o*m+m:end)=0;

A0(m+1:o+1:o*o-o*m)=-8;
A0(m+2:o+1:o*o-o*m)=2;
A0(o+m+1:o+1:o*o-o*m)=2;
A0(o*m+2*m:o*m+m:end)=0;
A0(o*p+2*m+1:o*m+m:end)=0;

A0(o*m+1:o+1:o*o)=-8;
A0(o*m+2:o+1:o*o)=2;
A0(o*i+1:o+1:o*o)=2;
A0(o*m*2+m:o*m+m:o*o)=0;
A0(o*p*2+o+i:o*m+m:o*o)=0;

A0(2*m+1:o+1:o*m*u)=1;

A0(o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
if xx>0
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*w*c*c/D;
B0(xx-1,1)=4*w*c*c/D;
B0(xx-2,1)=1*w*c*c/D;
B0(xx+1,1)=4*w*c*c/D;
B0(xx+2,1)=1*w*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
wklwrite('matA.wk1', F, 1, 1)
M = wklread('matA.wk1');
M(m+2,n+2)=0;
ifadebir=E*t/(2*(1-v*v));
gerilme=ifadebir*M
surf(-gerilme)
colormap(hot);
axis([0 m 0 n -(50*xxx) 500000])
else
    if xx==0
        B=B;
        y = inv(A0) * B;
        C=(1:1:m*n);
        DD = reshape(C, m, n);

```

EK-1 (Devam) Program Kodu

```
F=y(DD');
wklwrite('matA.wk1', F, 1, 1)
M = wklread('matA.wk1');
M(m+2,n+2)=0;
ifadebir=E*t/(2*(1-v*v));
gerilme=ifadebir*M
    end
end

% hObject      handle to pushbutton9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

DARBE YÜKÜ ALTINDA ANALİZ ARAYÜZÜ

```
function varargout = darbealtindaanalizgerilmeenerjili(varargin)
% DARBEALTINDAANALIZGERILMEENERJILI M-file for
darbealtindaanalizgerilmeenerjili.fig
%     DARBEALTINDAANALIZGERILMEENERJILI, by itself, creates a new
DARBEALTINDAANALIZGERILMEENERJILI or raises the existing
%     singleton*.
%
%     H = DARBEALTINDAANALIZGERILMEENERJILI returns the handle to a
new DARBEALTINDAANALIZGERILMEENERJILI or the handle to
%     the existing singleton*.
%
%
DARBEALTINDAANALIZGERILMEENERJILI('CALLBACK',hObject,eventData,handl
es,...) calls the local
%     function named CALLBACK in
DARBEALTINDAANALIZGERILMEENERJILI.M with the given input arguments.
%
%     DARBEALTINDAANALIZGERILMEENERJILI('Property','Value',...)
creates a new DARBEALTINDAANALIZGERILMEENERJILI or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
darbealtindaanalizgerilmeenerjili_OpeningFunction gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
darbealtindaanalizgerilmeenerjili_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
darbealtindaanalizgerilmeenerjili
```

EK-1 (Devam) Program Kodu

```
% Last Modified by GUIDE v2.5 20-May-2008 13:54:22

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @darbealtindaanalizgerilmeenerjili_OpeningFcn, ...
                  'gui_OutputFcn',    @darbealtindaanalizgerilmeenerjili_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 darbealtindaanalizgerilmeenerjili is made
% visible.
function darbealtindaanalizgerilmeenerjili_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
darbealtindaanalizgerilmeenerjili (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout =
darbealtindaanalizgerilmeenerjili_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
```

EK-1 (Devam) Program Kodu

```
% 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;

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (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 edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1
%         as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (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 edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (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 edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2
%         as a double
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (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
```

EK-1 (Devam) Program Kodu

```

function edit3_Callback(hObject, eventdata, handles)
global mm
mm = str2double(get(hObject, 'string'));
% hObject    handle to edit3 (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 edit3 as text
%        str2double(get(hObject, 'String')) returns contents of edit3
as a double
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (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 edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (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 edit4 as text
%        str2double(get(hObject, 'String')) returns contents of edit4
as a double
% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (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 edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (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 edit5 as text

```

EK-1 (Devam) Program Kodu

```
%          str2double(get(hObject,'String')) returns contents of edit5
as a double
% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (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 edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (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 edit6 as text
%          str2double(get(hObject,'String')) returns contents of edit6
as a double
% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (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 edit7_Callback(hObject, eventdata, handles)
global gg
gg = str2double(get(hObject,'string'));
% hObject      handle to edit7 (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 edit7 as text
%          str2double(get(hObject,'String')) returns contents of edit7
as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
```

EK-1 (Devam) Program Kodu

```
% hObject      handle to edit7 (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 edit8_Callback(hObject, eventdata, handles)
global hh
hh = str2double(get(hObject,'string'));
% hObject      handle to edit8 (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 edit8 as text
%          str2double(get(hObject,'String')) returns contents of edit8
as a double
% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (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 edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (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 edit9 as text
%          str2double(get(hObject,'String')) returns contents of edit9
as a double
% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (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.
```

EK-1 (Devam) Program Kodu

```
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (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 edit10 as text
%         str2double(get(hObject,'String')) returns contents of
edit10 as a double
% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit10 (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 edit11_Callback(hObject, eventdata, handles)
% hObject      handle to edit11 (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 edit11 as text
%         str2double(get(hObject,'String')) returns contents of
edit11 as a double
% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit11 (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
```

EK-1 (Devam) Program Kodu

```

function edit12_Callback(hObject, eventdata, handles)
% hObject      handle to edit12 (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 edit12 as text
%         str2double(get(hObject,'String')) returns contents of
edit12 as a double
% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit12 (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 pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global q;
global E;
global c;
global m;
global t;
global v;
global n;
global w;
global xx;
global xxx;
global mm;
global hh;
global gg;
xx=( (m*n)+1)/2;
D = (E*t*t*t)/(12*(1-v*v));
xy = q*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

```

EK-1 (Devam) Program Kodu

```

x=xy/2;
A0 = eye (o,o)*20;
A0 (1:o*o-1:end)=22;
A0 (o*p+m:o*o:end)=22;
A0 (o*o-o*p-p:o*o:end)=22;
A0 (o+2:o+1:o*p)=21;
A0 (o*o-o*s-s:o+1:o*o-1)=21;
A0 (o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0 (o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0 (2:o+1:o*o)=-8;
A0 (o+1:o+1:o*o)=-8;
A0 (3:o+1:o*o-o)=1;
A0 (o*2+1:o+1:o*o)=1;
A0 (o*m+m:o*m+m:end)=0;
A0 (o*m+p:o*m+m:end)=0;
A0 (o*i+m:o*m+m:end)=0;
A0 (o*s+i:o*m+m:end)=0;
A0 (o*p+i:o*m+m:end)=0;
A0 (o*p+g:o*m+m:end)=0;

A0 (m+1:o+1:o*o-o*m)=-8;
A0 (m+2:o+1:o*o-o*m)=2;
A0 (o+m+1:o+1:o*o-o*m)=2;
A0 (o*m+2*m:o*m+m:end)=0;
A0 (o*p+2*m+1:o*m+m:end)=0;

A0 (o*m+1:o+1:o*o)=-8;
A0 (o*m+2:o+1:o*o)=2;
A0 (o*i+1:o+1:o*o)=2;
A0 (o*m*2+m:o*m+m:o*o)=0;
A0 (o*p*2+o+i:o*m+m:o*o)=0;

A0 (2*m+1:o+1:o*m*u)=1;

A0 (o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*w*c*c/D;
B0(xx-1,1)=4*w*c*c/D;
B0(xx-2,1)=1*w*c*c/D;
B0(xx+1,1)=4*w*c*c/D;
B0(xx+2,1)=1*w*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
J=max(F);

```

EK-1 (Devam) Program Kodu

```

K=max(J);
TU=c*m;
U=mm*gg*hh;
YTU=TU/2;
EI=(0.00560518*w*YTU*YTU)/K;
    Ppower=96*U*EI/(TU*TU*TU);
    PPP=nthroot(Ppower,2);
    D = (E*t*t*t)/(12*(1-v*v));
    x = PPP*c*c*c*c/D;
o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

A0 = eye(o,o)*20;
A0(1:o-1:end)=22;
A0(o*p+m:o*o:end)=22;
A0(o*o-o*p-p:o*o:end)=22;
A0(o+2:o+1:o*p)=21;
A0(o*o-o*s-s:o+1:o-1)=21;
A0(o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0(o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0(2:o+1:o*o)=-8;
A0(o+1:o+1:o*o)=-8;
A0(3:o+1:o*o-o)=1;
A0(o*2+1:o+1:o*o)=1;
A0(o*m+m:o*m+m:end)=0;
A0(o*m+p:o*m+m:end)=0;
A0(o*i+m:o*m+m:end)=0;
A0(o*s+i:o*m+m:end)=0;
A0(o*p+i:o*m+m:end)=0;
A0(o*p+g:o*m+m:end)=0;

A0(m+1:o+1:o*o-o*m)=-8;
A0(m+2:o+1:o*o-o*m)=2;
A0(o+m+1:o+1:o*o-o*m)=2;
A0(o*m+2*m:o*m+m:end)=0;
A0(o*p+2*m+1:o*m+m:end)=0;

A0(o*m+1:o+1:o*o)=-8;
A0(o*m+2:o+1:o*o)=2;
A0(o*i+1:o+1:o*o)=2;
A0(o*m*2+m:o*m+m:o*o)=0;
A0(o*p*2+o+i:o*m+m:o*o)=0;

```

EK-1 (Devam) Program Kodu

```

A0(2*m+1:o+1:o*m*u)=1;

A0(o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*PPP*c*c/D;
B0(xx-1,1)=4*PPP*c*c/D;
B0(xx-2,1)=1*PPP*c*c/D;
B0(xx+1,1)=4*PPP*c*c/D;
B0(xx+2,1)=1*PPP*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
wklwrite('matA.wkl', F, 1, 1)
M = wklread('matA.wkl');
M(m+2,n+2)=0
surf(-M)
colormap(hot);

axis([0 m 0 n -(50*xxx) 0.0000005])

% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global q;
global E;
global c;
global m;
global t;
global v;
global n;
global w;
global xx;
global xxx;
global mm;
global hh;
global gg;
xx=(m*n+1)/2;
D = (E*t*t*t)/(12*(1-v*v));
xy = q*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;

```

EK-1 (Devam) Program Kodu

```

u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

x=xy/2;
A0 = eye (o,o)*20;
A0 (1:o*o-1:end)=22;
A0 (o*p+m:o*o:end)=22;
A0 (o*o-o*p-p:o*o:end)=22;
A0 (o+2:o+1:o*p)=21;
A0 (o*o-o*s-s:o+1:o*o-1)=21;
A0 (o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0 (o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0 (2:o+1:o*o)=-8;
A0 (o+1:o+1:o*o)=-8;
A0 (3:o+1:o*o-o)=1;
A0 (o*2+1:o+1:o*o)=1;
A0 (o*m+m:o*m+m:end)=0;
A0 (o*m+p:o*m+m:end)=0;
A0 (o*i+m:o*m+m:end)=0;
A0 (o*s+i:o*m+m:end)=0;
A0 (o*p+i:o*m+m:end)=0;
A0 (o*p+g:o*m+m:end)=0;

A0 (m+1:o+1:o*o-o*m)=-8;
A0 (m+2:o+1:o*o-o*m)=2;
A0 (o+m+1:o+1:o*o-o*m)=2;
A0 (o*m+2*m:o*m+m:end)=0;
A0 (o*p+2*m+1:o*m+m:end)=0;

A0 (o*m+1:o+1:o*o)=-8;
A0 (o*m+2:o+1:o*o)=2;
A0 (o*i+1:o+1:o*o)=2;
A0 (o*m*2+m:o*m+m:o*o)=0;
A0 (o*p*2+o+i:o*m+m:o*o)=0;

A0 (2*m+1:o+1:o*m*u)=1;

A0 (o*2*m+1:o+1:o*o)=1;

B = eye (o,1);
B (1:1:o)=x;
B0 = eye (o,1);
B0 (1,1)=0;
B0 (xx,1)=16*w*c*c/D;
B0 (xx-1,1)=4*w*c*c/D;
B0 (xx-2,1)=1*w*c*c/D;

```

EK-1 (Devam) Program Kodu

```

B0 (xx+1,1)=4*w*c*c/D;
B0 (xx+2,1)=1*w*c*c/D;
B=B0;
xxx=B (xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y (DD');
J=max (F);
K=max (J);
TU=c*m;
U=mm*gg*hh;
YTU=TU/2;
EI=(0.00560518*w*YTU*YTU)/K;
    Ppower=96*U*EI/(TU*TU*TU);
    PPP=nthroot (Ppower,2);
    D = (E*t*t*t)/(12*(1-v*v));
    x = PPP*c*c*c*c/D;
o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

A0 = eye (o,o)*20;
A0 (1:o*o-1:end)=22;
A0 (o*p+m:o*o:end)=22;
A0 (o*o-o*p-p:o*o:end)=22;
A0 (o+2:o+1:o*p)=21;
A0 (o*o-o*s-s:o+1:o*o-1)=21;
A0 (o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0 (o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0 (2:o+1:o*o)=-8;
A0 (o+1:o+1:o*o)=-8;
A0 (3:o+1:o*o-o)=1;
A0 (o*2+1:o+1:o*o)=1;
A0 (o*m+m:o*m+m:end)=0;
A0 (o*m+p:o*m+m:end)=0;
A0 (o*i+m:o*m+m:end)=0;
A0 (o*s+i:o*m+m:end)=0;
A0 (o*p+i:o*m+m:end)=0;
A0 (o*p+g:o*m+m:end)=0;

A0 (m+1:o+1:o*o-o*m)=-8;
A0 (m+2:o+1:o*o-o*m)=2;
A0 (o+m+1:o+1:o*o-o*m)=2;

```

EK-1 (Devam) Program Kodu

```

A0(o*m+2*m:o*m+m:end)=0;
A0(o*p+2*m+1:o*m+m:end)=0;

A0(o*m+1:o+1:o*o)=-8;
A0(o*m+2:o+1:o*o)=2;
A0(o*i+1:o+1:o*o)=2;
A0(o*m*2+m:o*m+m:o*o)=0;
A0(o*p*2+o+i:o*m+m:o*o)=0;

A0(2*m+1:o+1:o*m*u)=1;

A0(o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*PPP*c*c/D;
B0(xx-1,1)=4*PPP*c*c/D;
B0(xx-2,1)=1*PPP*c*c/D;
B0(xx+1,1)=4*PPP*c*c/D;
B0(xx+2,1)=1*PPP*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
wklwrite('matA.wkl', F, 1, 1)
M = wklread('matA.wkl');
M(m+2,n+2)=0;
pi=3.14159265;
pi4=pi*pi*pi*pi;
Dpi4=D*pi4;
Dp=Dpi4/32;
uzz=m*c;
uz=n*c;
terimbir=(3*uzz)/(uz*uz*uz);
terimiki=(3*uz)/(uzz*uzz*uzz);
terimuc=2/(uz*uzz);
terimler=terimbir+terimiki+terimuc;
gk=Dp*terimler;
Gerilme_Enerjisi=M*gk
surf(-Gerilme_Enerjisi)
colormap(hot);
axis([0 m 0 n -(50*xxx) 50000])
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

EK-1 (Devam) Program Kodu

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
global q;
global E;
global c;
global m;
global t;
global v;
global n;
global w;
global xx;
global xxx;
global mm;
global hh;
global gg;
xx=( (m*n)+1)/2;
D = (E*t*t*t)/(12*(1-v*v));
xy = q*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;

x=xy/2;
A0 = eye (o,o)*20;
A0(1:o*o-1:end)=22;
A0(o*p+m:o*o:end)=22;
A0(o*o-o*p-p:o*o:end)=22;
A0(o+2:o+1:o*p)=21;
A0(o*o-o*s-s:o+1:o*o-1)=21;
A0(o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0(o*k-o+k:o*m+m:o*o-o*p-m)=21;

A0(2:o+1:o*o)=-8;
A0(o+1:o+1:o*o)=-8;
A0(3:o+1:o*o-o)=1;
A0(o*2+1:o+1:o*o)=1;
A0(o*m+m:o*m+m:end)=0;
A0(o*m+p:o*m+m:end)=0;
A0(o*i+m:o*m+m:end)=0;
A0(o*s+i:o*m+m:end)=0;
A0(o*p+i:o*m+m:end)=0;
A0(o*p+g:o*m+m:end)=0;
A0(m+1:o+1:o*o-o*m)=-8;
A0(m+2:o+1:o*o-o*m)=2;
```

EK-1 (Devam) Program Kodu

```

A0 (o+m+1:o+1:o*o-o*m)=2;
A0 (o*m+2*m:o*m+m:end)=0;
A0 (o*p+2*m+1:o*m+m:end)=0;

A0 (o*m+1:o+1:o*o)=-8;
A0 (o*m+2:o+1:o*o)=2;
A0 (o*i+1:o+1:o*o)=2;
A0 (o*m*2+m:o*m+m:o*o)=0;
A0 (o*p*2+o+i:o*m+m:o*o)=0;

A0 (2*m+1:o+1:o*m*u)=1;

A0 (o*2*m+1:o+1:o*o)=1;

    B = eye(o,1);
    B(1:1:o)=x;
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*w*c*c/D;
B0(xx-1,1)=4*w*c*c/D;
B0(xx-2,1)=1*w*c*c/D;
B0(xx+1,1)=4*w*c*c/D;
B0(xx+2,1)=1*w*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
J=max(F);
K=max(J);
TU=c*m;
U=mm*gg*hh;
YTU=TU/2;
EI=(0.00560518*w*YTU*YTU)/K;
Ppower=96*U*EI/(TU*TU*TU);
PPP=nthroot(Ppower,2);
D = (E*t*t*t)/(12*(1-v*v));
x = PPP*c*c*c*c/D;

o=n*m;
p=m-1;
r=n-1;
s=m-2;
u=n-2;
k=2*m;
l=2*n;
i=m+1;
j=n+1;
g=m+2;
h=n+2;
A0 = eye(o,o)*20;
A0(1:o*o-1:end)=22;
A0(o*p+m:o*o:end)=22;
A0(o*o-o*p-p:o*o:end)=22;

```

EK-1 (Devam) Program Kodu

```
A0(o+2:o+1:o*p)=21;
A0(o*o-o*s-s:o+1:o*o-1)=21;
A0(o*m+m+1:o*m+m:o*o-o*p-m)=21;
A0(o*k-o+k:o*m+m:o*o-o*p-m)=21;
```

```
A0(2:o+1:o*o)=-8;
A0(o+1:o+1:o*o)=-8;
A0(3:o+1:o*o-o)=1;
A0(o*2+1:o+1:o*o)=1;
A0(o*m+m:o*m+m:end)=0;
A0(o*m+p:o*m+m:end)=0;
A0(o*i+m:o*m+m:end)=0;
A0(o*s+i:o*m+m:end)=0;
A0(o*p+i:o*m+m:end)=0;
A0(o*p+g:o*m+m:end)=0;
```

```
A0(m+1:o+1:o*o-o*m)=-8;
A0(m+2:o+1:o*o-o*m)=2;
A0(o+m+1:o+1:o*o-o*m)=2;
A0(o*m+2*m:o*m+m:end)=0;
A0(o*p+2*m+1:o*m+m:end)=0;
```

```
A0(o*m+1:o+1:o*o)=-8;
A0(o*m+2:o+1:o*o)=2;
A0(o*i+1:o+1:o*o)=2;
A0(o*m*2+m:o*m+m:o*o)=0;
A0(o*p*2+o+i:o*m+m:o*o)=0;
```

```
A0(2*m+1:o+1:o*m*u)=1;
```

```
A0(o*2*m+1:o+1:o*o)=1;
```

```
B = eye(o,1);
B(1:1:o)=x;
B0 = eye(o,1);
B0(1,1)=0;
B0(xx,1)=16*PPP*c*c/D;
B0(xx-1,1)=4*PPP*c*c/D;
B0(xx-2,1)=1*PPP*c*c/D;
B0(xx+1,1)=4*PPP*c*c/D;
B0(xx+2,1)=1*PPP*c*c/D;
B=B0;
xxx=B(xx,1);
y = inv(A0) * B;
C=(1:1:m*n);
DD = reshape(C, m, n);
F=y(DD');
wklwrite('matA.wkl', F, 1, 1)
M = wklread('matA.wkl');
M(m+2,n+2)=0;
ifadebir=E*t/(2*(1-v*v));
gerilme=ifadebir*M
surf(-gerilme)
colormap(hot);
```

EK-1 (Devam) Program Kodu

```
axis([0 m 0 n -(50*xxx) 500000])  
% hObject      handle to pushbutton3 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KOÇ, Uğur
Uyruğu : T.C.
Doğum tarihi ve yeri : 29.06.1981 Trabzon
Medeni hali : Bekar
Telefon : 0 (312) 287 65 41
e-mail : ugurkoc@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	İstanbul Teknik Üniversitesi /İnşaat Müh.	2004
Lise	Ordu Fen Lisesi	1999

İş Deneyimi

Yıl	Yer	Görev
2006-	Sağlık Bakanlığı-İnşaat Onarım Dai.Bşk.	İnşaat Mühendisi
2004-2006	Toyran İnşaat	İnşaat Mühendisi

Yabancı Dil

İngilizce

Fransızca

Rusça

Hobiler

Futbol, yüzme, tenis