



ÇUKUROVA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

Süleyman ENEZ

**DÜZLEM ÇERÇEVE SİSTEMLERİN
MOD BİRLEŞTİRME YÖNTEMİ İLE DİNAMİK ANALİZİ**

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

ADANA, 2009

ÇUKUROVA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

**DÜZLEM ÇERÇEVE SİSTEMLERİN
MOD BİRLEŞTİRME YÖNTEMİ İLE DİNAMİK ANALİZİ**

Süleyman ENEZ

YÜKSEK LİSANS TEZİ

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez **06/10/2009** Tarihinde Aşağıdaki Jüri Üyeleri Tarafından **Oybirliği** İle Kabul Edilmiştir.

İmza.....

İmza.....

İmza.....

Doç.Dr.Hüseyin R.YERLİ
DANIŞMAN

Doç.Dr. H. Murat ARSLAN
ÜYE

Doç.Dr. S. Seren GÜVEN
ÜYE

Bu tez enstitümüz İnşaat Mühendisliği Anabilim Dalında hazırlanmıştır.

Kod No:

Prof.Dr. Aziz ERTUNÇ
Enstitü Müdürü
İmza ve Mühür

Bu Çalışma Çukurova Üniversitesi Bilimsel Araştırma Projeleri Birimi Tarafından Desteklenmiştir.

Proje No: MMF.2007.YL.28

● **Not:** Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZ

YÜKSEK LİSANS TEZİ

**DÜZLEM ÇERÇEVE SİSTEMLERİN
MOD BİRLEŞTİRME YÖNTEMİ İLE DİNAMİK ANALİZİ**

Süleyman ENEZ

ÇUKUROVA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

İNŞAAT MÜHENDİSLİĞİ ANA BİLİM DALI

Danışman: Doç.Dr. Hüseyin R.YERLİ

Yıl: 2009, **Sayfa:** 164

Jüri: Doç.Dr. Hüseyin R.YERLİ
Doç.Dr. H. Murat ARSLAN
Doç.Dr. S. Seren GÜVEN

Bu çalışmada, doğru eksenli elemanlardan oluşan ve düzlemi içinde yüklenmiş çerçevelerin dinamik yükler altındaki davranışları mod süperpozisyon yöntemi ele alınarak incelenmiştir.

Yapılarda dinamik davranış incelenirken, öncelikle ele alınan sistemin matematiksel modeli kurulmaktadır. Daha sonra ise matematiksel modeli kurulmuş olan sistemin serbest titreşim analizi yapılmaktadır. Serbest titreşim analizi tamamlandıktan sonra sistem, mod süperpozisyon metoduyla sistem denklem takımı girişimsiz hale getirilmekte ve çeşitli yöntemler yardımı ile bu sistemlerin zorlanmış titreşimi analizi yapılmaktadır.

Bu çalışmanın sonucunda, çerçeve sistemlerin serbest ve zorlanmış titreşim analizlerini yapan bilgisayar programları hazırlanmıştır. Bu programların hazırlanmasında MATHEMATICA bilgisayar paket programı kullanılmıştır.

Anahtar Kelimeler: Serbest titreşim, Zorlanmış titreşim, Dinamik davranış, Mod Süperpozisyon Yöntemi, Mathematica

ABSTRACT

MASTER THESIS

DYNAMIC ANALYSIS OF PLANAR FRAMES WITH MODE SUPERPOSITION TECHNIQUE

Süleyman ENEZ

DEPARTMENT OF CIVIL ENGINEERING

INSTITUTE OF NATURAL AND APPLIED SCIENCES

UNIVERSITY OF ÇUKUROVA

Supervisor: Assoc. Prof. Dr. Hüseyin R. YERLİ

Year: 2009 , **Pages:** 164

Jury: Assoc. Prof. Dr. Hüseyin R. YERLİ
Assoc. Prof. Dr. H. Murat ARSLAN
Assoc. Prof. Dr. S. Seren GÜVEN

In this study, dynamic behaviour of planar frames with members of straight axes is investigated with the aid of mode superposition technique.

During the investigation of dynamic behaviour of planar frames, firstly mathematical model of system is obtained. Then, free vibration analysis of this system is studied. After the free vibration analysis, uncoupled system equations are obtained with the aid of mode superposition technique and then forced vibration analysis of system is performed with the aid of several methods.

At the end of this study, general purpose computer programs, which analyse free end forced vibration behaviours of planar frames, are prepared. In the preparation of computer programs, the computer algebra system MATHEMATICA is used.

KeyWords: Free vibration, Forced vibration, Dynamic response, Mode Superposition Technique, Mathematica

TEŞEKKÜR

Bu tez çalışmasında, bilgi ve tecrübelerini benden esirgemeyen değerli hocam Sayın Doç.Dr. Hüseyin R. YERLİ'ye ve sağlamış olduğu yüksek lisans burs yardımından dolayı Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK)'na teşekkürü bir borç bilirim.

İÇİNDEKİLER

SAYFA

ÖZ.....	I
ABSTRACT.....	II
TEŞEKKÜR.....	III
İÇİNDEKİLER.....	IV
SİMGELER VE KISALTMALAR.....	VII
ÇİZELGELER DİZİNİ.....	IX
ŞEKİLLER DİZİNİ.....	X
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR.....	2
3. MATERYAL VE METOD.....	4
4. YAPI DİNAMİĞİ HAKKINDA GENEL BİLGİLER.....	5
4.1. Yapıların Dinamik Yükler Altındaki Davranışı.....	5
4.2. Matematiksel Modelin Kurulması.....	6
4.2.1. D'Alembert Metodu.....	6
4.2.2. Virtüel Deplasman Metodu.....	7
4.3. Dinamik Davranışın Formülasyonu.....	8
4.4. Sistem Rijitlik ve Kütle Matrislerinin Teşkili.....	10
5. SERBEST TİTREŞİM.....	13
5.1. Giriş.....	13
5.2. Özdeğer ve Özvektörlerin Bazı Özellikleri.....	14
5.2.1. Özvektörlerin Normalizasyonu.....	14
5.2.2. Rayleigh Oranı	15
5.2.3. Sturm Teoremi.....	15
5.2.4. Genel Özdeğer Probleminin Standart Hale Dönüştürülmesi..	16
5.2.5. Kaydırma (Shift).....	18
5.3. Özdeğer Problemlerinin Çözüm Yöntemleri.....	19
5.3.1. Kesin Çözüm.....	19
5.3.2. Yaklaşık Çözüm Yöntemleri.....	20
5.3.2.1. Vektör İterasyon Yöntemleri.....	21

5.3.2.1.(1). Ters İterasyon Yöntemi.....	21
5.3.2.1.(2). İleri İterasyon Yöntemi.....	23
5.3.2.1.(3). Rayleigh Oranı ile İterasyon.....	25
5.3.2.1.(4). Gram-Schmidt Ortogonalizasyonu.....	26
5.3.2.2. Transformasyon Yöntemleri.....	28
5.3.2.2.(1). Jacobi Metodu.....	29
5.3.2.2.(2). Genel Jacobi Metodu.....	32
5.3.2.2.(3). Householder-QR-Ters İterasyon Metodu.....	34
5.3.3. Büyük Sistemlerin Özdeğer Problemlerinin Çözümü.....	35
5.3.3.1. Determinant Arama Metodu.....	35
5.3.3.2. Alt Uzaylarla İterasyon Metodu.....	36
6. ZORLANMIŞ TİTREŞİM.....	44
6.1. Giriş.....	44
6.2. Sayısal Çözüm Yöntemleri.....	44
6.2.1. Direkt İntegrasyon Metodları.....	44
6.2.1.1. Merkezi Sonlu Farklar Metodu.....	45
6.2.1.2. Houbolt Metodu.....	45
6.2.1.3. Wilson θ Metodu.....	45
6.2.1.4. Newmark Metodu.....	45
6.2.2. Mod Süperpozisyon Metodu.....	48
7. FOURIER TRANSFORM METODU.....	62
7.1. Fourier Dönüşümü.....	62
7.2. Fourier Dönüşümün Bazı Özellikleri.....	63
7.3. Ayırık Fourier Dönüşümü.....	65
7.3.1. Ayırık Fourier Dönüşümün Önemi.....	66
7.3.2. Kompleks Fourier Serileri.....	66
7.4. Ayırık Fourier Dönüşüm Formülleri.....	69
7.5. Fourier ve Ters Fourier Dönüşümün Bulunması.....	71
7.5.1. Fourier Dönüşümü Bulmak İçin İzlenecek Yöntem.....	71
7.5.2. Ters Fourier Dönüşümü Bulmak İçin İzlenecek Yöntem.....	72
7.6. Fourier Dönüşümün Yapı Dinamiği Problemlerine Uygulanması.....	72

7.6.1. Tek Serbestlik Dereceli Sistemlerin Çözümü.....	73
7.6.2. Çok Serbestlik Dereceli Sistemlerin Çözümü.....	74
8. DEPREM ANALİZİ.....	78
8.1. Giriş.....	78
8.2. Yer Hareketi Durumunda Yapıların Davranışı.....	78
8.3. Spektrum Analizi.....	80
8.4. Çok Serbestlik Dereceli sistemlerin Deprem Analizi.....	84
9. SONUÇLAR VE ÖNERİLER.....	90
KAYNAKLAR.....	92
ÖZGEÇMİŞ.....	93
EKLER.....	94
EK-1. BİLGİSAYAR PROGRAMLARI.....	94
E1.1. Mathematica Programları.....	94
E1.2. Program İçerikleri ve Listeleri.....	94
E1.2.1. MSSN.MAT Programı.....	94
E1.2.2. MSRN.MAT Programı.....	105
E1.2.3. MSSFT.MAT Programı.....	114
E1.2.4. MSRFT.MAT Programı.....	124
E1.2.5. MSSSP.MAT Programı.....	133
E1.2.6. MSRSP.MAT Programı.....	141
EK-2. ÖRNEKLERİN DATA DOSYALARI.....	148
E2.1. Örnek 5.9.'un Data Dosyası.....	148
E2.2. Örnek 6.1.'in Data Dosyası.....	158
E2.3. Örnek 6.2.'nin Data Dosyası.....	160
E2.4. Örnek 6.3.'ün Data Dosyası.....	161
E2.5. Örnek 7.1.'in Data Dosyası.....	162
E2.6. Örnek 8.1.'in Data Dosyası.....	164

SİMGELER VE KISALTMALAR

$\underline{M}$	: Sistem kütle matrisi,
$\underline{C}$	: Sistem sönüm matrisi,
$\underline{K}$	: Sistem rijitlik matrisi,
$\underline{X}$	: Sistem deplasman vektörü,
$\underline{P}(t)$	: Sistem yük vektörü,
ω	: Sistemin sönümsüz haldeki serbest titreşim frekansı
ξ	: Sistemin sönüm oranı
δW	: Dış kuvvetlerin virtüel işi
δU	: İç kuvvetlerin virtüel işi
$\underline{F}$	: Dıştan etkiyen dinamik yük vektörü
$\underline{a}$	: İvme vektörü
E	: Elastisite modülü
I	: Atalet momenti
L	: Elemanın boyu
r	: Atalet yarıçapı
γ	: Birim boyun ağırlığı
$\underline{T}$	: Transformasyon matrisi
λ	: Özdeğer (serbest titreşim frekanslarının karesi)
$\underline{\phi}$	: Özvektör (mod şekil fonksiyonu)
$[\Lambda]$	: Spektral matris
$[\Phi]$	: Modal matris
S_v	: Hız spektrumu
ω_D	: Sistemin sönümlü serbest titreşim frekansı
(u)	: Toplam deplasman
(x_g)	: Yer hareketi

S_d	: Deplasman spektrumu
S_a	: İvme spektrumu
T	: Serbest titreşim periyodu
C_0	: Deprem bölge katsayısı
S_{ak}	: K'nıncı modun ivme spektrum değeri
T_k	: K'nıncı modun periyodu
T_o	: Zemin hakim periyodu
K	: Yapı tipi katsayısı
I	: Yapı önem katsayısı
g	: Yerçekimi ivmesi
$\underline{r}$	: Statik etki katsayıları vektörü

ÇİZELGELER DİZİNİ

SAYFA

Çizelge 5.1. Örnek 5.9.'ait serbest titreşim frekansları (dönme var).....	42
Çizelge 5.2. Örnek 5.9.'ait serbest titreşim frekansları (dönme yok).....	42
Çizelge 6.1. Örnek 6.1.'e ait serbest titreşim frekansları.....	53
Çizelge 6.2. Örnek 6.1.'e ait max moment ve deplasman değerleri.....	56
Çizelge 6.3. Örnek 6.2.'e ait serbest titreşim frekansları.....	56
Çizelge 6.4. Örnek 6.3.'e ait kesme kuvveti ve deplasman değerleri.....	61
Çizelge 7.1. Fourier dönüşüm ifadeleri.....	64
Çizelge 7.2. Örnek 7.1.'e ait değerler.....	77
Çizelge 8.1. Tipik deplasman spektrum eğrisi değerleri.....	83
Çizelge 8.2. Örnek 8.1.'e ait sayısal veriler.....	89

ŞEKİLLER DİZİNİ

SAYFA

Şekil 4.1. Statik ve Dinamik Durumun Karşılaştırılması.....	5
Şekil 4.2. Virtüel Deplasman Durumu.....	7
Şekil 4.3. Tek Serbestlik Dereceli Sistem.....	8
Şekil 4.4. Serbest Cisim Diyagramı.....	8
Şekil 4.5. Kiriş Eleman İçin Kod Numaraları.....	10
Şekil 5.1. Karakteristik Polinom Grafiği.....	36
Şekil 5.2. Örnek 4.9'a Ait Şekil.....	41
Şekil 5.3. Serbest Titreşim Mod Şekilleri.....	43
Şekil 6.1. Örnek 5.1'e Ait Şekil.....	52
Şekil 6.2. Serbest Titreşim Mod Şekilleri.....	54
Şekil 6.3. Deplasman ve Moment Diyagramları.....	55
Şekil 6.4. Örnek 5.2'ye Ait Şekil.....	57
Şekil 6.5. Deplasman Diyagramı.....	58
Şekil 6.6. Örnek 5.3'ün Basitleştirilmiş Modeli.....	59
Şekil 6.7. Örnek 5.3'e Ait Şekil.....	60
Şekil 7.1. Zaman Uzayında $F(t)$ fonksiyonu.....	62
Şekil 7.2. $F(t)$ 'nin Ayırık Tanımı.....	65
Şekil 7.3. Ayırık Fourier Dönüşümünün Simetri ve Antisimetri Özelliği.....	70
Şekil 7.4. Örnek 6.1'e Ait Şekil.....	76
Şekil 8.1. Yer Hareketi Bulunmasın Durumunda Yapıların Hareketi.....	78
Şekil 8.2. Deplasmanın Zamana Göre Değişimi.....	81
Şekil 8.3. Deprem Deplasman Spektrum Eğrileri.....	83
Şekil 8.4. TDY'nin Önerdiği Deprem İvme Spektrum Eğrisi.....	84

1. GİRİŞ

Bu çalışmada amaçlanan, yapıların dinamik yükler altındaki davranışlarının incelenmesi ve dinamik yükler altında yapı sistemlerinin mühendislik özelliklerinin (deplasman, ivme, kesit tesirleri vb.) tespit edilmesidir.

Bilindiği gibi, yapıların dinamik yükler altındaki davranışı ikinci mertebeden diferansiyel denklem takımı tarafından idare edilmektedir. Bu tezde, diferansiyel denklemlerin sayısal çözüm yöntemleri üzerine çalışılmaktadır. Mod süperpozisyon metodunun kullanılması temel amaç olduğundan dolayı yapının serbest titreşim analizinin yapılmasının gerekliliği ortaya çıkmaktadır. Bu sebepten dolayı serbest titreşim konusu, bu tezde önemli bir yer tutmaktadır.

Yine mod süper pozisyon metodu ile girişimsiz hale gelen diferansiyel denklem takımının çeşitli yöntemler (Newmark metodu, Fourier transform metodu ve spektrum analizi) ile çözülmesi ve karşılaştırılması amaçlanmaktadır.

Son olarak deprem gibi yer hareketi söz konusu olduğu zaman yapıların davranışının incelenmesi amaçlanmaktadır.

Yukarıda anlatılan amaçlara uygun olarak, sembolik işlem yapabilen MATHEMATICA paket programı kullanılarak eğitim amaçlı bilgisayar programlarının geliştirilmesi amaçlanmaktadır.

2. ÖNCEKİ ÇALIŞMALAR

Tez üzerinde yapılan önceki çalışmalardan bazıları kısaca aşağıda görülmektedir.

Serbest titreşim konusunda yapılmış olan çalışmaların bazıları şu şekildedir:

BAUER (1957), ilk olarak vektör iterasyon yöntemleri üzerinde çalışmıştır. Kuvvet metodları üzerinde de çalışmalar yaparak, çeşitli tipteki matrisler için bu metodların özelliklerini tespit etmeye çalışmıştır.

SCHWARZ (1968), vektör iterasyon yöntemlerinin yakınsama özellikleri üzerinde bir dizi çalışmalar yapmıştır.

RUTISHAUSER (1969), vektör iterasyon yöntemlerini, simetrik ve tam pozitif matrisler üzerinde uygulamıştır. Ayrıca iterasyon vektörlerinin ortogonal olma özelliğinin, Gram-Schmidt ortogonalizasyonu ve Cholesky çarpanlarına ayırma yöntemlerini kullanarak muhafaza edilmesi üzerinde çalışmalar yapmıştır.

BATHE ve WILSON (1970–1979), Rayleigh-Ritz analizi olarak da bilinen alt uzaylarla iterasyon yönteminin vektör iterasyon yöntemleri ile birlikte kullanılmasının iyi sonuçlar verdiğini işaret etmişlerdir. Bu yöntemin, büyük hacimli problemler için uygun olduğunu göstermiştir.

WILSON ve ITOH (1983), hesaplama esnasında, daha önceden bulunan özdeğerlere tekrar yakınsamayı önlemek için, iterasyon vektörlerine ortogonalizasyon uygulamışlardır. Ayrıca hızlı yakınsamayı sağlamak için shift özelliğini kullanmışlardır.

Zorlanmış titreşim konusunda önceden yapılan çalışmaların bazıları şu şekildedir:

CLOUGH (1975), keyfi olarak zamanla değişen yükler altında yapıların davranışını incelemiştir. Ayrıca tipik birkaç sistem ve yükleme için seri şeklinde kapalı çözümler elde etmiştir. Bununla beraber deprem mühendisliği konusunda, özellikle yapı-zemin etkileşimi konusunda çalışmalar yapmıştır.

BATHE ve WILSON (1976), yapıların dinamik yükler altındaki davranışlarını idare eden diferansiyel denklemlerin çözümleri üzerinde

çalışmışlardır. Çözüm metodu olarak direkt integrasyon yöntemleri ile mod süperpozisyon metodunu kullanmışlardır.

CRAIG (1981), yapıların çeşitli tipteki yüklemeler altındaki davranışlarını ve serbest titreşim analizi üzerinde çalışmalar yapmıştır. Sistemlerin Fourier uzayındaki çözümleri üzerinde de çalışmıştır.

3. MATERYAL VE METOD

Göz önüne alınan kiriş elemana ait malzemenin, homojen, lineer elastik ve izotop olduğu kabul edilmektedir. Ele alınan yapı sistemlerinin bu malzemeden meydana geldiği varsayılmaktadır.

Tezde ilk olarak sonlu elemanlar metodu kullanılarak, düzlemi içinde yüklü çubuk sistemler için elde edilen eleman rijitlik ve kütle matrisleri verilmektedir. Sonra kodlama tekniği vb. bir metodla sistem rijitlik ve kütle matrisleri oluşturulmaktadır. Daha sonra serbest titreşim analizine değinilmektedir. Bundan sonrada zorlanmış titreşim analizi yapılmaktadır. Tez çalışması esnasında yapılanların tezde sunuluş şekli sırasıyla aşağıda gösterilmektedir.

Dördüncü bölümde, sistemin davranışını idare eden diferansiyel denklemin bulunuşu anlatılmakta, eleman rijitlik ve kütle matrisleri verilmekte ve sonra ortak bir takımda sistem matrislerinin teşkilinden bahsedilmektedir.

Beşinci bölümde, yapı sistemlerinin serbest titreşimi anlatılmaktadır. Serbest titreşim analizi esnasında ortaya çıkan özdeğer problemi ve özellikleri hakkında özet bilgi verildikten sonra bu tip problemlerin sayısal olarak çözümünü yapan değişik türde ve özellikteki metodlardan bahsedilmekte ve konu ile ilgili örnekler verilmektedir.

Altıncı bölümde, yapıların dinamik yükler altındaki zorlanmış titreşim analizine değinilmektedir. Zorlanmış titreşim analizinde uygulanan çözüm metodlarından bahsedilmektedir. Bölümün sonunda ise hazırlanan bilgisayar programları ile çözülmüş sayısal uygulamalar verilmektedir.

Yedinci bölümde, Fourier transform metodu anlatılmaktadır. Ayrıca bu metodun yapı dinamiğine nasıl uygulanacağı gösterilmektedir.

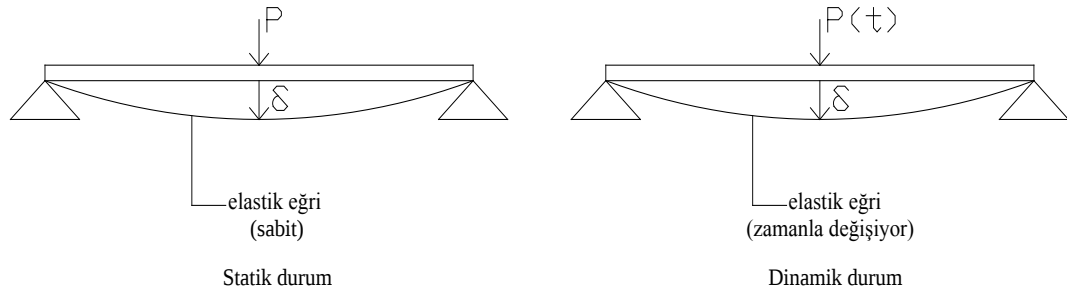
Sekizinci bölümde, deprem gibi yer hareketi durumunda yapıların davranışının mod süperpozisyon metodu ve spektrum analizi ile nasıl yapıldığı konusu anlatılmaktadır.

Ekler bölümünde, MATHEMATICA programları ile hazırlanan bilgisayar programları tanıtılmakta ve listeleri verilmektedir. Ayrıca tez içinde çözülmüş örneklere ait data dosyalarının listeleri de sunulmaktadır.

4. YAPI DİNAMİĞİ HAKKINDA GENEL BİLGİLER

4.1. Yapıların Dinamik Yükler Altındaki Davranışı

Fizikte olayın zamanla değişimi dinamik kelimesiyle ifade edilmektedir. Benzer bir statik problemine göre, bir yapı dinamiği problemi başlıca iki konuda farklılık göstermektedir. Bunlardan ilki, dinamik problemlerde yükün ve davranışın sabit olmadığı yani zamanla değişmesidir (Şekil 4.1.). İkinci değişiklik ise dinamik problemlerde, dinamik yer değiştirme esnasında sisteme atalet kuvvetlerinin de dahil olmasıdır. Bunların yanı sıra dinamik problemlerin çözümü, statik problemlerden farklı olarak tek bir çözümü olmayıp, zamana bağlı olarak bir çözüm takımından meydana gelmektedir. Yukarıda anlatılan sebeplerden ötürü, dinamik çözümün statik çözüme oranla daha zor olduğu sonucu ortaya çıkmaktadır, (Clough ve Penzien, 1993).



Şekil 4.1.- Statik ve dinamik durumun karşılaştırılması

Yapı dinamiğinin konusu içerik itibarı ile zamana bağlı olarak değişen yükler altında taşıyıcı sistemlerde meydana gelen gerilmelerin ve yer değiştirmelerin hesaplanmasıdır.

4.2. Matematiksel Modelin Kurulması

Sistemin davranışına ait matematiksel modelin kurulmasında, yapı dinamiğinde başlıca iki farklı metod kullanılmaktadır. Bunlar,

- a) D'Alembert metodu,
- b) Virtüel deplasman metodlarıdır.

4.2.1. D'Alembert Metodu

D'Alembert metodunda Newton'un ikinci hareket kanunu kullanılmakta ve düzlemsel halde aşağıda gösterildiği gibidir.

$$\underline{F} = m\underline{a} \quad (4.1)$$

$\underline{F} = (F_x, F_y, M)$: Dıştan etkiyen dinamik yük vektörü

$\underline{a} = (a_x, a_y, \alpha)$: İvme vektörü

$$a_x = \frac{d^2x}{dt^2}; a_y = \frac{d^2y}{dt^2}; a = \frac{d^2\theta}{dt^2} \quad (4.2)$$

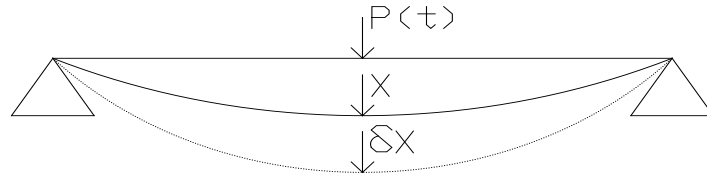
Yukarıdaki eşitlikte görülen $m\underline{a}$ atalet kuvvetini göstermektedir. D'Alembert prensibine göre, sistemin analizi sırasında atalet kuvvetlerinin göz önünde alınması şartı ile dinamik problem statik bir problemmiş gibi ele alınarak çözülebilmektedir. Açıklamak gerekirse;

$$\begin{aligned} \sum F_x + (-ma_x) &= 0 \\ \sum F_y + (-ma_y) &= 0 \\ \sum M + (-Ia) &= 0 \end{aligned} \quad (4.3)$$

Şeklinde yazıldığı takdirde, sistem statik bir problemmiş gibi çözülmektedir.

4.2.2. Virtüel Deplasman Metodu

Şekil 4.2. de görüldüğü gibi denge halinde bulunan sistemin x deplasmanına δx kadar virtüel (keyfi) bir deplasman verilmektedir.



Şekil 4.2. Virtüel deplasman durumu

$$x \rightarrow x + \delta x \quad (4.4)$$

Virtüel deplasman yönteminde dikkat edilmesi gereken husus, δx virtüel deplasmanı sistemin kinematik sınır şartlarını sağlamalıdır. Virtüel deplasman metoduna göre, virtüel deformasyon sırasında dış kuvvetlerin yaptığı virtüel iş, iç kuvvetlerin yaptığı virtüel işe eşittir.

$$\delta W = \delta U \quad (4.5)$$

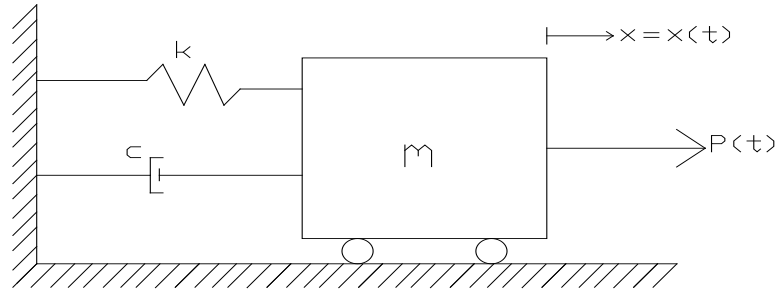
δW : Dış kuvvetlerin virtüel işi

δU : İç kuvvetlerin virtüel işi

Bu şekilde virtüel deplasman metodu kullanılarak sistemin matematiksel modeli kurulabilmektedir.

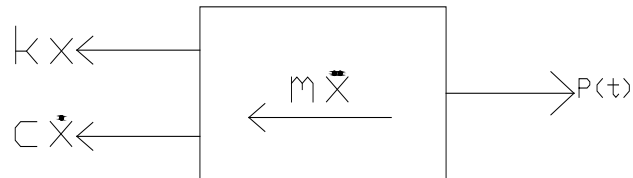
4.3. Dinamik Davranışın Formülasyonu

Burada tek serbestlik dereceli bir sistem üzerine D'Alembert metodu uygulanarak dinamik davranışın formülasyonu elde edilmektedir.



Şekil 4.3. Tek serbestlik dereceleri sistem

Sistemin deformasyon yapmış halde serbest cisim diyagramı aşağıda görülmektedir.



Şekil 4.4. Serbest cisim diyagramı

Şekil 4.4. teki sistem için D'Alembert metodu uygulanırsa sistem denklemi,

$$\sum F_x = 0 \Rightarrow P(t) - m\ddot{x} - c\dot{x} - kx = 0 \quad (4.6)$$

$$m\ddot{x} + c\dot{x} + kx = P(t) \quad (4.7)$$

olarak elde edilir. Tek serbestlik dereceli sistemler için aşağıdaki tarifler yapılmaktadır.

$$\frac{k}{m} = \omega^2; \frac{c}{m} = 2\xi\omega \quad (4.8)$$

ω : Sistemin sönümsüz haldeki serbest titreşim frekansı

ξ : Sistemin sönüm oranı

Yapı sistemlerinin çoğu için sönüm oranı $0.01 \leq \xi \leq 0.1$ arasında değer almaktadır. (4.8) ifadeleri (4.7) bağıntısında yerine yazılırsa $x = x(t)$ için olayı idare eden diferansiyel denklem aşağıdaki şekle gelmektedir:

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = \frac{P(t)}{m} \quad (4.9)$$

Sistem serbestlik derecesi birden fazla olan sistemler (sistemin hareketi sırasında meydana gelen atalet kuvvetlerini belirlemek için gerekli olan deplasman sayısı birden fazla olan sistemler), çok serbestlik dereceli sistemler olarak adlandırılır. Çok serbestlik dereceli sistemlerde, hareketi diferansiyel denklem takımı idare etmekte ve aşağıdaki gibi matris şeklinde ifade edilebilmektedir.

$$\underline{M} \ddot{\underline{X}} + \underline{C} \dot{\underline{X}} + \underline{K} \underline{X} = \underline{P}(t) \quad (4.10)$$

$\underline{M}$: Sistem kütle matrisi,

$\underline{C}$: Sistem sönüm matrisi,

$\underline{K}$: Sistem rijitlik matrisi,

$\underline{X}$: Sistem deplasman vektörü,

$\underline{P}(t)$: Sistem yük vektörü,

Sisteme dıştan etkiyen bir yük yok, $\underline{P}(t) = 0$, ise sistem serbest titreşim frekansı yapmaktadır. Sisteme dıştan etkiyen ve zamanla değişen bir yük var, $\underline{P}(t) \neq 0$, ise sistem zorlanmış titreşim hareketi yapmaktadır. İlerleyen konularda serbest ve zorlanmış titreşim olayları ayrıntılı olarak incelenecektir.

4.4. Sistem Rijitlik ve Kütle Matrislerinin Teşkili

Bu başlık altında düzlemi içinde yüklenmiş, kiriş elemanlardan oluşan düzlemsel sistemler için eleman rijitlik ve eleman kütle matrisleri verilmektedir.



Şekil 4.5. Kiriş eleman için kod numaraları

Şekil 4.5. deki, kendi düzlemi içinde yüklenmiş tipik bir kiriş eleman için, eleman koordinatlarında rijitlik ve kütle matrisleri aşağıda gösterilen şekillerdeki gibidir, (James ve Smith, 1989) :

Eleman Rijitlik Matrisi:

$$k = \frac{EI}{L^3} \begin{bmatrix} (L/r)^2 & 0 & 0 & -(L/r)^2 & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ -(L/r)^2 & 0 & 0 & (L/r)^2 & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} r = \sqrt{\frac{I}{A}} \quad (4.11)$$

Eleman Kütle Matrisi:

$$\underline{m} = \frac{\gamma L}{420} \begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2 \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2 \end{bmatrix} \quad (4.12)$$

Yukarıda geçen simgeleri tanımlamak gerekirse;

E: Elastisite modülü

I: Atalet momenti

L: Elemanın boyu

r: Atalet yarıçapı

γ : Birim boyun ağırlığı

Eleman koordinatlarında verilmiş olan bu matrislerin bütün sistem için ortak bir takıma göre yeniden düzenlenmesi gerekmektedir. Bunun için aşağıda gösterildiği şekilde bir transformasyon işlemi uygulanmaktadır.

$$\begin{aligned} \underline{k}' &= \underline{T}^T \underline{k} \underline{T} \\ \underline{m}' &= \underline{T}^T \underline{m} \underline{T} \end{aligned} \quad (4.13)$$

$\underline{T}$ matrisi transformasyon matrisi olup, düzlemi içinde yüklenmiş sistemler için aşağıda gösterildiği şekliyle verilmektedir.

$$\underline{T} = \begin{bmatrix} \underline{t} & \underline{0} \\ \underline{0} & \underline{t} \end{bmatrix} ; \quad \underline{t} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

Bu şekilde global takımında elde edilen eleman matrislerinin uygun bileşenleri kullanılarak, kodlama tekniği gibi bir yöntemle sistem rijitlik ve kütle matrisleri oluşturulmaktadır.

$$\begin{aligned} \sum \underline{k}_i' \rightarrow \underline{K} \\ \sum \underline{m}_i' \rightarrow \underline{M} \end{aligned} \quad (4.15)$$

5. SERBEST TİTREŞİM

5.1. Giriş

Serbest titreşimde, 4. bölümde bahsedilen (4.10) denkleminde $\underline{P}(t) = 0$ olması sonucunda sistem denklemleri aşağıda gösterildiği şekilde olmaktadır.

$$\underline{M} \ddot{\underline{X}} + \underline{C} \dot{\underline{X}} + \underline{K} \underline{X} = \underline{0} \quad (5.1)$$

Serbest titreşimde yapılan hesaplamalarda, sistemin sönüm özelliği ihmal edilmektedir ($\underline{C} = \underline{0}$). Bu durumda (5.1) eşitliği aşağıda gösterildiği şekilde olmaktadır.

$$\underline{M} \ddot{\underline{X}} + \underline{K} \underline{X} = \underline{0} \quad (5.2)$$

şekline gelmektedir. Ayrıca sistemin bilinmeyenleri için çözümün,

$$\underline{X} = \underline{\phi} \sin \omega t \quad (5.3)$$

şeklinde olduğu varsayıldığında, sistem ivme vektörü aşağıdaki şekle gelmektedir:

$$\ddot{\underline{X}} = -\omega^2 \underline{\phi} \sin \omega t \quad (5.4)$$

(5.3) ve (5.4) bağıntıları (5.2) eşitliğinde yerine yazılırsa, sistem denklemleri aşağıdaki gibi cebrik özdeğer problemine dönüşmektedir.

$$\begin{aligned} (\underline{K} - \omega^2 \underline{M}) \underline{\phi} &= \underline{0} \\ \underline{K} \underline{\phi} &= \lambda \underline{M} \underline{\phi} ; \lambda = \omega^2 \end{aligned} \quad (5.5)$$

λ : Özdeğer (serbest titreşim frekanslarının karesi)

$\underline{\phi}$: Özvektör (mod şekil fonksiyonu)

$$\underline{A} = \underline{K} - \lambda \underline{M} \quad (5.6)$$

Yukarıdaki gibi bir $\underline{A}$ matrisi tanımlanırsa, (5.5) denklemindeki özdeğer probleminin çözümü için aşağıda belirtilen şartlar ortaya çıkmaktadır.

$$\begin{aligned} |\underline{A}| \neq 0 &\Rightarrow \underline{\phi} = \underline{0} \\ |\underline{A}| = 0 &\Rightarrow \underline{\phi} \neq \underline{0} \end{aligned} \quad (5.7)$$

Yukarıdaki şartlardan da görülebileceği üzere, (5.5) denkleminin çözümünün olabilmesi için katsayılar determinatının sıfır olması gerekmektedir. Bundan dolayı problemlerde katsayılar matrisinin determinantını sıfır yapacak λ değerlerini bulmak gerekmektedir. Yani özdeğer problemi çözmek gerekmektedir.

5.2. Özdeğer ve Özvektörlerin Bazı Özellikleri

Bu başlık altında, özdeğer ve özvektörlerin yalnızca bu tezde kullanılan özellikleri hakkında bilgi verilmektedir.

5.2.1. Özvektörlerin Normalizasyonu

Özdeğer problemlerin çözümü sonucunda bulunan özvektörler, çeşitli amaçlar doğrultusunda değişik formlarda ifade edilmektedirler. Yapılan bu değiştirme işlemine özvektörlerin normalizasyonu denilmektedir. Tez çerçevesinde, özvektörlerin kütle matrisine göre normalizasyonu kullanılmaktadır. Kütle matrisine göre normalizasyona aynı zamanda kütle ve rijitlik matrislerine göre ortogonalite şartları da denilmekte ve aşağıdaki şekilde ifade edilmektedir.

$$\begin{aligned} [\phi_i]^T [M] [\phi_j] &= \delta_{ij} \\ [\phi_i]^T [K] [\phi_j] &= \lambda_i \delta_{ij} \end{aligned} \quad \begin{pmatrix} i = j & \delta_{ij} = 1 \\ i \neq j & \delta_{ij} = 0 \end{pmatrix} \quad (5.8)$$

5.2.2. Rayleigh Oranı

Rastgele seçilmiş bir $\underline{v}$ vektörü için (5.9) Rayleigh oranı denilmektedir. $\underline{v}$ vektörü sistemin herhangi bir özvektörüne eşitse ($\underline{v} = \underline{\phi}_i$), Rayleigh oranı özdeğere (λ_i) karşılık gelmektedir.

$$\rho(v) = \frac{[v]^T [K] [v]}{[v]^T [M] [v]}, \quad \lambda_1 \leq \rho(v) \leq \lambda_n \quad (5.9)$$

$$\rho(\phi_i) = \frac{[\phi_i]^T [K] [\phi_i]}{[\phi_i]^T [M] [\phi_i]} = \lambda_i \quad (5.10)$$

5.2.3. Sturm Teoremi

Genel bir özdeğer probleminde keyfi bir μ sayısı seçilecek olursa;

$$[\tilde{K}] = [K] - \mu [M] \quad (5.11)$$

yukarıdaki şekildeki gibi bir matris tanımlanacak olursa, tanımlanmış olan bu matris aşağıdaki gibi çarpanlarına ayrılabilir.

$$[\tilde{K}] = [L][D][L]^T \quad (5.12)$$

5.12 formülünde görülen $[D]$ diagoanal matris, $[L]$ ise alt üçgen matristir. $[D]$ matrisinin diagonalı üzerindeki negatif elemanların sayısı p ise, Sturm teoremine göre, bu özdeğer problemi için μ sayısından küçük p adet özdeğer mevcut anlamına gelmektedir, (Bathe, 1982).

5.2.4. Genel Özdeğer Probleminin Standart Hale Dönüştürülmesi

Kuvvetli çözüm yöntemlerinin geliştirilmiş olmasından dolayı, genel bir özdeğer problemi sistematik bir şekilde kolayca standart hale dönüştürülebilmektedir. Standart hale dönüştürme işlemi için başlıca iki farklı yol izlenebilmektedir.

Birinci tip dönüşümde $[M]$ kütle matrisinin tam pozitif olması hali için aşağıda gösterilmektedir. İlk olarak kütle matrisi,

$$[M] = [S][S]^T \quad (5.13)$$

şeklinde çarpanlara ayrılıp, aşağıdaki dönüşümler yapılırsa,

$$\begin{aligned} [\tilde{K}] &= [S]^{-1} [K] [S]^T \\ [\tilde{\phi}] &= [S]^T [\phi] \end{aligned} \quad (5.14)$$

şeklinde (5.5) genel özdeğer problemi,

$$[\tilde{K}][\tilde{\phi}] = \lambda [\tilde{\phi}] \quad (5.15)$$

şeklinde standart hale dönüştürülmektedir. İkinci tip dönüşümde ise $[K]$ rijitlik matrisinin tam pozitif olması hali için aşağıda gösterildiği şekilde yapılmaktadır. İlk olarak sistem rijitlik matrisi,

$$[K] = [S][S]^T \quad (5.16)$$

şeklinde çarpanlarına ayrıldıktan sonra,

$$\begin{aligned} [\tilde{M}] &= [S]^{-1} [M] [S]^{-T} \\ [\tilde{\phi}] &= [S]^T [\phi] \end{aligned} \quad (5.17)$$

Dönüşümleri yapılır ise, problem

$$[\tilde{M}][\tilde{\phi}] = \frac{1}{\lambda} [\tilde{\phi}] \quad (5.18)$$

şeklinde standart hale dönüşmektedir.

Örnek 5.1. Rijitlik ve kütle matrisleri aşağıda gösterilen genel özdeğer probleminin standart hale dönüştürülmesi:

$$\underline{K}\underline{\phi} = \lambda \underline{M}\underline{\phi} \quad , \quad \underline{K} = \begin{bmatrix} 5 & -4 & 1 & 0 \\ -4 & 6 & -4 & 1 \\ 1 & -4 & 6 & -4 \\ 0 & 1 & -4 & 5 \end{bmatrix} ; \quad \underline{M} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

İlk adım olarak kütle matrisi (5.13) teki gibi çarpanlarına ayrılacak olursa,

$$\underline{M} = \underline{S}\underline{S}^T \Rightarrow \underline{S} = \underline{S}^T = \begin{bmatrix} \sqrt{2} & & & \\ & \sqrt{2} & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \quad \underline{S}^{-1} = \underline{S}^{-T} = \begin{bmatrix} 1/\sqrt{2} & & & \\ & 1/\sqrt{2} & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

matrisleri elde edilir. Bu matrisler yardımıyla,

$$\tilde{K} = \underline{S}^{-1} \underline{K} \underline{S}^{-T} \Rightarrow \tilde{K} = \begin{bmatrix} 2.5 & -2 & 1/\sqrt{2} & 0 \\ -2 & 3 & -2.828 & 1/\sqrt{2} \\ 1/\sqrt{2} & -2.828 & 6 & -4 \\ 0 & 1/\sqrt{2} & -4 & 5 \end{bmatrix}$$

şeklinde yeni sistem matrisi elde edilir. Bu şekilde genel tip problem, (5.15) denklemindeki gibi standart hale dönüştürülmüş olur.

5.2.5. Kaydırma (Shift)

Genel bir özdeğer probleminde, μ keyfi bir sayı olmak üzere,

$$[\tilde{K}] = [K] - \mu[M] \quad (5.19)$$

Dönüşümü yapıldıktan sonra, (5.5) denkleminde yerine konulursa, özdeğer problemi

$$[\tilde{K}][\phi] = \eta[M][\phi] \quad (5.20)$$

halini almaktadır. Bu genel özdeğer probleminin çözümü ile aşağıda gösterildiği şekilde, ilk sistemin özdeğerleri (λ_i) bulunmaktadır.

$$\lambda_i = \eta_i + \mu \quad (5.21)$$

Yukarıdaki gibi bir yol izlemenin başlıca iki önemli nedeni bulunmaktadır. Bunlardan birincisi, sistem matrislerinden herhangi biri tam pozitif değil ise, çözüm yöntemlerinin birçoğu kullanım dışında kalmaktadır. Bu durumun görüldüğü sistemlerde shift yöntemi kullanılarak matrisler tam pozitif hale getirilebilmektedir. İkinci bir neden ise, iteratif yöntemlerle işlemleri hızlandırmasıdır. Seçmiş olduğumuz shift, özdeğere ne kadar yakın ise problem o derece yakınlıkta çabuk yakınsamaktadır.

5.3. Özdeğer Problemlerin Çözüm Yöntemleri

Genel özdeğer problemleri ve standart özdeğer problemler için çeşitli çözüm yöntemleri geliştirilmiştir. Geliştirilmiş olan çözüm yöntemlerini genel olarak iki gruba ayırmak mümkündür.

- a) Kesin çözüm,
- b) Yaklaşık çözüm.

5.3.1. Kesin Çözüm

Aşağıda gösterildiği şekilde bir özdeğer problemi,

$$(\underline{K} - \lambda \underline{M}) \phi = \underline{0} \quad (5.22)$$

gösterilen formülün tanımından hareketle, çözümün olabilmesi için katsayılar matrisinin determinantının sıfır olması gerekmektedir. (5.22) denkleminde katsayılar matrisinin determinantı alınırsa λ , ya bağlı olarak bir polinom elde edilmektedir. Elde edilen bu denkleme karakteristik polinom denilmektedir.

$$\begin{aligned} p(\lambda) &= \text{Det}(\underline{K} - \lambda \underline{M}) = 0 \\ p(\lambda) &= a_n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0 \\ \lambda_1 &< \lambda_2 < \dots < \lambda_n \end{aligned} \quad (5.23)$$

(5.23)' teki karakteristik polinom kökleri, özdeğerleri $(\lambda_1, \lambda_2, \dots, \lambda_n)$ vermektedir. Özdeğerler bulunduktan sonra her bir özdeğere karşılık gelen özvektörler aşağıda gösterilen (5.24) bağıntısı ile bulunmaktadır.

$$(\underline{K} - \lambda_i \underline{M}) \phi_i = \underline{0} \quad (5.24)$$

Karakteristik polinomdaki doğru bir şekilde hesaplanması ve sonrada polinom köklerinin hassas bir şekilde hesaplanması hususuna dikkat edilmelidir. Yüksek dereceli polinomların köklerinin hassas olarak belirlenmesi oldukça zor veya imkansızdır. Bundan dolayı özdeğer problemleri için, karakteristik polinomun köklerinin hesabına dayanmayan çeşitli çözüm yöntemleri geliştirilmiştir.

Örnek 5.2. Örnek 5.1.'deki sistemin kesin çözümünün bulunması:

$$p(\lambda) = \text{Det}(\underline{K} - \lambda \underline{M}) = \begin{vmatrix} 5-2\lambda & -4 & 1 & 0 \\ -4 & 6-2\lambda & -4 & 1 \\ 1 & -4 & 6-\lambda & -4 \\ 0 & 1 & -4 & 5-\lambda \end{vmatrix} \Rightarrow p(\lambda) = 4\lambda^4 - 66\lambda^3 + 276\lambda^2 - 285\lambda + 25$$

yukarıda gösterildiği şekilde sistemin karakteristik polinomu bulunur. Polinomun kökleri hesaplanırsa sistemin özdeğerleri bulunur.

$$\lambda_1 = 0.09654 \Rightarrow \omega_1 = 0.31071 \text{ rad / sn}$$

$$\lambda_2 = 1.39147 \Rightarrow \omega_2 = 1.17961 \text{ rad / sn}$$

$$\lambda_3 = 4.37355 \Rightarrow \omega_3 = 2.09130 \text{ rad / sn}$$

$$\lambda_4 = 10.6384 \Rightarrow \omega_4 = 3.26166 \text{ rad / sn}$$

Özdeğerlerde bulunduktan sonra (5.24) bağıntısı yardımı ile her bir özdeğere karşılık gelen özvektörler yani mod şekil fonksiyonları hesaplanmaktadır.

5.3.2. Yaklaşık Çözüm Yöntemleri

Yaklaşık çözüm yöntemlerini genel olarak kendi içinde iki farklı gruba ayırmak mümkündür:

- a) Vektör iterasyon yöntemleri,
- b) Transformasyon yöntemleri.

5.3.2.1. Vektör İterasyon Yöntemleri

Bu yöntemlerde bir dizi iterasyon sonunda özdeğerler ve özvektörler bulunmaktadır. Vektör iterasyon yöntemlerini üç grupta toplamak mümkündür.

- a) Ters iterasyon yöntemi,
- b) İleri iterasyon yöntemi,
- c) Rayleigh oranı ile iterasyon yöntemi.

Bu yöntemler aynı zamanda kuvvet metodları olarak ta adlandırılmaktadır.

5.3.2.1.(1). Ters İterasyon Yöntemi

Bu yöntemin kullanılabilmesi için sistem rijitlik matrisinin, $\underline{K}$, tam pozitif olması gerekmektedir. Eğer tam pozitif değil ise shift özelliğinden faydalanılarak bu yöntem kullanılabilir. Bunun yanı sıra kütle matrisi, $\underline{M}$, herhangi bir özel koşul aranmamaktadır. Bu yöntemin algoritması aşağıda gösterildiği şekildedir.

1- İlk olarak bir başlangıç vektörü seçilir. Başlangıç vektörü genellikle birim vektör seçilmektedir.

$$[X_1] = [1 \quad 1 \quad \dots \quad 1]^T$$

2- Daha sonra problem yakınsayana kadar aşağıdaki işlem döngüsü yapılmaktadır.

$$\left(\begin{array}{l} [K][\bar{X}_{k+1}] = [M][X_k] \\ \rho(\bar{X}_{k+1}) = \frac{[\bar{X}_{k+1}]^T [K] [\bar{X}_{k+1}]}{[\bar{X}_{k+1}]^T [M] [\bar{X}_{k+1}]} \\ [X_{k+1}] = \frac{[\bar{X}_{k+1}]}{([\bar{X}_{k+1}]^T [M] [\bar{X}_{k+1}])^{1/2}} \end{array} \right) \quad k = 1, 2, \dots, \infty \quad (5.25)$$

3- L adım sonunda problemin yakınsadığı kabul edilirse,

$$\rho(\bar{X}_{L+1}) \rightarrow \lambda_1 ; [X_{L+1}] \rightarrow [\phi_1] \quad (5.26)$$

şeklinde sistemin özdeğer ve özvektörlerinden biri (serbest titreşim frekansı ve mod şekil fonksiyonlarından biri) bulunmaktadır. (Bathe ve Wilson, 1976)

Ters iterasyon yönteminin en önemli özelliği, başlangıç vektörü ne seçilirse seçilsin daima en küçük özdeğer ve özvektör bulunmaktadır, (Bathe ve Wilson, 1976). Bu yöntemde (5.25) döngüsü aşağıdaki yakınsama kriteri sağlanıncaya kadar tekrarlanmaktadır.

$$\frac{|\lambda_1^{(k+1)} - \lambda_1^{(k)}|}{\lambda_1^{(k+1)}} \leq Tol \quad (5.27)$$

Yakınsama kriterindeki tolerans değeri 10^{-2s} olarak kullanılmaktadır (en küçük özdeğerinin 2s digit doğru olarak bulunması için).

Örnek 5.3. Ters iterasyon yöntemini kullanarak, örnek 5.1.'deki sistemin özdeğer hesabı:

Başlangıç vektörü $[X_1] = [1 \ 1 \ 1 \ 1]^T$ ve yakınsama toleransı 10^{-6} (s=3) olacak şekilde ters iterasyon algoritması uygulandığında, aşağıdaki değerler elde edilmektedir.

k=1

$$[\bar{X}_2] = [7.8 \ 12.2 \ 11.8 \ 7.2]^T \Rightarrow \rho(\bar{X}_2) = 0.0966516$$

k=2

$$[\bar{X}_3] = [3.239 \ 5.132 \ 4.962 \ 3.001]^T \Rightarrow \rho(\bar{X}_3) = 0.0965374$$

$$k=3$$

$$[\bar{X}_4] = [3.238 \quad 5.133 \quad 4.963 \quad 3.002]^T \Rightarrow \rho(\bar{X}_4) = 0.0965373$$

k=3 adım sonunda problem yakınsadığından iterasyona son verilerek aşağıdaki sonuçlar elde edilmektedir.

$$\lambda_1 = 0.09654 \quad [\phi_1] = \begin{bmatrix} 0.3126 \\ 0.4955 \\ 0.4791 \\ 0.2898 \end{bmatrix}$$

5.3.2.1.(2). İleri İterasyon Yöntemi

Bu yöntem, ters iterasyon yöntemi ile benzerlik göstermektedir. Ters iterasyon yönteminin aksine, bu yöntemle en büyük özdeğer ve buna karşılık gelen özvektör bulunmaktadır. (Bathe ve Wilson, 1976). Bu yöntemin kullanılabilmesi için kütle matrisinin, $\underline{M}$, tam pozitif olması gerekmektedir. Kütle matrisinin tam pozitif olmadığı durumlarda shift uygulanarak bu yöntem kullanılabilir. Yöntemin algoritması aşağıda gösterildiği şekildedir.

1- İlk olarak bir başlangıç vektörü seçilir. Başlangıç vektörü genellikle birim vektör seçilmektedir.

$$[X_1] = [1 \quad 1 \quad . \quad . \quad 1]^T$$

2- Daha sonra problem yakınsayana kadar aşağıdaki işlem döngüsü yapılmaktadır.

$$\left(\begin{array}{l} [M][\bar{X}_{k+1}] = [K][X_k] \\ \rho(\bar{X}_{k+1}) = \frac{[\bar{X}_{k+1}]^T [K][\bar{X}_{k+1}]}{[\bar{X}_{k+1}]^T [M][\bar{X}_{k+1}]} \\ [X_{k+1}] = \frac{[\bar{X}_{k+1}]}{\left([\bar{X}_{k+1}]^T [M][\bar{X}_{k+1}]\right)^{1/2}} \end{array} \right) \quad k = 1, 2, \dots, \infty \quad (5.28)$$

3- L adım sonunda problemin yakınsadığı kabul edilirse,

$$\rho(\bar{X}_{L+1}) \rightarrow \lambda_n ; [X_{L+1}] \rightarrow [\phi_n] \quad (5.29)$$

şeklinde sistemin en büyük özdeğer ve özvektörü bulunmaktadır. Bu yöntemde de iterasyonu sonlandırmak için, (5.27) yakınsama kriteri kullanılmaktadır.

Örnek 5.4. İleri iterasyon yöntemini kullanarak, örnek 5.1.'deki sistemin özdeğer hesabı:

Başlangıç vektörü $[X_1] = [1 \ 1 \ 1 \ 1]^T$ ve yakınsama toleransı 10^{-6} (s=3) olacak şekilde ileri iterasyon algoritması uygulandığında, aşağıdaki değerler elde edilmektedir.

k=1

$$[\bar{X}_2] = [1 \ -0.5 \ -1.0 \ 2]^T \Rightarrow \rho(\bar{X}_2) = 5.93333$$

k=2

$$[\bar{X}_3] = [1.096 \ -0.183 \ -4.017 \ 4.929]^T \Rightarrow \rho(\bar{X}_3) = 8.57887$$

$$k=9$$

$$[\bar{X}_{10}] = [-1.138 \quad 2.713 \quad -7.748 \quad 5.982]^T \Rightarrow \rho(\bar{X}_{10}) = 10.63844$$

$$k=10$$

$$[\bar{X}_{11}] = [-1.142 \quad 2.717 \quad -7.748 \quad 5.982]^T \Rightarrow \rho(\bar{X}_{11}) = 10.63845$$

k=10 adım sonunda problem yakınsadığından iterasyona son verilerek aşağıdaki sonuçlar elde edilmektedir

$$\lambda_4 = 10.63845 \quad [\phi_4] = \begin{bmatrix} -0.10731 \\ 0.25539 \\ -0.72827 \\ 0.56227 \end{bmatrix}$$

5.3.2.1.(3). Rayleigh Oranı İle İterasyon Yöntemi

Yapılan çalışmalar sonucu shift ile uygulanan ters iterasyon yönteminin daha çabuk yakınsadığı anlaşılmıştır. Seçilmiş olan shift değeri aranan özdeğere ne kadar yakın ise yöntem o kadar çabuk yakınsamaktadır. Buradaki temel zorluk uygun shift değerinin nasıl seçileceğidir. Uygun shift değerini seçmenin bir yolu, shift olarak Rayleigh oranının seçilmesidir. Ters iterasyon algoritması sırasında her adımda bulunan Rayleigh oranı bir sonraki adım için shift değeri olarak kullanılırsa yöntem daha çabuk yakınsamaktadır. Bundan dolayı Rayleigh oranının shift olarak seçilmesi ile uygulanan ters iterasyon yöntemine Rayleigh oranı ile iterasyon yöntemi denilmektedir. Ters iterasyon yöntemi için söylenenler, bu yöntem içinde geçerlidir (Bathe ve Wilson, 1976).

Yöntemin algoritması aşağıda gösterilmektedir:

- 1- İlk olarak bir başlangıç vektörü, $[X_1]$, ve başlangıç shift değeri seçilir. Genellikle başlangıçta shift değeri sıfır, $\rho(\overline{X}_1) = 0$, seçilmektedir.
- 2- Daha sonra problem yakınsayana kadar aşağıdaki işlem döngüsü yapılmaktadır.

$$\left(\begin{array}{l} [\tilde{K}][\overline{X}_{k+1}] = [M][X_k] \\ [\tilde{K}] = [K] - \rho(\overline{X}_k)[M] \\ \rho(\overline{X}_{k+1}) = \frac{[\overline{X}_{k+1}]^T [K][\overline{X}_{k+1}]}{[\overline{X}_{k+1}]^T [M][\overline{X}_{k+1}]} + \rho(\overline{X}_k) \\ [X_{k+1}] = \frac{[\overline{X}_{k+1}]}{([\overline{X}_{k+1}]^T [M][\overline{X}_{k+1}])^{1/2}} \end{array} \right) \quad k = 1, 2, \dots, \infty \quad (5.30)$$

- 3- L adım sonunda problemin yakınsadığı kabul edilirse,

$$\rho(\overline{X}_{L+1}) \rightarrow \lambda_i ; [X_{L+1}] \rightarrow [\phi_i] \quad (5.31)$$

şeklinde sistemin özdeğer ve özvektörlerinden biri bulunmaktadır. Bu yöntemde de iterasyonu sonlandırmak için, (5.27) yakınsama kriteri kullanılmaktadır.

5.3.2.1.(4). Gram-Schmidt Ortogonalizasyonu

Ters iterasyon yöntemi ile en küçük özdeğer, ileri iterasyon yöntemi ile ise en büyük özdeğer bulunmaktadır. Arada kalan özdeğerlerin bulunması için ise Rayleigh oranı ile iterasyon kullanılmaktadır. Rayleigh oranı ile iterasyon yöntemi kullanılırken başlangıçtaki shift değerini seçme zorluğu ortaya çıkmaktadır. Bu

nedenle, aradaki özdeğer ve özvektör çiftlerini bulmak için başka bir yöntem arama zorunluluğu ortaya çıkmaktadır. Alternatif yöntemlerden bir tanesi her seferinde başlangıç vektörünün değiştirilmesidir. Bu işlem için geliştirilen metodlardan biri Gram-Schmidt metodudur. Bu metotta yeni başlangıç vektörü, önceden bulunan özvektörlere dik olacak şekilde hesaplanmaktadır. Yeni vektörün hesaplanmasında daha önce bulunan özvektör kullanılmakta ve bu şekilde yeni bir özdeğer, özvektör çifti bulunmaktadır. Gram-Schmidt ortogonalizasyonu ile yeni başlangıç vektörünün hesabı için aşağıdaki bağıntılar kullanılmaktadır, (Bathe ve Wilson, 1976).

$$\alpha_i = [\phi_i]^T [M][X_1] \quad i = 1, 2, \dots, m \quad (5.32)$$

$$[\tilde{X}_1] = [X_1] - \sum_{i=1}^m \alpha_i [\phi_i] \quad (5.33)$$

Bu şekilde hesaplanan yeni başlangıç vektörü ve Rayleigh oranı ile iterasyon metodu kullanılarak aradaki özdeğer ve özvektör çiftleri bulunmaktadır.

Örnek 5.5. Gram-Schmit ortogonalizasyonu ve Rayleigh oranı ile itersayon yöntemini kullanarak, örnek 5.1.'deki sistemin özdeğer hesabı:

Ters iterasyon metodu ile örnek 5.3.'te en küçük, ileri iterasyon metodu ile örnek 5.4'te en büyük özdeğer ve özvektör bulunmuştu. Diğer bir özdeğer ve özvektörün hesabı için, Gram-Schmidt metodu ile yeni başlangıç vektörü aşağıdaki gibi bulunmaktadır.

$$\alpha_1 = [\phi_1]^T [M][X_1] \Rightarrow \alpha_1 = 2.3850$$

$$\alpha_2 = [\phi_4]^T [M][X_1] \Rightarrow \alpha_2 = 0.1299$$

$$[\tilde{X}_1] = [X_1] - \sum_{i=1}^2 \alpha_i [\phi_i] \Rightarrow [\tilde{X}_1] = \begin{bmatrix} 0.2683 \\ -0.2149 \\ -0.0481 \\ 0.2358 \end{bmatrix}$$

Elde edilen bu başlangıç vektörü ve başlangıç shift değeri $\rho(\bar{X}_1)=0$ kullanılarak Rayleigh oranı ile iterasyon algoritması uygulanırsa aşağıda görülen özdeğer, özvektör çifti bulunur.

$$\lambda_3 = 4.37355 \quad [\phi_3] = \begin{bmatrix} 0.43867 \\ -0.41674 \\ -0.02322 \\ 0.516965 \end{bmatrix}$$

5.3.2.2. Transformasyon Yöntemleri

Transformasyon yöntemlerinde temel amaç, sistem rijitlik ve kütle matrislerini diagonal forma getirmektir. Matrislerin diagonal forma gelmesi için sağdan ve soldan sırası ile $\underline{P}$ ve $\underline{P}^T$ gibi ortogonal matrislerle çarpma işlemi uygulanmaktadır. Bu işleme benzerlik dönüşümleri de denilmektedir. Bu yolla diagonal forma gelmiş olan matrislerden oluşan sistemin özdeğerleri ile orijinal haldeki sistemin özdeğerleri tamamen aynıdır. Böylece diagonal forma gelen sistemin özdeğerlerini bulmak oldukça kolaydır. $[M_1]=[M]$ ve $[K_1]=[K]$ olmak üzere, sembolik olarak

$$[K_{k+1}] = [P_k]^T [K_k] [P_k] \quad (5.34)$$

$$[M_{k+1}] = [P_k]^T [M_k] [P_k]$$

şeklindeki transformasyon işlemlerinden sonra $k \rightarrow \infty$ için $[K_{k+1}]$ ve $[M_{k+1}]$ matrislerinin diagonal forma geldiği kabul edilirse özdeğer ve özvektörler aşağıdaki gibi bulunmaktadır, (Bathe, 1982):

$$[\Lambda] = \text{Diag}(K_r^{(L+1)} / M_r^{(L+1)}) \quad (5.35)$$

$$[\Phi] = [P_1][P_2] \cdots [P_L] \text{Diag}(1/\sqrt{M_r^{(L+1)}})$$

$[\Lambda]$: Spektral matris

$[\Phi]$: Modal matris

Yukarıdaki denklemde görülen spektral matris, diagonal elemanları özdeğerler olan matris; modal matris ise her bir kolonu özvektörlere karşılık gelen matristir.

Transformasyon yöntemlerinin en önemli özelliklerinden biri, bu metodlar ile sonuçta tüm özdeğerlerin bulunmasıdır. Transformasyon yöntemleri genel olarak üç metottan meydana gelmektedir. Bunlar,

- a) Jacobi metodu,
- b) Genel Jacobi metodu,
- c) Hoselholder-QR-Ters iterasyon metodudur.

5.3.2.2.(1). Jacobi Metodu

Jacobi metodu, sadece standart tip özdeğer problemlerin çözümü için geliştirilmiş bir metottur. Genel bir özdeğer problemi jacobi metodu ile çözülmenden önce problem standart hale dönüştürülmelidir. Daha sonra jacobi yöntemi kullanılarak problem çözülmelidir, (Bathe, 1982).

$$[P_k] = \begin{bmatrix} 1 & & & \\ & \cdot & & \\ & & \cos \theta & -\sin \theta \\ & & \sin \theta & \cos \theta \\ & & & & \cdot \\ & & & & & 1 \end{bmatrix} \quad (5.36)$$

Bu metotta sistem matrislerini diagonal forma getirmek için ortogonal rotasyon matrisleri, $[P_k]$, kullanılmaktadır. (5.36) tipik bir rotasyon matrisine örnek olarak gösterilebilir. (5.36) ifadesinde görülen θ açısı sistem matrislerinin sıfırlanacak elemanlarına bağlı olarak (5.37) denkleminde bulunmaktadır. Bununla birlikte transformasyon matrisinde gösterilmeyen elemanların değerleri sıfırdır.

$$k_{ii}^{(k)} \neq k_{jj}^{(k)} \Rightarrow \tan 2\theta = \frac{2k_{ij}^{(k)}}{k_{ii}^{(k)} - k_{jj}^{(k)}} \quad (5.37)$$

$$k_{ii}^{(k)} = k_{jj}^{(k)} \Rightarrow \theta = \frac{\pi}{4}$$

Transformasyon işlemine son vermek için aşağıdaki yakınsama kriterlerinin birlikte sağlanması gerekmektedir.

$$\frac{|k_{ii}^{(k+1)} - k_{ii}^{(k)}|}{k_{ii}^{(k+1)}} \leq 10^{-2s} \quad i = 1, 2, \dots, n$$

$$\left[\frac{(k_{ij}^{(k+1)})^2}{k_{ii}^{(k+1)} k_{jj}^{(k+1)}} \right]^{1/2} \leq 10^{-2s} \quad i, j = 1, 2, \dots, n$$

$$(i < j)$$
(5.38)

L adım sonunda problemin yakınsadığı farz edilirse,

$$[K_{L+1}] = [\Lambda]$$

$$[\Phi] = [P_1][P_2] \dots [P_L]$$
(5.39)

olacak şekilde spektral ve modal matrisleri bulunmaktadır.

Örnek 5.6. Jacobi metodunu kullanarak, örnek 5.1.'deki sistemin özdeğer hesabı:

Jacobi metodu ile standart problemler çözüldüğünden dolayı genel problem önce standart hale dönüştürülmelidir. Örnek 5.1.'de problemin standart hali elde edilmişti, burada bu sistemin özdeğerleri hesaplanmaktadır. Örnek 5.1.'de elde edilen sistem matrisine $[K_1]$ deyip, yakınsama toleransı olarak 10^{-6} ($s = 3$) seçilirse, sistemin özdeğer ve özvektörleri aşağıdaki gibi bulunmaktadır.

1. tarama:

$i = 1, j = 1$ için rotasyon matrisi şu şekilde elde edilmektedir:

$$\tan 2\theta = \frac{2(-2)}{(2.5) - 3} = \frac{-4}{-0.5} = 8 \Rightarrow 2\theta = 82.874984 \Rightarrow \theta = 41.437492$$

$$\sin \theta = 0.662, \cos \theta = 0.7497$$

$$[P_1] = \begin{bmatrix} 0.7497 & -0.6618 & 0 & 0 \\ 0.6618 & 0.7497 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow [K_2] = [P_1]^T [K_1] [P_1] = \begin{bmatrix} 0.7344 & 0 & -1.342 & 0.468 \\ 0 & 4.765 & -2.588 & 0.530 \\ -1.342 & -2.588 & 6 & -4 \\ 0.468 & 0.53 & -4 & 5 \end{bmatrix}$$

Diagonal harici diğer elemanlar içinde benzer işlemler yapılmakta ve 4. taramanın ($k=4$) sonunda problem yakınsamaktadır. Sonuç olarak sistemin özdeğer ve özvektörleri aşağıdaki gibi bulunmaktadır:

$$[\Lambda] = \begin{bmatrix} 0.09653 & & & \\ & 1.39147 & & \\ & & 4.37355 & \\ & & & 10.6384 \end{bmatrix} \quad [\Phi] = \begin{bmatrix} 0.4421 & -0.6297 & 0.62037 & 0.15212 \\ 0.7007 & -0.1760 & -0.58936 & -0.36152 \\ 0.4791 & 0.4894 & -0.02322 & 0.72826 \\ 0.2898 & 0.5770 & 0.51696 & -0.56197 \end{bmatrix}$$

5.3.2.2.(2). Genel Jacobi Metodu

Genel jacobi metodu genel özdeğer problemlerin çözümü için geliştirilmiş bir yöntemdir. Bu yöntemde kullanılan transformasyon matrisi aşağıdaki gibi elde edilmektedir.

$$[P_k] = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \alpha \\ & & & \ddots & \\ & \gamma & & & 1 \\ & & & & & 1 \end{bmatrix} \quad (5.40)$$

(5.40) ifadesinde bulunan α ve γ sayıları,

$$\begin{aligned} \bar{k}_{ii}^{(k)} &= k_{ii}^{(k)} m_{ij}^{(k)} - m_{ii}^{(k)} k_{ij}^{(k)} \\ \bar{k}_{jj}^{(k)} &= k_{jj}^{(k)} m_{ij}^{(k)} - m_{jj}^{(k)} k_{ij}^{(k)} \\ \bar{k}^{(k)} &= k_{ii}^{(k)} m_{jj}^{(k)} - m_{ii}^{(k)} k_{jj}^{(k)} \end{aligned} \quad (5.41)$$

$$x = \bar{k}^{(k)} / 2 + \text{sign}(\bar{k}^{(k)}) \sqrt{\left(\bar{k}^{(k)} / 2\right)^2 + k_{ii}^{(k)} k_{jj}^{(k)}} \quad (5.42)$$

$$\alpha = \frac{\bar{k}_{jj}^{(k)}}{x} \quad ; \quad \gamma = -\frac{\bar{k}_{ii}^{(k)}}{x} \quad (5.43)$$

ifadelerinden elde edilmektedir, (Bathe, 1982).

Bu yöntemde iterasyona son vermek için aşağıdaki yakınsama kriterlerinin birlikte sağlanması gerekmektedir.

$$\frac{|\lambda_1^{(k+1)} - \lambda_1^{(k)}|}{\lambda_1^{(k+1)}} \leq 10^{-2s} \quad i = 1, 2, \dots, n \quad (5.44)$$

$$\lambda_i = k_{ii} / m_{ii}$$

$$\left[\frac{(k_{ij}^{(k+1)})^2}{k_{ii}^{(k+1)} k_{jj}^{(k+1)}} \right]^{1/2} \leq 10^{-2s} \quad \left[\frac{(m_{ij}^{(k+1)})^2}{m_{ii}^{(k+1)} m_{jj}^{(k+1)}} \right]^{1/2} \leq 10^{-2s} \quad i, j = 1, 2, \dots, n \quad (i < j) \quad (5.45)$$

Yakınsanma kriterleri sağlandıktan sonra iterasyona son verilip, diagonal hale gelen sistem matrisleri ve (5.35) bağıntısı yardımı ile sistemin özdeğer ve özvektörleri hesaplanmaktadır.

Örnek 5.7. Genel Jacobi metodu ile örnek 5.1.'deki sistemin özdeğer hesabı:

$[K_1] = [K]$ ve $[M_1] = [M]$ olmak üzere iterasyona başlanır ise;

1.tarama:

i=1, j=2 için transformasyon matrisi şöyle elde edilir:

$$\begin{aligned} \bar{k}_{11} &= 5 \times 0 - 2 \times (-4) = 8 & x &= -9.062 \\ \bar{k}_{22} &= 6 \times 0 - 2 \times (-4) = 8 \Rightarrow & \gamma &= 0.883 \\ \bar{k} &= 5 \times 2 - 6 \times 2 = -2 & \alpha &= -0.883 \end{aligned} \Rightarrow [P_1] = \begin{bmatrix} 1 & -0.883 & 0 & 0 \\ 0.883 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bundan sonra transformasyon işlemi yapılır ise yeni matrisler bulunur.

$$[K_2] = [P_1]^T [K_1] [P_1] \Rightarrow [K_2] = \begin{bmatrix} 2.61 & 0 & -2.53 & 0.883 \\ 0 & 16.96 & -4.883 & 1 \\ -2.53 & -4.883 & 6 & -4 \\ 0.883 & 1 & -4 & 5 \end{bmatrix}$$

$$[M_2] = [P_1]^T [M_1] [P_1] \Rightarrow [M_2] = \begin{bmatrix} 3.5594 & 0 & 0 & 0 \\ 0 & 3.5594 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Benzer işlemler tüm diagonal dışı elemanlar içinde yapılmaktadır. 5. tarama sonunda problem yakınsamaktadır. Bundan sonra (5.35) bağıntısından sistemin spektral ve modal matrisleri aşağıdaki gibi elde edilmektedir.

$$[\Lambda] = \begin{bmatrix} 0.09654 & & & \\ & 1.39147 & & \\ & & 4.37355 & \\ & & & 10.6384 \end{bmatrix} \quad [\Phi] = \begin{bmatrix} 0.3126 & -0.4453 & 0.4387 & 0.10756 \\ 0.4955 & -0.1244 & -0.4167 & -0.25563 \\ 0.4791 & 0.4894 & -0.0232 & 0.72826 \\ 0.2898 & 0.5770 & 0.5170 & -0.56197 \end{bmatrix}$$

5.3.2.2.(3). Houselholder-QR-Ters İterasyon Metodu

Diğer bir transformasyon metodu da Houselholder-QR-Ters iterasyon metodudur. Bu yöntemin en önemli özelliği, Jacobi metodu gibi sadece standart tip problemler için geliştirilmiş olmasıdır. Eğer genel bir problem için bu metot kullanılacak ise, problem önce standart hale dönüştürülmelidir.

Houselholder-QR-Ters iterasyon yöntemi üç çözüm aşamasından meydana gelmektedir, (Bathe, 1982):

- 1) İlk olarak Householder transformasyon metodu kullanılarak sistem matrisi üçlü diagonal forma indirgenmektedir.
- 2) Sonra indirgenmiş olan sistem matrisine QR metodu uygulanarak sistem matrisi diagonal forma getirilip, özdeğerleri bulunmaktadır.
- 3) Son işlem olarak ise ters iterasyon yöntemi kullanılarak sistemin özvektörleri bulunmaktadır.

5.3.3. Büyük Sistemlerin Özdeğer Problemlerinin Çözümü

Yapı mekaniğinde çoğunlukla hacimleri geniş olan problemlerle karşılaşmaktadır. Bu sebepten ötürü, şu ana kadar değinilmiş olan yöntemler bu tip problemlerin çözümünde yetersiz kalmakta veya uygun bir şekilde kullanılamamaktadır. Bu sebeplerden başka yapı mekaniği problemlerinde, tüm serbest titreşim frekansları ve modları gerekli olmayıp yalnızca en küçük birkaç modun hesaplanması çözüm için yeterli olmaktadır. Bundan dolayı, büyük sistemlerin hesaplarında kullanılacak kadar en küçük özdeğeri bulmak için çeşitli metodlar geliştirilmiştir. Bu metodlardan bazıları şunlardır:

- a) Determinant arama metodu,
- b) Alt uzaylarda iterasyon metodu.

5.3.3.1. Determinant Arama Metodu

Genel bir tip özdeğer problemi için çözüm kesin olarak,

$$p(\lambda) = \text{Det}(\underline{K} - \lambda \underline{M}) = 0 \quad (5.46)$$

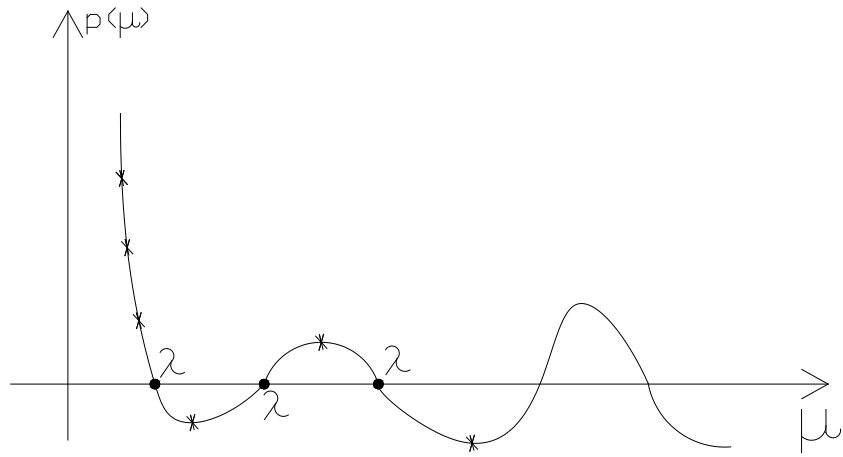
$$p(\lambda) = a_n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0$$

şeklinde bulunmaktadır. Keyfi bir μ sayısı için karakteristik polinomun determinanı, aşağıda görülen (5.47) bağıntısı ile bulunabilmektedir.

$$([K] - \mu[M]) = [L][D][L]^T \quad (5.47)$$

$$p(\mu) = \text{Det}([K] - \mu[M]) = \prod_{i=1}^n d_{ii}$$

Bu yöntemde, keyfi olarak seçilen μ sayıları için, karakteristik polinom değeri bulunup, eksenleri $p(\mu) - \mu$ olan koordinat takımına işaretlenmektedir (şekil 5.1.). Sonra determinantın işaret değiştirdiği yerde polinomun kökü bulunmaktadır. Bu şekilde istenen sayıda en küçük özdeğer bulunabilmekte ve istenildiği zaman Sturm teoremi ile (bkz. 5.2.3) bulunan köklerin aranılan en küçük özdeğer olup olmadığı kontrol edilebilmektedir. Son olarak ters iterasyon yöntemi ile özvektörler bulunmaktadır. (Bathe, 1982).



Şekil 5.1. Karakteristik polinomun grafiği

5.3.3.2. Alt Uzaylarla İterasyon Metodu

Alt uzaylarla iterasyon metodundaki temel amaç aşağıda görülmekte olan (5.48) ve (5.49) bağıntılarını sağlayan p adet en küçük özdeğerin hesaplanmasıdır.

$$[K]_{n \times n} [\Phi]_{n \times p} = [M]_{n \times n} [\Phi]_{n \times p} [\Lambda]_{p \times p} \quad (5.48)$$

$$[\Lambda]_{p \times p} = \text{Diag}(\lambda_i); [\Phi]_{n \times p} = [\Phi_1 \Phi_2 \dots \Phi_p]$$

$$[\Phi]_{pxn}^T [M]_{nxn} [\Phi]_{nxp} = [I]_{pxp} \quad (5.49)$$

$$[\Phi]_{pxn}^T [K]_{nxn} [\Phi]_{nxp} = [\Lambda]_{pxp}$$

Yukarıdaki (5.49) bağıntılarına aynı zamanda kütle ve rijitlik matrislerine göre ortogonalite şartları denilmektedir. Alt uzaylarla iterasyon metodunun algoritması aşağıda görüldüğü şekildedir, (Bathe, 1982).

1) İlk olarak başlangıç matrisi, $[X_1]_{nxp}$, seçilir.

2) Daha sonra problem yakınsayana kadar aşağıdaki işlem döngüsü yapılır.

$$E_k \rightarrow E_{k+1} \quad k=1, 2, \dots, \infty$$

$$[K] [\bar{X}_{k+1}] = [M] [X_k] \quad (5.50)$$

İlk olarak (5.50) denklemi çözülür ve $[K]$ ile $[M]$ nin E_{k+1} uzayındaki iz düşümleri bulunur.

$$[K_{k+1}]_{pxp} = [\bar{X}_{k+1}]^T [K] [\bar{X}_{k+1}] \quad (5.51)$$

$$[M_{k+1}]_{pxp} = [\bar{X}_{k+1}]^T [M] [\bar{X}_{k+1}]$$

E_{k+1} uzayında elde edilen matrislerden oluşan yeni sistemin (5.52) özdeğer problemi uygun bir metod ile (Genel Jacobi vb.) çözülür ve alt uzaydaki sistem için,

$$[K_{k+1}] [Q_{k+1}] = [M_{k+1}] [Q_{k+1}] [\Lambda_{k+1}] \quad (5.52)$$

$[Q_{k+1}]$ ile $[\Lambda_{k+1}]$ elde edilir. Bulunanlardan bir sonraki iterasyon adımı için yeni matris hesaplanır.

$$[X_{k+1}]_{n \times p} = [\bar{X}_{k+1}]_{n \times p} [Q_{k+1}]_{p \times p} \quad (5.53)$$

3) $k \rightarrow \infty$ için sistemin özdeğer ve özvektörleri şu şekilde hesaplanır:

$$[\Lambda_{k+1}] \rightarrow [\Lambda] \quad , \quad [X_{k+1}] \rightarrow [\Phi] \quad (5.54)$$

Alt uzaylarla iterasyon metodunda yakınsama kriteri olarak, vektör iterasyon yöntemlerinde bahsedilen ifade kullanılmaktadır.

$$\frac{|\lambda_1^{(k+1)} - \lambda_1^{(k)}|}{\lambda_1^{(k+1)}} \leq 10^{-2s} \quad i = 1, 2, \dots, p \quad (5.55)$$

Ayrıca çalışmalar sonucunda, p adet en küçük özdeğerin bulunması için q adet, $q = \min(2p, p+8)$, özdeğer bulmak gerekmektedir. Bu ifade, q adet özdeğerin en küçük p adedi, aynı zamanda sisteminde p adet en küçüğü anlamına gelmektedir. Böyle bir seçimle yapılan çözümün yaklaşık olarak on iterasyon adımından sonra yakınsayacağı kabul edilmektedir, (Bathe, 1982).

Örnek 5.8. Alt uzaylarla iterasyon yöntemini kullanarak, örnek 5.1.'deki sistemin özdeğer hesabı:

Bu örnekte en küçük iki değer ($p=2$) hesaplanmaktadır. İlk olarak bir başlangıç matrisi seçilmektedir. Başlangıç matrisi olarak, kolonları birim vektör olan 4×2 mertebeli matris kullanılmaktadır.

$$[X_1] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \Rightarrow [\bar{X}_2] = \begin{bmatrix} 2.4 & 3.2 \\ 3.2 & 5.2 \\ 2.8 & 4.8 \\ 1.6 & 2.8 \end{bmatrix}$$

(5.51) bağıntısından indirgenmiş sistem matrisleri elde edilmektedir.

$$[K_2] = \begin{bmatrix} 4.8 & 6.4 \\ 6.4 & 10.4 \end{bmatrix} \quad [M_2] = \begin{bmatrix} 42.4 & 66.56 \\ 66.56 & 105.44 \end{bmatrix}$$

Yukarıda görülen indirgenmiş matrislerden oluşan yeni sistem genel Jacobi metodu ile çözüldüğü zaman,

$$[\Lambda_2] = \begin{bmatrix} 2.2544 & 0 \\ 0 & 0.0983 \end{bmatrix} \quad [Q_2] = \begin{bmatrix} 1.61495 & -1.02063 \\ 0.01945 & 0.085099 \end{bmatrix}$$

matrisleri elde edilmektedir. Sonra (5.53) bağıntısı ile bir sonraki adım için yeni matris hesaplanarak iterasyona devam edilmektedir.

$$[X_2] = \begin{bmatrix} 0.6099 & 0.3190 \\ -0.1394 & 0.5048 \\ -0.3771 & 0.4629 \\ -0.2738 & 0.2694 \end{bmatrix}$$

Benzer şekilde işlemler yapılarak, k=7 adımdan sonra problem yakınsamakta ve sistemin en küçük iki özdeğeri ve özvektörü şu şekilde bulunmaktadır.

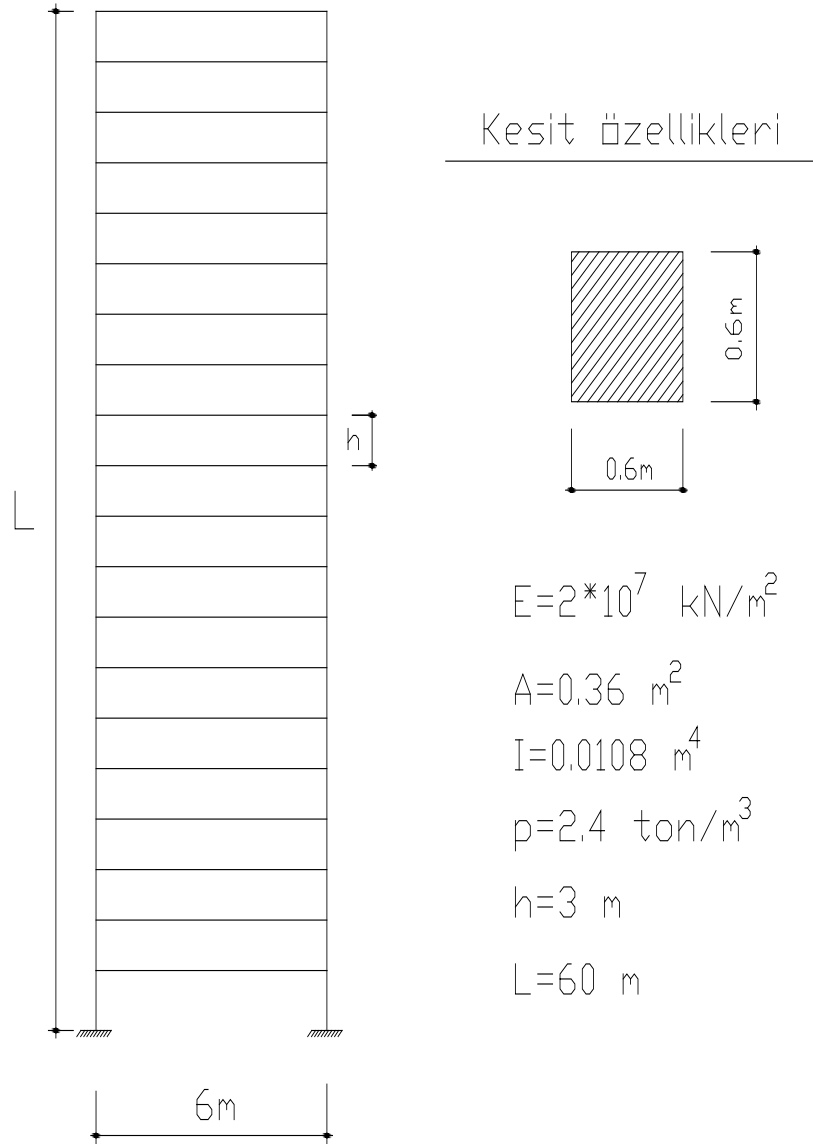
$$[\Lambda] = \begin{bmatrix} 1.39147 & \\ & 0.09654 \end{bmatrix} \quad [\Phi] = \begin{bmatrix} 0.4455 & 0.3126 \\ 0.1242 & 0.4955 \\ -0.4895 & 0.4791 \\ -0.5767 & 0.2898 \end{bmatrix}$$

Örnek 5.9. Şekil 5.2.'de gösterilen yapı sisteminin serbest titreşim frekanslarının ve mod şekillerinin bulunması:

Kesme tipi binalarda kolonların aksenal deformasyon yapmadığı ve kat hizalarında dönme olmadığı kabul edilirse, sistemin serbest titreşim frekansları ve mod şekilleri aşağıdaki gibi kapalı olarak ifade edilebilmektedir, (Clough ve Penzien, 1993).

$$\omega_n = \frac{2n-1}{2} \pi \sqrt{\frac{12 \sum EI}{m_i h L^2}} \quad n = 1, 2, \dots \quad (5.56)$$

$$\phi_n(x) = \sin\left(\frac{2n-1}{2} \frac{\pi x}{L}\right)$$



Şekil 5.2. Örnek 5.9.'ait şekil

Bu örnek (Zakout, 1992)'de değişik birkaç metod ile çözülmüş olup serbest titreşim frekansları ($f = \omega / 2\pi \text{ Hz}$) aşağıda gösterilmektedir (kat hizalarında dönme var) :

Çizelge 5.1. Örnek 5.9.'ait serbest titreşim frekansları (kat hizalarında dönme var) :

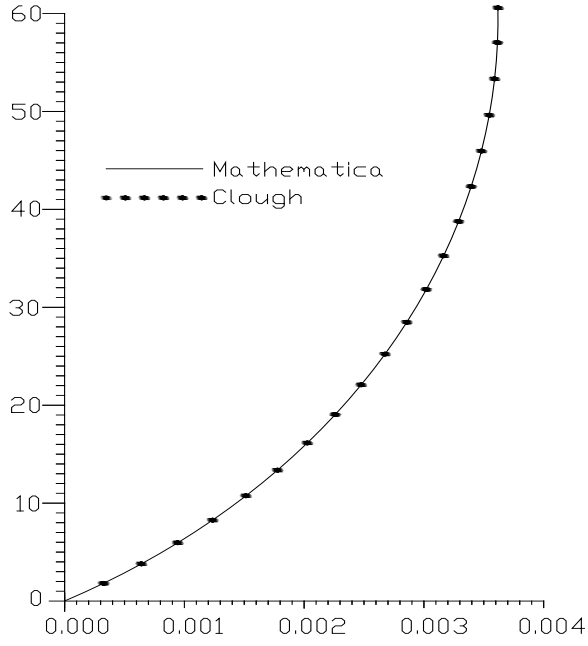
Determinant Arama	Alt Uzaylarla İterasyon	Lanczos Metodu	Mathematica MSSN.MAT	Denklem (5.56)
0.75932	0.75932	0.75932	0.77557	1.70103
2.29729	2.29729	2.29729	2.34723	5.10310
3.89175	3.89175	3.89192	3.97919	8.50517

Bu örnek kesme tipi bir bina olarak göz önüne alındığı zaman, serbest titreşim frekans değerleri aşağıdaki gibi elde edilmiştir (kat hizalarında dönme yok) :

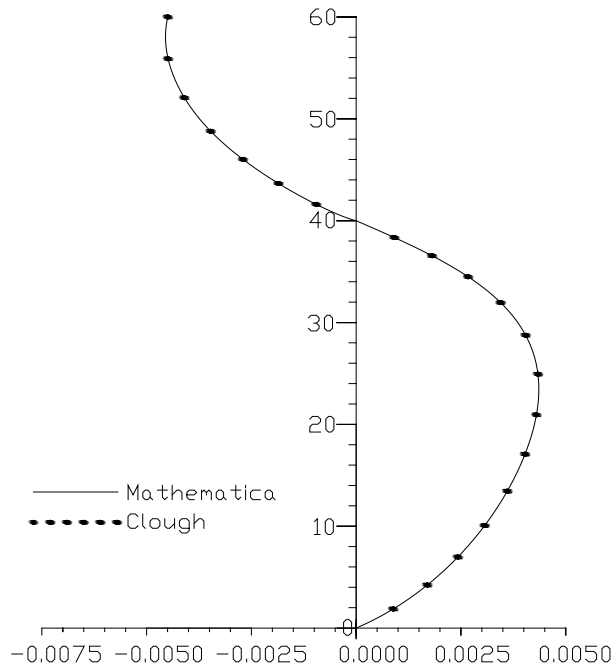
Çizelge 5.2. Örnek 5.9.'ait serbest titreşim frekansları (kat hizalarında dönme yok)

Determinant Arama	Alt Uzaylarla İterasyon	Lanczos Metodu	Mathematica MSSN.MAT	Denklem (5.56)
1.59661	1.59661	1.59661	1.67913	1.70103
4.78005	4.78005	4.78005	5.03726	5.10310
7.93421	7.93421	7.93451	8.38679	8.50517

Bunlara ek olarak ilk 2 serbest titreşim frekansına karşılık gelen modlar şekil 5.3.'te gösterilmektedir.



1. mod şekli



2. mod şekli

Şekil 5.3. Serbest titreşim mod şekilleri

6. ZORLANMIŞ TİTREŞİM

6.1. Giriş

Zorlanmış titreşim, sisteme dışardan dinamik yükler etkimesi haline karşılık gelmektedir. Dıştan etkiyen yükler, ya sistemin düğümlerine etkiyen zamanla değişen yük veya yer hareketi gibi durumlar nedeni ile dolaylı olarak etkiyen yüklerdir.

$$\underline{M} \ddot{\underline{X}} + \underline{C} \dot{\underline{X}} + \underline{K} \underline{X} = \underline{P}(t) \quad ; \quad \underline{P}(t) \neq 0 \quad (6.1)$$

Denklemden de görüldüğü üzere, zorlanmış titreşim olayında ikinci mertebeden diferansiyel denklem takımı çözümü ile uğraşmaktadır. Ancak taşıyıcı sistemlerin büyük olması durumunda diferansiyel denklem takımını çözmek daha zor bir hal almaktadır. Bu sebeple çeşitli sayısal çözüm yöntemleri geliştirilmiştir.

6.2. Sayısal Çözüm Yöntemleri

Sayısal çözüm yöntemlerini genel olarak iki başlık altında toplamak mümkündür.

- a) Direkt integrasyon metodları,
- b) Mod süperpozisyon metodu.

6.2.1. Direkt İntegrasyon Metodları

Bu yöntemin özelliği, işlemlerin adım adım sayısal olarak yapılmasıdır. Direkt integrasyon yöntemleri aşağıdaki gibi başlıca dört metottan meydana gelir:

- a) Merkezi sonlu farklar metodu,
- b) Houbolt metodu,
- c) Wilson θ metodu,
- d) Newmark metodu.

6.2.1.1. Merkezi Sonlu Farklar Metodu

Bu yöntemde, diferansiyel denklem takımı çözümü için sonlu farklar metodundan yararlanılmaktadır. Bu yöntemde birinci ve ikinci türevlerin merkezi farklar cinsinden ifade edilmesi ile elde edilen hız ve ivme bağıntıları belirli bir zaman aralığında ($0 \leq t \leq T$) seçilen uygun zaman artırımını Δt ile adım adım hesaplanarak, her t anında sistemin bilinmeyenleri hesaplanmaktadır, (Bathe, 1982).

6.2.1.2. Houbolt Metodu

Bu yöntem, merkezi farklar metodu ile benzerlik göstermekle beraber aralarındaki fark, Houbolt metodunda kullanılan hız ve ivme bağıntıları geri farklar ifadelerinden elde edilmektedir. Ayrıca Houbolt metodunda, sistem denkleminin t anında ve daha önceki değerleri bilinirken $t + \Delta t$ anında çözülerek sistemin bilinmeyenleri bulunmaktadır, (Bathe, 1982).

6.2.1.3. Wilson θ Metodu

Wilson θ metodu, lineer ivme metodunun biraz genişletilmiş halidir. Lineer ivme metodunda ivmenin t anı ile $t + \Delta t$ arasında lineer değiştiği kabul edilmektedir. Wilson θ metodunda ise ivmenin t anı ile $t + \theta \Delta t$ ($\theta \geq 1$) arasında lineer değiştiği kabul edilmektedir. Sistemin kararlı olması için $\theta \geq 1.37$ olarak seçilmekte olup, genellikle hesaplarda $\theta = 1.40$ olarak kullanılmaktadır, (Bathe, 1982).

6.2.1.4. Newmark Metodu

Newmark integrasyon metodu, lineer ivme ile ortalama sabit ivme kabullerine dayanmaktadır. Bu yöntemde deplasman ve hızların hesabı için aşağıdaki bağıntılar kullanılmaktadır, (Bathe, 1982).

$$\dot{X}_{t+\Delta t} = \dot{X}_t + \left[(1-\delta)\ddot{X}_t + \delta \ddot{X}_{t+\Delta t} \right] \Delta t \quad (6.2)$$

$$X_{t+\Delta t} = X_t + \dot{X}_t \Delta t + \left[(0.5-\alpha)\ddot{X}_t + \alpha \ddot{X}_{t+\Delta t} \right] \Delta t^2 \quad (6.3)$$

Denklemlerde görülen α ve δ parametreleri, lineer ivme ve ortalama sabit ivme kabullerine göre değerler almaktadır. $\delta = 1/2$ ve $\alpha = 1/6$ olduğu zaman (6.2) ve (6.3) denklemleri lineer ivme metodu bağıntılarına dönüşmektedir (bununla beraber Wilson θ metodunda $\theta=1$ durumu). Burada da sistem denklemi $t + \Delta t$ anı için yazılıp deplasmanlar adım adım bulunmaktadır. Herhangi bir $t + \Delta t$ anında sistem denklemi,

$$\underline{M} \ddot{X}_{t+\Delta t} + \underline{C} \dot{X}_{t+\Delta t} + \underline{K} X_{t+\Delta t} = \underline{P}(t)_{t+\Delta t} \quad (6.4)$$

şeklindedir. (6.3) denkleminde $t + \Delta t$ anındaki ivme çekilip (6.2) de yerine yazılırsa, $t + \Delta t$ anındaki ivme ve hızın deplasman cinsinden ifadeleri bulunmaktadır. Bulunanlar (6.4) denkleminde yerine konulup, denklem çözülerek deplasmanlar elde edilmektedir. Daha sonra (6.2) ve (6.3) ifadeleri yardımı ile sistemin hız ve ivme değerleri bulunur. Bütün bu işlemler Δt artımlar ile tüm zaman aralığında yapılarak sistemin her t anındaki çözümü hesaplanmaktadır.

Newmark metodunun çözüm algoritması aşağıda gösterildiği şekildedir:

- 1) Önce başlangıç değerleri (deplasman, hız ve ivme için) tespit edilir.
- 2) Sonra zaman artımı Δt seçilir ve α ve δ parametreleri hesaplanır.

$$\delta \geq 0.50 \quad ; \quad \alpha \geq 0.25(0.5 + \delta)^2 \quad (6.5)$$

3) Sonra integrasyon sabitleri hesaplanır:

$$\alpha_0 = \frac{1}{\alpha \Delta t^2} ; \alpha_1 = \frac{\delta}{\alpha \Delta t} ; \alpha_2 = \frac{1}{\alpha \Delta t} ; \alpha_3 = \frac{1}{2\alpha} - 1$$

$$\alpha_4 = \frac{\delta}{\alpha} - 1 ; \alpha_5 = \frac{\Delta t}{2} \left(\frac{\delta}{\alpha} - 2 \right) ; \alpha_6 = \Delta t (1 - \delta) ; \alpha_7 = \alpha \Delta t$$
(6.6)

4) Efektif rijitlik matrisi oluşturulur.

$$\tilde{\underline{K}} = \underline{K} + \alpha_0 \underline{M} + \alpha_1 \underline{C}$$
(6.7)

5) Bu adımdan sonra her zaman aralığı için aşağıdaki işlemler yapılır.

5.1) $t + \Delta t$ zamanındaki efektif yük hesaplanır:

$$\tilde{\underline{P}}_{t+\Delta t} = \underline{P}_{t+\Delta t} + \underline{M} \left(\alpha_0 \underline{\ddot{X}}_t + \alpha_2 \underline{\dot{X}}_t + \alpha_3 \underline{\ddot{X}}_t \right) + \underline{C} \left(\alpha_1 \underline{\dot{X}}_t + \alpha_4 \underline{\dot{X}}_t + \alpha_5 \underline{\ddot{X}}_t \right)$$
(6.8)

5.2) $t + \Delta t$ zamanı için sistem denkleminde deplasmanlar bulunur:

$$\tilde{\underline{K}} \underline{X}_{t+\Delta t} = \tilde{\underline{P}}_{t+\Delta t}$$
(6.9)

5.3) $t + \Delta t$ zamanı için ivme ve hız değerleri hesaplanır:

$$\underline{\ddot{X}}_{t+\Delta t} = \alpha_0 (\underline{X}_{t+\Delta t} - \underline{X}_t) - \alpha_2 \underline{\dot{X}}_t - \alpha_3 \underline{\ddot{X}}_t$$

$$\underline{\dot{X}}_{t+\Delta t} = \underline{\dot{X}}_t + \alpha_6 \underline{\ddot{X}}_t + \alpha_7 \underline{\ddot{X}}_{t+\Delta t}$$
(6.10)

Her t anı için bu işlemler tekrarlanarak sistem bilinmeyenleri bulunmaktadır.

6.2.2. Mod Süperpozisyon Metodu

Direkt integrasyon metotlarında çözüm için uygulanacak adım sayısı, tamamı ile analizde kullanılan çözüm aralığı ve zaman artımı ile doğru orantılıdır. Bu sebepten dolayı çözüm aralığı büyük seçilen sistemlerin analizi için direkt integrasyon metotlarını kullanmak çözümü uzatıp, zorlaştırmaktadır. Bu sebepten dolayı, büyük sistemler ve büyük zaman aralığında çözüm yapmak için daha efektif başka bir yöntemin kullanılması zorunlu hale gelmektedir. Böyle durumlarda, çözüm için daha efektif olan mod süperpozisyon yöntemi kullanılmaktadır.

Başlangıç şartlarının,

$$\underline{X}(0) = \underline{X}_0 \quad ; \quad \dot{\underline{X}}(0) = \underline{V}_0 \quad (6.11)$$

olduğu varsayılırsa, bu metodun amacı yine (6.1) ifadesinde görülen genel hareket denkleminin çözümünün, $\underline{X} = \underline{X}(t)$, bulunmasıdır. Sistem deplasmanları, mod süperpozisyon yöntemi ile aşağıdaki çözüm adımları izlenerek bulunmaktadır:

Adım-1) İlk olarak, beşinci bölümde anlatıldığı gibi,

$$(\underline{K} - \lambda \underline{M}) \phi = \underline{0} \quad ; \quad \lambda = \omega^2 \quad (6.12)$$

özdeğer problemi çözülerek, sistemin serbest titreşim frekansları (ω_1) ile mod şekil fonksiyonları ($\underline{\phi}_i$) hesaplanıp, bunlardan modal, $[\Phi]$, ve spektral, $[\Lambda]$, matrisler elde edilmektedir.

$$[\Phi] = [\underline{\phi}_1 \quad \underline{\phi}_2 \quad \dots \quad \underline{\phi}_n] \quad (6.13)$$

$$[\Lambda] = \text{Diag}(\omega_i^2)$$

Burada dikkat edilmesi gereken önemli hususlardan biride mod şekil fonksiyonlarının kütleye göre normalize edilmiş olarak hesaplanmasıdır. Böylece sistem denklemleri ileride görüleceği gibi daha basit bir forma dönüşmektedir.

Adım-2) Sistem deplasman vektörü için çözümün, $\underline{Y}$ modal deplasman vektörü olmak üzere,

$$\underline{X} = \underline{\Phi} \underline{Y} \quad (6.14)$$

şeklinde olduğu varsayılırsa, sistem denklemi,

$$\underline{M} \underline{\Phi} \ddot{\underline{Y}} + \underline{C} \underline{\Phi} \dot{\underline{Y}} + \underline{K} \underline{\Phi} \underline{Y} = \underline{P} \quad (6.15)$$

haline gelmektedir. Bu denklemin her iki tarafı soldan $[\underline{\Phi}]^T$ ile çarpılırsa,

$$\underline{\Phi}^T \underline{M} \underline{\Phi} \ddot{\underline{Y}} + \underline{\Phi}^T \underline{C} \underline{\Phi} \dot{\underline{Y}} + \underline{\Phi}^T \underline{K} \underline{\Phi} \underline{Y} = \underline{\Phi}^T \underline{P} \quad (6.16)$$

şeklini almaktadır. (6.16) sistem denkleminde görülen ifadeler beşinci bölümde anlatıldığı üzere aşağıdaki şekilde elde edilmektedir:

$$\begin{aligned} \underline{\tilde{M}} &= \underline{\Phi}^T \underline{M} \underline{\Phi} = [\underline{I}] \\ \underline{\tilde{K}} &= \underline{\Phi}^T \underline{K} \underline{\Phi} = [\underline{\Lambda}] \\ \underline{\tilde{C}} &= \underline{\Phi}^T \underline{C} \underline{\Phi} = \text{Diag}(2\xi\omega_i) \\ \underline{\tilde{P}} &= \underline{\Phi}^T \underline{P} \end{aligned} \quad (6.17)$$

Yukarıdaki ifadeler (6.16) denkleminde yerine yazılırsa, sistem denklem takımı,

$$\underline{\tilde{M}} \ddot{\underline{Y}} + \underline{\tilde{C}} \dot{\underline{Y}} + \underline{\tilde{K}} \underline{Y} = \underline{\tilde{P}} \quad (6.18)$$

şeklinde elde edilir. Sistem denklem takımı açık bir biçimde yazılacak olursa, aşağıdaki şekli alır.

$$\ddot{Y}_I + 2\xi\omega_i\dot{Y}_I + \omega_i^2 Y_I = P_I \quad I = 1, 2, \dots, n \quad (6.19)$$

Başlangıçta sistem serbestlik derecesi kadar birbiri ile girişimli diferansiyel denklem takımı varken, (6.14) denklemindeki dönüşüm sayesinde sistem serbestlik derecesi kadar girişimsiz diferansiyel denklem takımı elde edilmektedir. Bundan sonra bu girişimsiz denklem takımı herhangi bir metotla (Newmark metodu, Fourier transform metodu, Laplace dönüşümü vb.) çözülerek modal deplasmanlar bulunmaktadır.

Adım-3) Modal deplasmanlar bulunduktan sonra, (6.14) denkleminde sistemin esas deplasmanlarına geçilmektedir.

$$\underline{X} = \underline{\Phi} \underline{Y} = \begin{bmatrix} \underline{\phi}_1 & \underline{\phi}_2 & \dots & \underline{\phi}_n \end{bmatrix} \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \quad (6.20)$$

Bu ifadeyi açacak olursak aşağıdaki şekliyle yazılabilmektedir;

$$\underline{X} = \underline{\phi}_1 Y_1 + \underline{\phi}_2 Y_2 + \dots + \underline{\phi}_n Y_n \quad (6.21)$$

(6.21) bağıntısındaki birinci terim birinci modun katkısını, ikinci terim ise ikinci modun katkısını vs. göstermektedir.

Adım-4) Sistem deplasmanları bulunduktan sonra, sistemin eleman kuvvetleri veya kesit tesirleri hesaplanabilmektedir.

Bugüne kadar yapılan araştırmalarda, mod süperpozisyon metodu ile hesap yapılırken sistemin deplasmanlarına en büyük katkının en küçük modlardan geldiği

gözenmiştir. (Genellikle en büyük katkı ilk moddan gelmektedir). Bu nedenle hasaplarda, tüm modların bulunması yerine, sadece en küçük **p** adet modu kullanarak deplasmanların bulunması yeterli olmaktadır. Bu özellik sayesinde problemler daha az işlemle ve yeteri kadar hassas olarak daha kısa zamanda çözülebilmektedir.

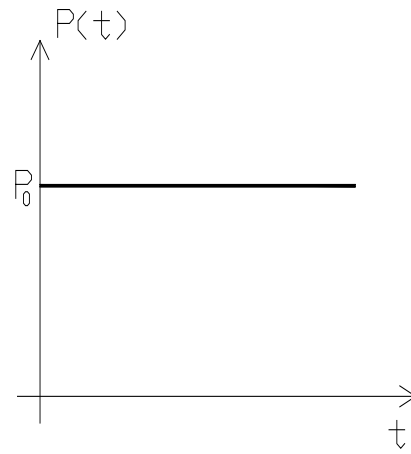
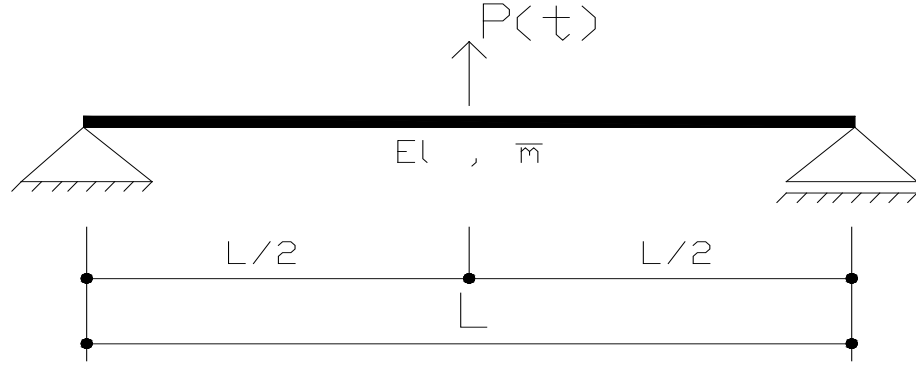
Bu tez çalışmasında, Rahleigh oranı ve alt uzaylarda iterasyon yöntemleri ile mod süperpozisyon metodu kullanılarak (6.1) denklemini girişimsiz hale getirilerek, Newmark ve Fourier transform metodları ile çözüm yapan programlar Mathematica programı ile hazırlanmıştır. Ek bölümünde bu programlar ile ilgili detaylı veriler sunulmaktadır.

Örnek 6.1. Şekil 6.1.'de görülen adım tipi yükle zorlanan iki ucu basit mesnetli kiriş sisteminin mod süperpozisyon yöntemi ile orta noktadaki deplasmanın ve momentinin zamanla değişiminin bulunması:

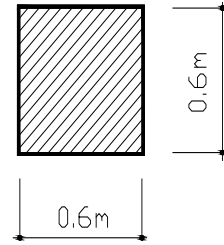
Bu örnek (Clough ve Penzien, 1993)'de çözülmüş olup, sistemin çözümü olarak aşağıdaki kapalı ifadeler verilmektedir.

$$\omega_n = n^2 \pi^2 \sqrt{\frac{EI}{mL^4}} \quad n = 1, 2, \dots \quad (6.22)$$

$$\phi_n(x) = \sin\left(\frac{n\pi x}{L}\right)$$



Kesit özellikleri



$$\begin{aligned}
 E &= 2 \cdot 10^7 \text{ kN/m}^2 & I &= 0.0108 \text{ m}^4 \\
 A &= 0.36 \text{ m}^2 & p &= 2.4 \text{ ton/m}^3 \\
 L &= 10 \text{ m} & P_0 &= 10 \text{ ton}
 \end{aligned}$$

Şekil 6.1. Örnek 6.1.'e ait şekil

Sistem deplasmanının zamanla değişimi,

$$v(x, t) = \frac{2P_0 L^3}{\pi^4 EI} \sum_{n=1}^{\infty} \frac{\alpha_n}{n^4} (1 - \cos \omega_n t) \sin\left(\frac{n\pi x}{L}\right) \quad (6.23)$$

şeklindedir. Kirişin herhangi bir noktasındaki moment değerinin zamanla değişimi de aşağıdaki bağıntı ile verilmektedir.

$$M(x, t) = -\frac{2P_0 L}{\pi^2} \sum_{n=1}^{\infty} \frac{\alpha_n}{n^2} (1 - \cos \omega_n t) \sin\left(\frac{n\pi x}{L}\right) \quad (6.24)$$

Yukarıdaki denklemlerde görülen α_n sayısı aşağıdaki gibidir :

$$\alpha_n = \begin{pmatrix} 1 & n = 1, 5, 9, \dots \\ -1 & n = 3, 7, 11, \dots \\ 0 & n = 2, 4, 6, \dots \end{pmatrix}$$

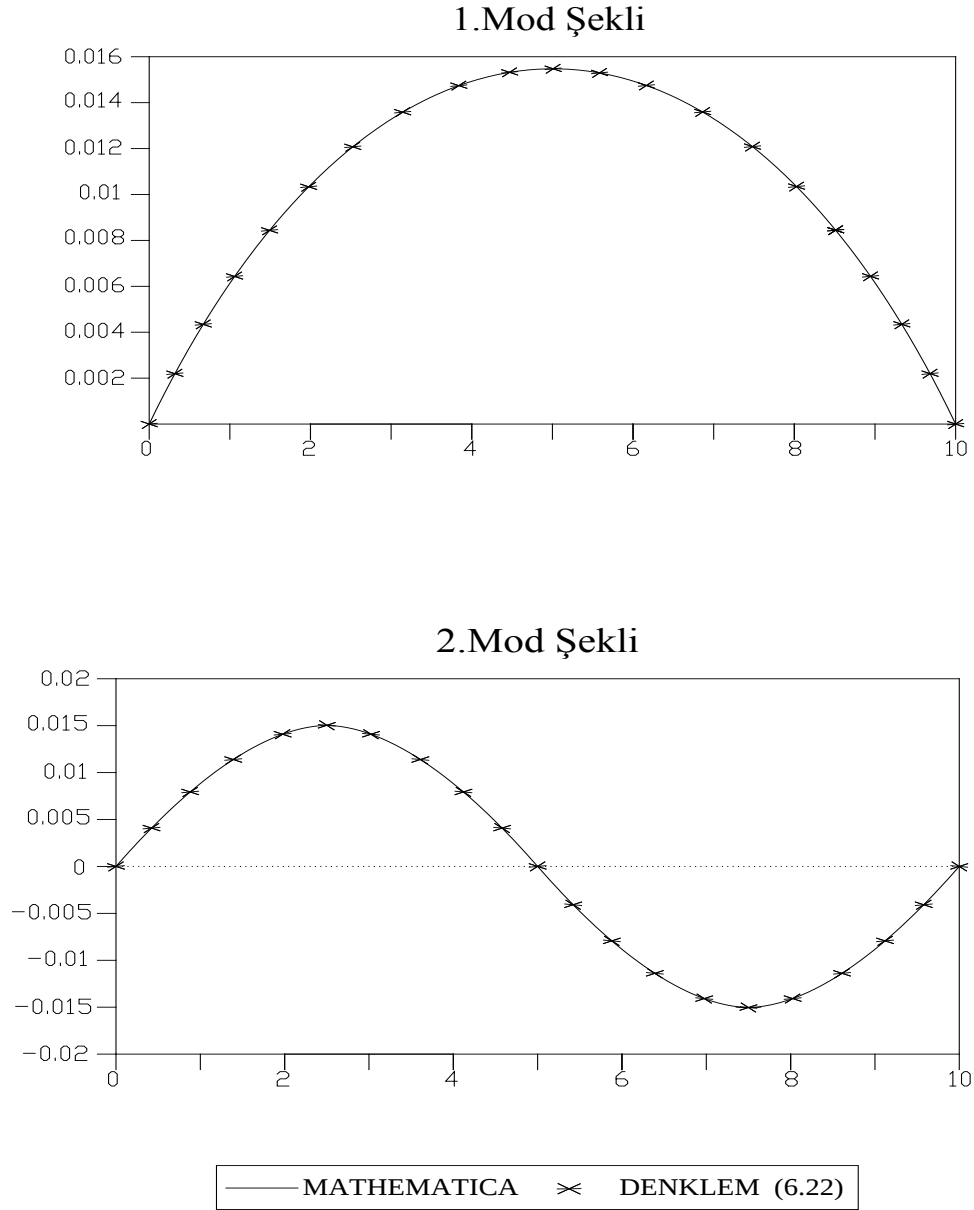
Aynı örnek (Zakout, 1992)'de çözülmüş olup, üç çözüm içinde elde edilmiş olan serbest titreşim frekansları ($f = \omega / 2\pi$ devir / sn) tabloda gösterilmiştir.

Çizelge 6.1. Örnek 6.1.'e ait serbest titreşim frekansları

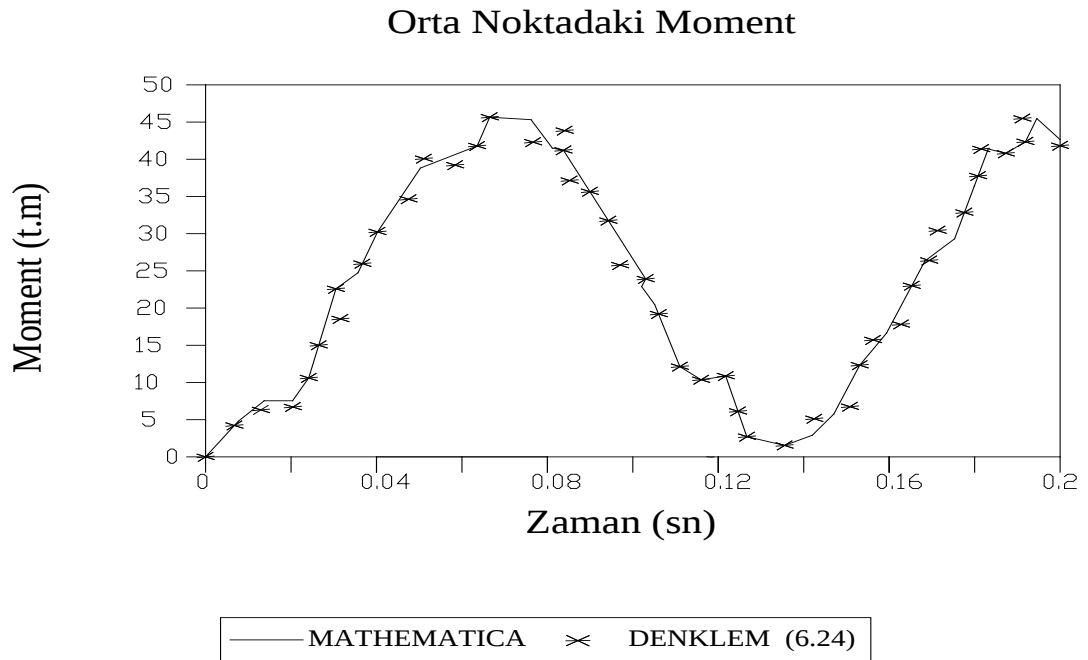
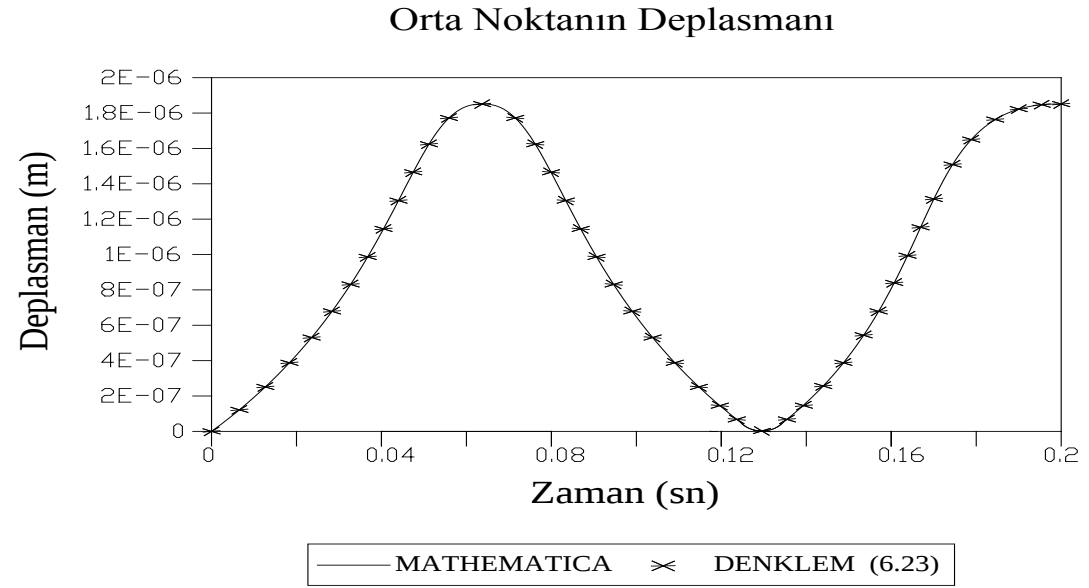
Lanczos M.	Mathematica MSSN.MAT	Denklem (6.22)
7.82309	7.85403	7.85398
30.9272	31.4193	31.41593
68.2561	70.7236	70.68583

Tablodan da görüldüğü üzere Mathematica ile yapılan çözümle bulunan değerler denklem (6.22) ile elde edilen değerlere oldukça yakındır.

Sistemin bu frekanslarına karşılık gelen modları şekil 6.2.'de gösterilmiştir.



Şekil 6.2. Serbest titreşim mod şekilleri



Şekil 6.3. Deplasman ve moment diyagramları

Sistemin orta noktasının maksimum deplasmanı ve moment değerleri aşağıdaki gibi elde edilmektedir.

Çizelge 6.2. Örnek 6.1.'e ait max moment ve deplasman değerleri

	Mathematica MSSN.MAT	Denklem (6.23), (6.24)
v	$1.92432 \cdot 10^{-6}$	$1.926795 \cdot 10^{-6}$
M	46.2468	46.318787

Kiriş orta noktasının deplasman ve momentinin zamanla değişimleri şekil 6.3'de görülmektedir.

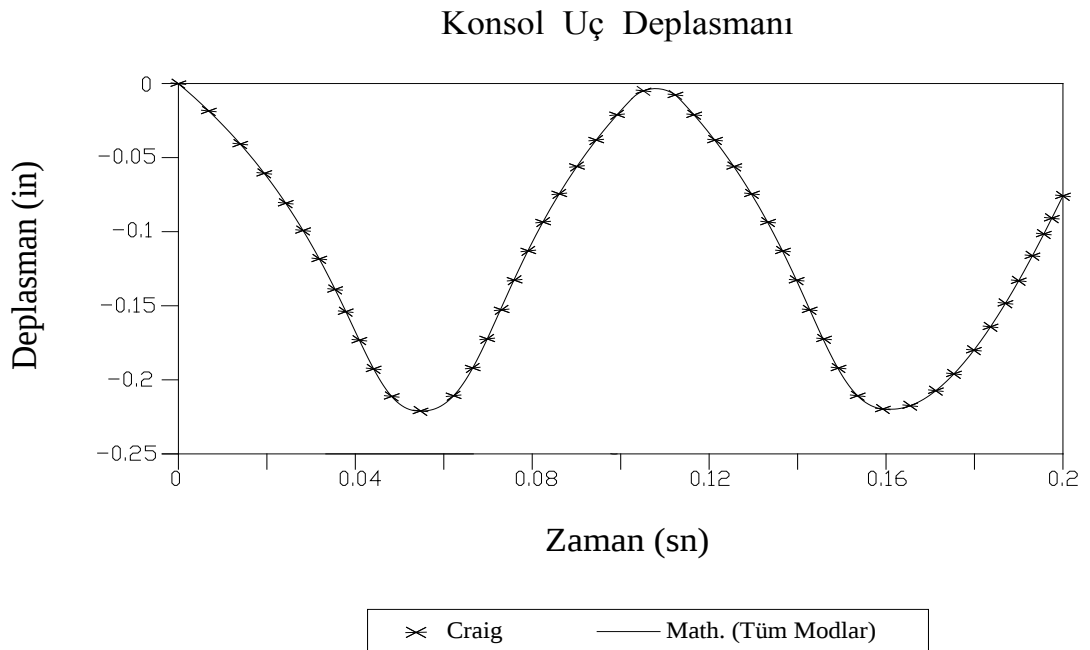
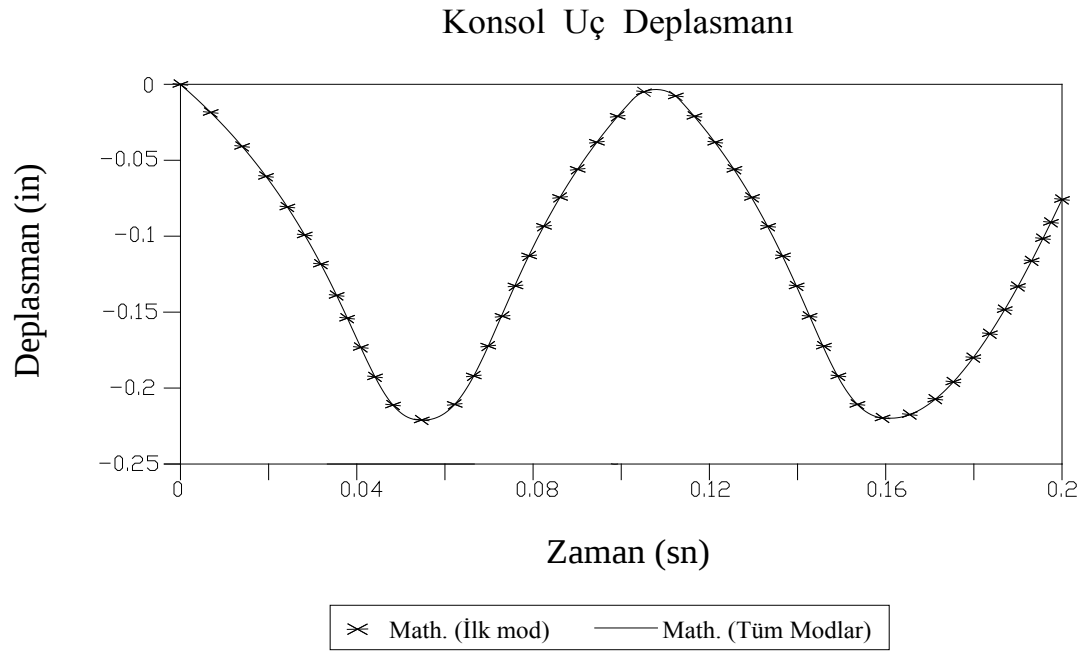
Örnek 6.2. Şekil 6.4.'te görülen ve adım tipi yükle zorlanan konsol kiriş sisteminin mod süperpozisyon yöntemi ile uç noktasındaki deplasmanın zamanla değişiminin bulunması:

Bu örnek (Craig, 1981)'de çözülmüş olup, serbest titreşim frekansları aşağıdaki gibi elde edilmiştir.

Çizelge 6.3. Örnek 6.2.'e ait serbest titreşim frekansları

	1	2	3	4	5	6
Craig	9.19655	157.7099	512.8766	1192.354	2290.459	4055.944
MSSN	9.19229	157.637	512.639	1191.8	2289.4	4054.07

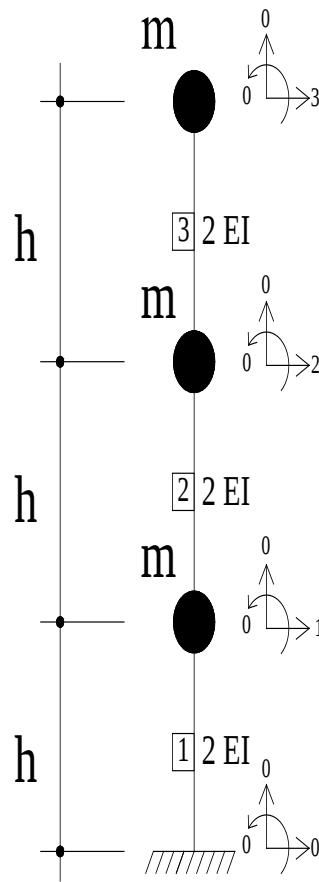
Konsol ucundaki maksimum deplasman (Craig, 1981)' de $\delta = 0.215949$ inc, MSSN.MAT ile $\delta = 0.216056$ inc bulunmuştur. Bu örnekte sadece ilk mod göz önüne alınarak yapılan hesap sonucunda bulunan değerlerin yeterli olduğu gözlenmiştir. Konsol uç deplasmanının zamanla değişimi şekil 6.5.'te görülmektedir.



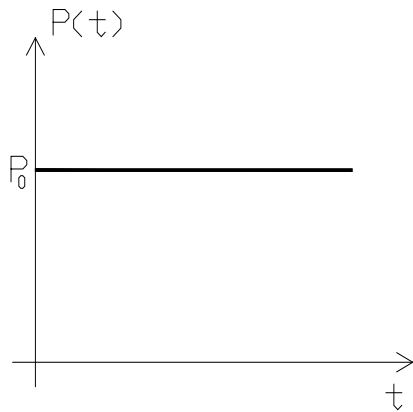
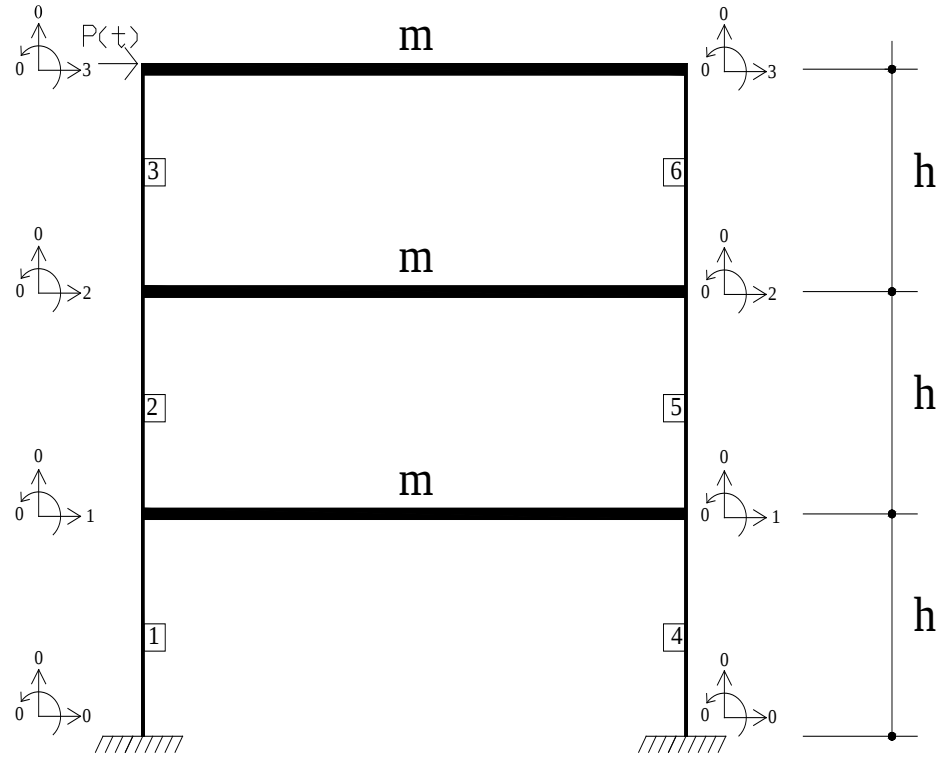
Şekil 6.5. Deplasman diyagramı

Örnek 6.3. Şekil 6.7.'de görülen çerçeve sistem adım tipi yükle zorlanmaktadır. Sistemde kütlenin kat hizalarında toplandığı, kolonların kütsüz olduğu ve ayrıca kolonların eksenel deformasyon yapmadığı kabul edilmektedir. Bu durumda sistemin 3 numaralı serbestlik derecesi yönündeki deplasmanının maksimum değerin ve 1 numaralı elemandaki maksimum kesme kuvvetinin bulunması istenmektedir.

Yukarıda yapılan kabullerden yararlanarak, çerçeve sistem şekil 6.6.'da görüldüğü gibi modellenenbilmektedir.



Şekil 6.6. Örnek 6.3.'ün Basitleştirilmiş modeli



$$h=300 \text{ cm}$$

$$m=1 \frac{\text{t.s}^2}{\text{cm}}$$

$$EI=40 \cdot 10^6 \text{ t.cm}^2$$

$$P_0 = 10 \text{ ton}$$

$$\xi_1=0.05=\xi_2=\xi_3$$

Şekil 6.7. Örnek 6.3.'e ait şekil

Bu sistem çözülmüş olup, sonuçlar aşağıda görüldüğü gibi elde edilmektedir. Serbest titreşim frekansları,

$$\omega_1 = 2.65372$$

$$\omega_2 = 7.43555$$

$$\omega_3 = 10.74467$$

Üst kat yatay deplasmanı ve alt kat kolon kesme kuvveti modların katkıları ile beraber aşağıda görülmektedir.

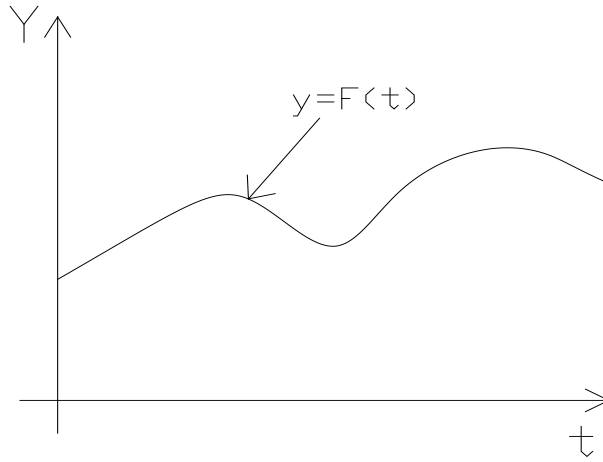
Çizelge 6.4. Örnek 6.3'e ait kesme kuvveti ve deplasman değerleri

		İlk mod	İlk iki mod	Tüm modlar
Max $ X_3 $ cm	Mengi	1.43	1.53	1.53
	MSRN.MAT	1.4302	1.5276	1.5324
Max $ V_1 $ ton	Mengi	11.31	9.73	10.08
	MSRN.MAT	11.3158	9.7347	10.1095

7. FOURIER TRANSFORM METODU

7.1. Fourier Dönüşümü

Zaman uzayında ve $t \geq 0$ zaman aralığında tanımlanan, reel değerlere sahip bir $F(t)$ fonksiyonu ele alınacak olursa (şekil 7.1),



Şekil 7.1. Zaman uzayında $F(t)$ fonksiyonu

zaman uzayındaki bu fonksiyon için Fourier dönüşüm ifadesi,

$$F^F(\omega) = \int_0^{\infty} F(t) e^{-i\omega t} dt \quad (7.1)$$

şeklinde verilmektedir. Denkleminde görülen ω açısal frekans değerine karşılık gelmektedir. (7.1) ifadesinden görüldüğü üzere, fonksiyonun Fourier uzayındaki ifadesi kompleks değerlere sahip olmaktadır. Fourier uzayında tanımlanmış bir fonksiyonun zaman uzayındaki ifadesini bulmak için ise aşağıda görülen ters Fourier dönüşüm bağıntısı kullanılmaktadır, (Clough ve Penzien, 1993).

$$F(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F^F(\omega) e^{i\omega t} d\omega \quad (7.2)$$

Yukarıda görülen (7.1) ve (7.2) ifadeleri yardımı ile herhangi bir fonksiyon için Fourier ve ters Fourier dönüşüm değerleri bulunmaktadırlar.

7.2. Fourier Dönüşümün Bazı Özellikleri

1-) $F(t)$ fonksiyonu $t \geq 0$ için reel değildir. Diğer taraftan, fonksiyonun Fourier dönüşümü, $F^F(\omega)$, kompleks değerlere sahip olan bir fonksiyon olup $(-\infty, \infty)$ aralığında tanımlanmaktadır. Ayrıca, $F^F(\omega)$ fonksiyonun $(-\omega)$ ve $(+\omega)$ noktalarındaki değerleri birbirlerinin kompleks eşleniği olmaktadır.

$$F^F(-\omega) = \overline{F^F(\omega)} \quad (7.3)$$

2-) $F(t)$ fonksiyonun zamana göre birinci türevinin (dF/dt) Fourier dönüşümü,

$$\dot{F}^F(\omega) = \int_0^{\infty} \dot{F}(t) e^{-i\omega t} dt \quad (7.4)$$

şeklinde bulunabilmektedir. (7.4) ifadesi kısmi integrasyon ile çözümlerse,

$$\dot{F}^F(\omega) = F(t) e^{-i\omega t} \Big|_0^{\infty} + i\omega \int_0^{\infty} F(t) e^{-i\omega t} dt \quad (7.5)$$

$$\dot{F}^F(\omega) = F(t) e^{-i\omega t} \Big|_{t=\infty} - F(0) + i\omega F^F \quad (7.6)$$

olacak şekilde fonksiyonun Fourier dönüşümü bulunmaktadır. Ayrıca,

$$\lim_{t \rightarrow \infty} F(t) = 0 \quad (7.7)$$

kabulü yapılmaktadır (Radition Condition). Yukarıdaki ifade, $F(t)$ fonksiyonun zaman ilerledikçe sönümlendiğinin kabul edildiği anlamına gelmektedir. Bu durum göz önüne alındığında (7.6) ifadesi,

$$\dot{F}^F(\omega) = i\omega F^F - F(0) \quad (7.8)$$

olarak elde edilmektedir. Benzer olarak, $F(t)$ fonksiyonun zamana göre ikinci türevinin $(d^2 F / dt^2)$ Fourier dönüşümü aşağıdaki gibi elde edilmektedir.

$$\ddot{F}^F(\omega) = (i\omega)^2 F^F - i\omega F(0) - \dot{F}(0) \quad (7.9)$$

Buraya kadar yapılanlar dikkate alınarak, n 'inci dereceden türev için Fourier dönüşüm ifadesi aşağıdaki gibi genelleştirilebilmektedir.

$$(F^n)^F = (i\omega)^n F^F - (i\omega)^{n-1} F(0) - \dots - i\omega F^{n-2}(0) - F^{n-1}(0) \quad (7.10)$$

$F(t)$ fonksiyonu ve türevleri için başlangıç şartlarının sıfır olduğu varsayılırsa Fourier dönüşüm ifadeleri şu hale gelmektedir:

Çizelge 7.1. Fourier dönüşüm ifadeleri

i

Zaman Uzayı	Fourier Uzayı
F	F^F
dF / dt	$i\omega F^F$
$d^2 F / dt^2$	$(i\omega)^2 F^F$
$d^n F / dt^n$	$(i\omega)^n F^F$

7.3. Ayırık Fourier Dönüşümü

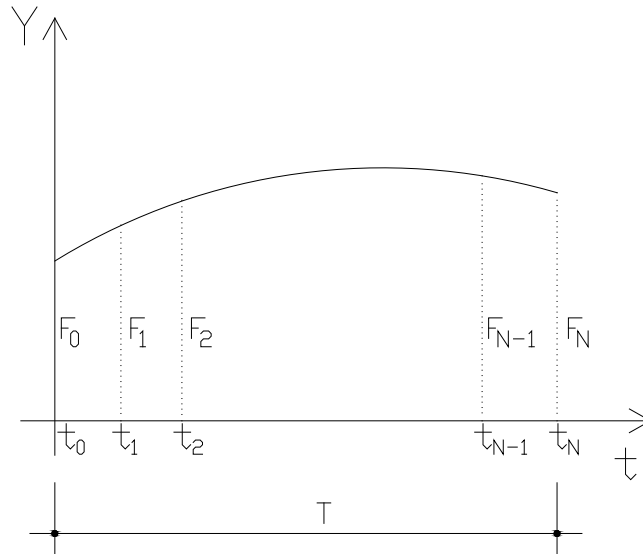
Burada herhangi bir $F(t)$ fonksiyonun sürekli olduğu ve $[0, T]$ aralığında Δt artımlara ayırık olarak,

$$F_i = F(t_i) \quad i = 0, 1, \dots, N-1 \quad (7.11)$$

şeklinde olduğu kabul edilmektedir (şekil 7.2.) (7.11) ifadesindeki t_i zaman ifadesi,

$$t_i = i \Delta t \quad \Delta t = \frac{T}{N} \quad (7.12)$$

olarak elde edilmektedir.



Şekil 7.2. $F(t)$ nin ayırık tanımı

Ayrık Fourier dönüşüm metodu, ayırık olarak verilen $F(t)$ fonksiyonun Fourier dönüşümünü bulmak için kullanılmaktadır.

7.3.1. Ayırık Fourier Dönüşümünün Önemi

Ayrık Fourier dönüşümünün önemini genel olarak iki noktada toparlayabiliriz. Bunlardan birincisi cisimlerin, yapıların, akışkanların, dinamik davranışını incelemeye yönelik deneylerde ivme, deplasman, hız gibi büyüklüklerin zamana göre değişimleri küçük zaman aralıkları (Δt) ile belirli bir zaman aralığı $[0, T]$ 'de ayırık olarak ölçülüp sonuçlar kaydedilmektedir. Elde edilen bu verilerin değerlendirilmesi için, özellikle deprem ve titreşim analizlerinde, büyüklüklerin ayırık olarak ölçülmüş zamana göre değişimlerinin Fourier spektrumlarının bulunması istenmektedir. Böyle bir amaca, ayırık Fourier dönüşüm formüllerini kullanarak erişmek mümkündür.

İkinci olarak ayırık Fourier dönüşümü, cisimlerin ve yapıların dinamik davranışının analitik olarak bulunmasında kullanılmaktadır. Bilindiği gibi yapıların dinamik davranışı, diferansiyel denklemler tarafından idare edilmektedir. Fourier dönüşümünün kullanılması halinde, yapıların istenilen bir noktasındaki dinamik davranışı, diğer noktalarındaki davranış belirlenmeden bulunabilmektedir. Oysaki klasik integrasyon metotlarında, bir noktadaki davranışı bulmak için diğer noktalardaki davranışın bilinmesine ihtiyaç duyulmaktadır.

7.3.2. Kompleks Fourier Serileri

Ayrık Fourier dönüşüm formülleri, kompleks Fourier serileri kullanılarak elde edileceğinden, bu konuda öncelikle kompleks Fourier serilerinin elde edilmesi ve sürekli Fourier dönüşümü ile ilişkilerine değinilmektedir.

Periyodik bir fonksiyon, belirli bazı koşulları sağladığı takdirde, harmonik fonksiyonlar ile ifade edilebilmekte ve bunlara Fourier serileri ismi verilmektedir. Periyodu T olan, $t \geq 0$ zaman aralığında ve reel değerlere sahip bir $F(t)$ fonksiyonu,

$$F(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi n t}{T}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi n t}{T}\right) \quad (7.13)$$

şeklinde Fourier serisine açılabilir. (7.13) ifadesinde bulunan a_0, a_n, b_n sayılarına Fourier transform katsayıları denmekte ve aşağıdaki şekli ile ifade edilmektedir, (Celep ve Kumbasar, 1992).

$$\begin{aligned} a_0 &= \frac{1}{T} \int_0^T F(t) dt & a_n &= \frac{1}{T} \int_0^T F(t) \cos\left(\frac{2\pi n t}{T}\right) dt \\ b_n &= \frac{1}{T} \int_0^T F(t) \sin\left(\frac{2\pi n t}{T}\right) dt \end{aligned} \quad (7.14)$$

Bunlarla birlikte, trigonometrik ve üstel fonksiyonlar arasındaki,

$$\sin t = \frac{e^{it} - e^{-it}}{2i} \quad \cos t = \frac{e^{it} + e^{-it}}{2} \quad (7.15)$$

bağlantıları göz önüne alındığı takdirde (7.13) ifadesi aşağıdaki hale gelmektedir:

$$\begin{aligned} F(t) &= \sum_{k=-\infty}^{\infty} C_k e^{(i\omega_k t)} \\ C_k &= \frac{1}{T} \int_0^T F(t) e^{(-i\omega_k t)} dt \end{aligned} \quad (7.16)$$

Yukarıda görülen ω_k k'nıncı Fourier bileşeninin açısal frekansını göstermekte olup,

$$\begin{aligned} \omega_k &= \frac{2\pi k}{T} = \frac{2\pi}{T_k} = 2\pi f_k \\ f_k &= \frac{k}{T} = k \Delta f \quad \Delta f = \frac{1}{T} \end{aligned} \quad (7.17)$$

bağıntısı ile tanımlanmaktadır. (7.16) ifadelerinde görülen C_k Fourier sabitleri Δf frekans artımına bölünürse, yayılı Fourier sabitleri elde edilmektedir.

$$F_k^F = \frac{C_k}{\Delta f} \quad (7.18)$$

Böylece, Fourier serisi denklemleri yayılı Fourier sabitleri cinsinden aşağıdaki gibi elde edilmektedir.

$$F(t) = \Delta f \sum_{k=-\infty}^{\infty} F_k^F e^{(i\omega_k t)} \quad (7.19)$$

$$F_k^F = \int_0^T F(t) e^{(-i\omega_k t)} dt$$

Sürekli Fourier dönüşüm formülleri, kompleks Fourier serisi formüllerinden periyod (T) sonsuza giderken elde edilmektedir. Başka bir deyişle,

$$T \rightarrow \infty \Rightarrow \left(\begin{array}{ll} \Delta f = \frac{1}{T} & \rightarrow df \\ \omega_k = 2\pi f_k & \rightarrow \omega = 2\pi f \end{array} \right) \quad (7.20)$$

olur ve dolayısıyla Kompleks Fourier serileri,

$$F(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F^F(\omega) e^{i\omega t} d\omega \quad (7.21)$$

$$F^F(\omega) = \int_0^{\infty} F(t) e^{-i\omega t} d\omega$$

şeklinde, (7.1) ve (7.2) ile verilen sürekli Fourier transform formüllerine dönüşmektedir.

7.4. Ayırık Fourier Dönüşüm Formülleri

Ayrık Fourier dönüşüm formülleri, kompleks Fourier serisi formülleri yardımı ile elde edilmektedir. (7.19) ayrık Fourier serisi denkleminde ikincisi olan integral ifadesi, nümerik olarak dikdörtgenler metodu ile bulunup, $t = t = n \times \Delta t$ zamanında yazılmak istenirse,

$$\omega_n t_k = (2\pi n \Delta f)(k \Delta t) = 2\pi \frac{nk}{N} \quad (7.22)$$

olmak üzere,

$$F_n = \Delta f \sum_{k=-\infty}^{\infty} F_k^F e^{i2\pi \frac{nk}{N}} \quad (7.23)$$

$$F_n^F = \Delta t \sum_{k=0}^{N-1} F_k e^{-i2\pi \frac{nk}{N}}$$

ifadeleri elde edilmektedir. Yukarıdaki (7.23) ifadelerinde Δt aralıklar ile temsil edilebilecek Fourier bileşenlerinin maksimum frekansı,

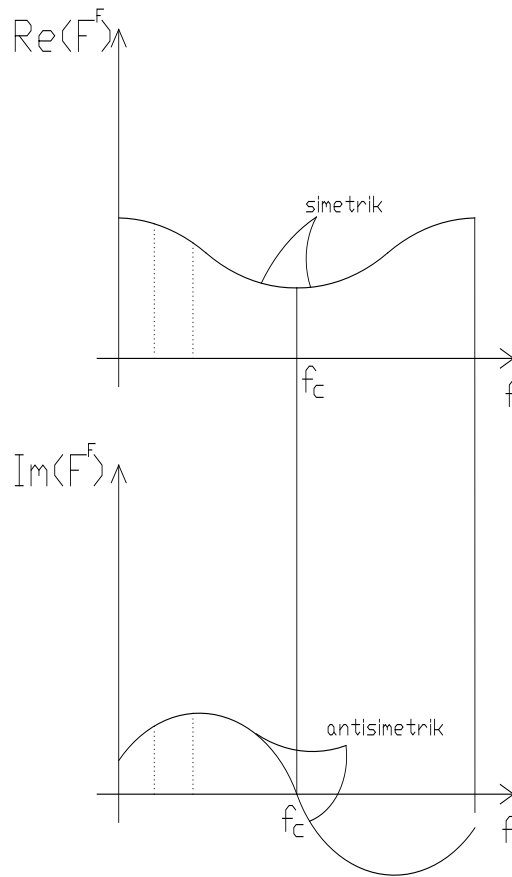
$$f_c = \frac{1}{2\Delta t} = \frac{N}{2} \Delta f \quad (7.24)$$

olarak elde edilmektedir. Burada f_c 'ye katılma frekansı denmekte ve f_c 'den daha büyük frekanslı Fourier bileşenlerinin, f_c den küçük frekanslı Fourier bileşenleri üzerine katıldığı varsayılmaktadır. Bu şekilde tanımlanan katlama frekansının özelliğinden dolayı, ayırık Fourier dönüşüm formülleri son şeklini almaktadır, (Clough ve Penzien, 1993).

$$F_n = \Delta f \sum_{k=-\infty}^{\infty} F_k^F e^{i2\pi \frac{nk}{N}}$$

$$F_n^F = \Delta t \sum_{k=0}^{N-1} F_k e^{-i2\pi \frac{nk}{N}} \quad (7.25)$$

Yukarıdaki denklemle elde edilen Fourier dönüşümün reel ve imajiner kısımları f_c katlama frekansına göre sırası ile simetrik ve antisimetrik olmaktadır (şekil 7.3.)



Şekil 7.3. Ayırık Fourier dönüşümünün simetri ve antisimetri özelliği

7.5. Fourier ve Ters Fourier Dönüşümünün Bulunması

Zaman uzayında verilen herhangi bir $F(t)$ fonksiyonun Fourier dönüşümünü veya frekans uzayında verilen veya bulunan bir $F^F(\omega)$ fonksiyonun ters Fourier dönüşümünü bulmak için (7.25) ayrık Fourier dönüşüm denklemleri kullanılmaktadır.

7.5.1. Fourier Dönüşümü Bulmak İçin İzlenecek Yöntem

$F(t)$ nin belirli bir zaman aralığında verildiği kabul edilirse, $F^F(\omega)$ fonksiyonunu bulmak için aşağıdaki adımlar takip edilmektedir, (Craig, 1981).

- a-) Analizde göz önüne alıncak zaman aralığı $[0, T]$ seçilir.
- b-) Bölme sayısı N seçilir ve $\Delta t = T / N$ olacak şekilde her Δt zaman artırımını için fonksiyonun değerleri tesbit edilir.
- c-) Aşağıdaki bağıntıdan X_n değerleri bulunur:

$$X_n = \sum_{k=0}^{N-1} W^{nk} F_k \quad n = 1, 2, \dots, N-1 \quad (7.26)$$

$$W = e^{-i \frac{2\pi}{N}}$$

- d-) Son olarak Fourier dönüşümü hesaplanır:

$$F_n^F = \Delta t X_n \quad n = 1, 2, \dots, N-1 \quad (7.27)$$

7.5.2. Ters Fourier Dönüşümünü Bulmak İçin İzlenecek Yöntem

$F^F(\omega)$ fonksiyonun verildiği kabul edilirse, $F(t)$ fonksiyonunu bulmak için sırası ile aşağıdaki işlemler yapılmaktadır, (Craig, 1981).

- a-) Analizde göz önüne alınacak frekans aralığı $[0, f_c]$ belirlenir.
- b-) Bölme sayısı N seçilir ve $F^F(\omega)$ nın Δf aralıklar ile değerleri tesbit edilir.
- c-) $F^F(\omega)$ fonksiyonu, reel ve imajiner kısımları katlama frekansı etrafında simetrik ve antisimetrik olacak şekilde $[0, 2f_c]$ aralığında tanımlanır.
- d-) X_n değeri hesaplanır.

$$X_n = \sum_{k=0}^{N-1} W^{nk} F_k \quad n = 1, 2, \dots, N-1$$

$$W = e^{i \frac{2\pi}{N}}$$
(7.28)

- e-) Ters Fourier dönüşümü hesaplanır.

$$F_n = \Delta f X_n \quad n = 1, 2, \dots, N-1$$
(7.29)

7.6. Fourier Dönüşümünün Yapı Dinamiği Problemlerine Uygulanması

Yapı dinamiğinde sistemin hareketi diferansiyel denklemler tarafından idare edilmektedir. Bu diferansiyel denklemlerin nasıl çözüleceği daha önceki bölümlerde bahsedilmekteydi. Burada, bahsedilen denklem sistemlerinin çözümünün Fourier transform tekniği ile nasıl yapıldığı anlatılmaktadır.

7.6.1. Tek Serbestlik Dereceli Sistemlerin Çözümü

Tek serbestlik dereceli bir sistemde olayı idare eden diferansiyel denklem aşağıdaki gibidir.

$$\ddot{x} + 2\xi\bar{\omega}\dot{x} + \bar{\omega}^2 x = \frac{P}{m} \quad (7.30)$$

Burada $\bar{\omega}$ serbest titreşim frekansıdır. Bu aşamada (7.30) denklemi Fourier uzayında yazılmak istenirse, sistem denklemi aşağıdaki halini almaktadır.

$$\dot{x}^F = (i\omega)x^F \quad (7.31)$$

$$\ddot{x}^F = (i\omega)^2 x^F$$

$$-\omega^2 x^F + 2\xi\bar{\omega}(i\omega)x^F + \bar{\omega}^2 x^F = \frac{P^F}{m} \quad (7.32)$$

Denklemlerde görülen ω Fourier transform parametresidir. Yukarıdaki denkleme, frekans uzayında olayı idare eden denklem denir ve cebrik bir denklemi ifade etmektedir. Bu denklemin çözümünden x^F bulunmaktadır, (Celep ve Kumbasar, 1992).

$$x^F = \frac{1}{m} \left[\frac{1}{-\omega^2 + 2\xi\bar{\omega}(i\omega) + \bar{\omega}^2} \right] P^F \quad (7.33)$$

Fourier transformun girişim özelliğinden (Convolution property) yararlanarak x^F in ters Fourier transformu aşağıdaki gibi bulunmaktadır.

$$x(t) = \frac{1}{m \omega_D} \int_0^t p(\tau) e^{-\xi \bar{\omega}(t-\tau)} \sin[\omega_D(t-\tau)] d\tau$$

$$\omega_D = \bar{\omega} \sqrt{1 - \xi^2}$$
(7.34)

Yukarıda görülen ω_D sistemin sönümlü serbest titreşim frekansı olmakta ve bu denkleme tek serbestlik dereceli sistemler için Duhamel integrali denilmektedir, (Celep ve Kumbasar, 1992).

7.6.2. Çok Serbestlik Dereceli Sistemlerin Çözümü

Çok serbestlik dereceli sistemler için olayı idare eden dieferansiyel denklem takımı matris formatı ile aşağıdaki gibi ifade edilmektedir.

$$\underline{M} \ddot{\underline{X}} + \underline{C} \dot{\underline{X}} + \underline{K} \underline{X} = \underline{P}$$
(7.35)

Aşağıdaki gibi bir dinamik rijitlik matrisi tanımlanacak olursa sistem denklemi,

$$\underline{D} \underline{X} = \underline{P}$$

$$\underline{D} = (\underline{M} d^2 / dt^2 + \underline{C} d / dt + \underline{K})$$
(7.36)

haline gelmektedir. Denklemdaki $\underline{D}$, zaman uzayında dinamik rijitlik matrisine karşılık gelmektedir. (7.36) denklemi ile görülen sistem denkleminin Fourier transformu alınır, frekans uzayında sistem denklemi şu hale gelmektedir.

$$\underline{D}^F \underline{X}^F = \underline{P}^F$$

$$\underline{D}^F = (i\omega)^2 \underline{M} + (i\omega) \underline{C} + \underline{K}$$
(7.37)

Denklemdaki $\underline{D}^F$ matrisine frekans uzayında dinamik rijitlik matrisi denilmektedir. Sistem denkleminde de görüldüğü üzere, frekans uzayında olayı idare eden denklem takımı cebrik denklem takımına dönüşmektedir, (Celep ve Kumbasar, 1992).

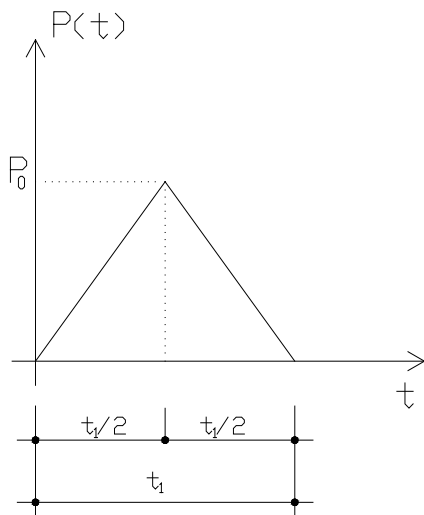
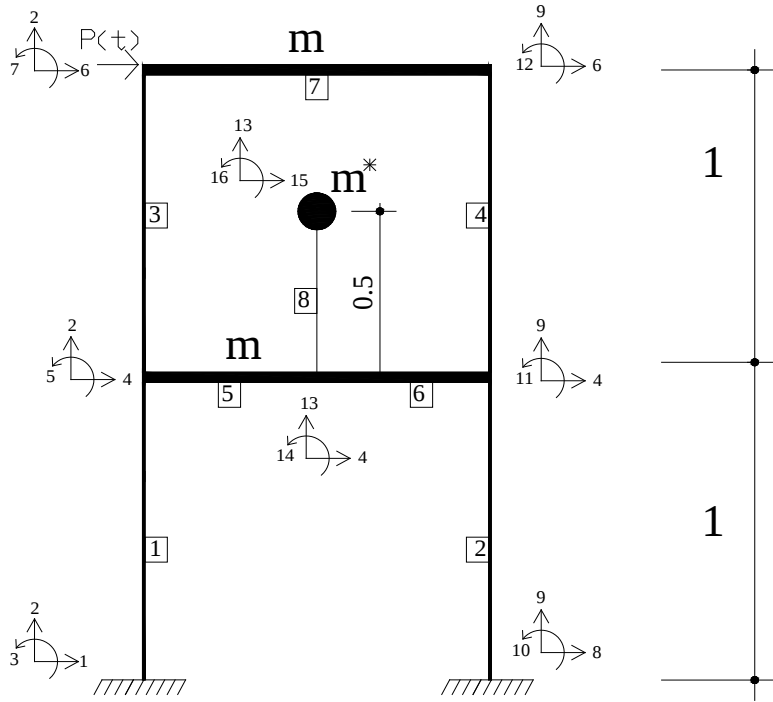
$$\underline{X}^F = \underline{H} \underline{P}^F \quad (7.38)$$

$$\underline{H}(\omega) = (\underline{D}^F)^{-1}$$

$$\underline{X}(t) = \int_0^t \underline{h}(t - \alpha) \underline{P}(\alpha) d\alpha \quad , \quad \underline{h}^F(\omega) = \underline{H} \quad (7.39)$$

(7.37) denklem takımının çözümü frekans uzayında bilinmeyenleri vermektedir ($\underline{X}^F$). Daha sonra buna ters Fourier transformu uygulanarak zaman uzayında sistemin bilinmeyenleri hesaplanmaktadır, $\underline{X}(t)$. Zaman uzayında bilinmeyenler bulunduğundan sonra istenen diğer büyüklükler hesaplanabilmektedir.

Örnek 7.1. Şekil 7.4.'te görülen sistem, üçgen yükleme ile zorlanmaktadır. Bu yükleme altında sistemde 8 numaralı elemandaki kesme kuvvetinin ve 15 numaralı serbestlik derecesi yönündeki deplasmanın maksimum değerlerinin bulunması istenmektedir.



Boyutsuz büyüklükler

$$\bar{P}=1 \quad ; \quad \bar{t}_1=1 \quad ; \quad \bar{t}_1/\bar{T}=0.3$$

$$EI=1 \quad ; \quad \bar{\xi}=0.02 \quad ; \quad \bar{T}=2\pi/\bar{\omega}$$

$$\bar{m}^* = \frac{24}{(2\pi\bar{t}/\bar{T})^2} = 6.7547$$

Şekil 7.4. Örnek 7.1.'e ait şekil

Bu problem boyutsuz uzayda çözüleceğinden gerekli boyutsuz büyüklükler şekilde verilmektedir. Hem Fourier transform metodu hemde Newmark metodu ile ekler bölümünde verilen Mathematica programları yardımı ile çözülmüş sonuçlar aşağıda gösterilmektedir.

Çizelge 7.2. Örnek 7.1.'e ait değerler

	MSRFT.MAT	MSRN.MAT	Mengi
$\delta_{\max}$	0.0590622	0.0535303	0.0507012
$V_{\max}$	1.0663	1.14109	1.17624

Tablodan da görülebileceği üzere, verilen değerlere, Newmark metodu ile çözüm yapan MSRN.MAT ile elde edilen değerler, Fourier transform metodu ile çözüm yapan MSRF.MAT ile elde edilen değerlerden daha yakındır.

8. DEPREM ANALİZİ

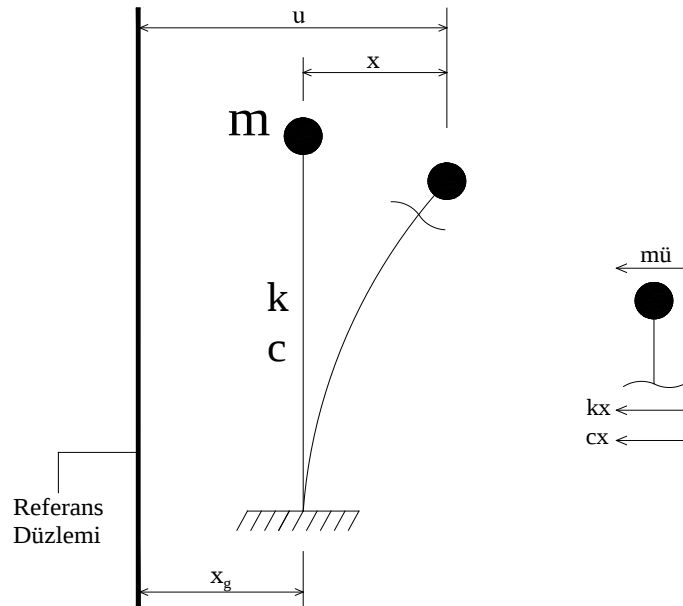
8.1. Giriş

Depremler aniden meydana gelmesinden dolayı, doğal afetler arasında önemli bir yer tutmaktadır. Güvenilir bir uyarı sisteminin geliştirilememiş olması, yapıların depreme karşı dayanıklı ve sağlam tasarlanarak, deprem etkilerinin azaltılması gerekliliği ortaya çıkmıştır.

Deprem yerkabuğunun bir titreşimi olduğu için, yapıların yerkabuğu ile temas ettiği noktalarda zamana bağlı bir yer değiştirme oluşarak, yapı dinamik bir etkiye maruz kalmaktadır.

8.2. Yer Hareketi Durumunda Yapıların Davranışı

Yer hareketinin meydana geldiği anda yapıya etkiyen yük, yapının kütlesi ve yer hareketinin zamanla değişiminin büyüklüğüne bağlı olarak oluşmaktadır (şekil 8.1.).



Şekil 8.1. Yer hareketi bulunması durumunda yapıların hareketi

Şekil 8.1'den de görüldüğü üzere sistemin toplam deplasmanı (u), yer hareketi (x_g) ve yapının yere göre rölatif deplasmanı (x) cinsinden,

$$u = x + x_g \quad (8.1)$$

şeklinde elde edilir. Yine şekil 8.1'de görülen serbest cisim diyagramı için D'Alembert prensibi uygulanırsa, hareket denklemi şöyle bulunmaktadır:

$$\sum F_x = 0 \Rightarrow m\ddot{u} + c\dot{x} + kx = 0 \quad (8.2)$$

$$\begin{aligned} \ddot{u} &= \ddot{x} + \ddot{x}_g \\ m\ddot{x} + m\ddot{x}_g + c\dot{x} + kx &= 0 \end{aligned} \quad (8.3)$$

$$m\ddot{x} + c\dot{x} + kx = -m\ddot{x}_g(t) \quad (8.4)$$

(8.4) denkleminde eşitliğin sağ tarafına götürülen ifade, yer hareketi nedeni ile meydana gelen dinamik yüke karşı gelmektedir. Denkleminde eşitliğin her iki tarafı m ile bölünür ve uygun dönüşümler yapılırsa,

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = -\ddot{x}_g(t) \quad (8.5)$$

şeklinde hareket denklemi elde edilmektedir. Görüldüğü gibi hareket denklemi, diferansiyel denklem şeklindedir. Bu denklem, daha önceki konularda anlatılan metotlardan herhangi biri ile (Newmark, Wilson- θ vb.) çözülüp sistemin deplasmanlarının zamanla değişimi bulunmakta ve istenirse eleman kuvvetleri tespit edilmektedir.

Eğer sistem çok serbestlik dereceli ise, öncelikle sistemin hareket denklemi elde edilmekte ve daha sonra uygun bir yöntemle (Mod süperpozisyon vb.) sistemin bilinmeyenleri bulunmaktadır.

8.3. Spektrum Analizi

Bütün modlar için sayısal integrasyon yapmak ve zamana bağlı deplasmanları bulduktan sonra bunları süperpoze ederek gerçek deplasmanlara ve iç kuvvetlere geçmek çok uğraştırıcı ve zaman alıcıdır. Mühendislik problemlerinde genellikle maksimum değerler gerekli olduğundan, her mod için bilinmeyenlerin sadece maksimum değeri bulunup, bu maksimumların süperpozisyonu yapılarak sonuca gidilmektedir.

(8.5) diferansiyel denkleminin çözümü, Duhamel integrali yardımı ile aşağıdaki gibi verilmektedir, (Celep ve Kumbasar, 1992).

$$x(t) = \frac{1}{m\omega_D} \int_0^t p(\tau) e^{-\xi\omega(t-\tau)} \sin[\omega_D(t-\tau)] d\tau \quad (8.6)$$

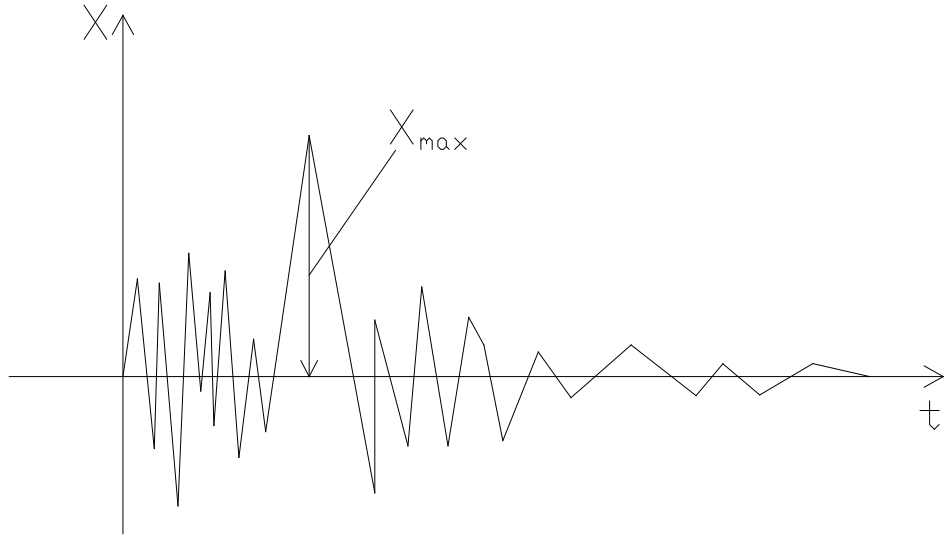
$$P(\tau) = -m\ddot{x}_g(\tau) \quad ; \quad \omega_D = \omega\sqrt{1-\xi^2}$$

(8.6) denklemindeki ω_D , sistemin sönümlü serbest titreşim frekansıdır. Çoğu yapı mekaniği problemlerinde, sönüm oranı $\xi \leq 0.1$ olduğundan dolayı sistemin sönümlü serbest titreşim frekansı sönümsüz serbest titreşim frekansına eşit alınmaktadır ($\omega_D \approx \omega$). Bu durumda (8.6) denklemi düzenlenirse, Duhamel integrali şu hale gelmektedir.

$$x(t) \cong -\frac{1}{\omega} \int_0^t \ddot{x}_g(\tau) e^{-\xi\omega(t-\tau)} \sin[\omega(t-\tau)] d\tau \quad (8.7)$$

(8.7) denkleminin çözümü sistem deplasmanlarının zamanla değişimini vermektedir.

Çözümün aşağıdaki gibi olduğunu varsayıp,



Şekil 8.2. Deplasmanın zamana göre değişimi

maksimum deplasmana $x_{\max} = \max |x|$ denilirse, (8.7) denklemi şu hale gelmektedir:

$$x_{\max} = \frac{1}{\omega} \max \left| \int_0^t \ddot{x}_g(\tau) e^{-\xi \omega(t-\tau)} \sin[\omega(t-\tau)] d\tau \right| \quad (8.8)$$

Mutlak değer içindeki ifadeye hız spektrumu (S_v) denilmektedir.

$$S_v = \max \left| \int_0^t \ddot{x}_g(\tau) e^{-\xi \omega(t-\tau)} \sin[\omega(t-\tau)] d\tau \right| \quad (8.9)$$

Bu şekilde Duhamel integrali aşağıdaki halini almaktadır.

$$x_{\max} = \frac{S_v}{\omega} = S_d \quad (8.10)$$

(8.10) denklemindeki S_d , deplasman spektrumunu göstermektedir. S_a , ivme spektrumu olmak üzere aşağıdaki şekli ile ifade edilmektedir.

$$S_a = \omega S_v \quad (8.11)$$

Deprem spektrumlarından birinin bilinmesi durumunda diğerlerini bulmak oldukça kolaydır. Sistem mutlak maksimum deplasmanın (deprem deplasman spektrumu) bilindiği farz edilirse, diğerleri aşağıda gösterildiği şekilde bulunmaktadır, (Clough ve Penzien, 1993)

$$\begin{pmatrix} S_d = x_{\max} \\ S_v = \omega S_d \\ S_a = \omega S_v = \omega^2 S_d \end{pmatrix}, \quad \omega = \frac{2\pi}{T} \quad (8.12)$$

Deprem spektrumları, sistemin sönüm oranı (ξ) ve serbest titreşim periyoduna (T) bağlı olarak, eğriler ile tanımlanmaktadırlar.

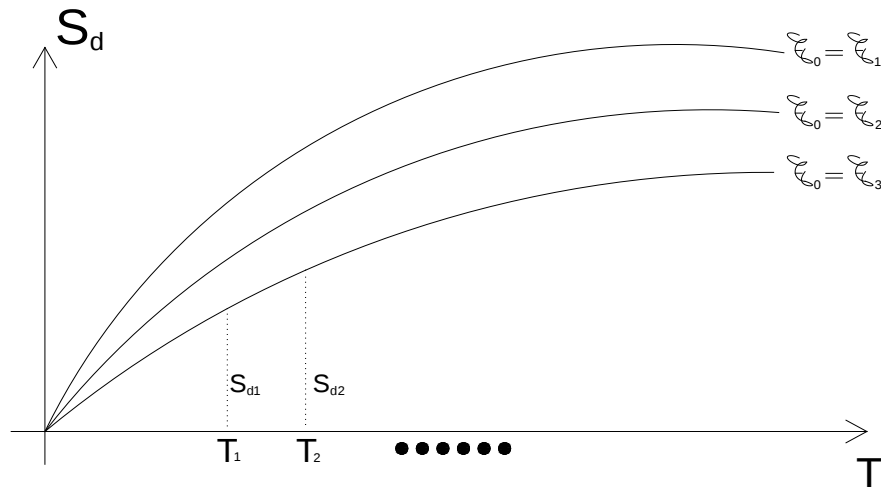
$$S_d = S_d(T, \xi) ; S_v = S_v(T, \xi) ; S_a = S_a(T, \xi) \quad (8.13)$$

Tipik olarak bir deplasman spektrum eğrisi aşağıdaki tabloda gösterildiği gibi elde edilmektedir.

Çizelge 8.1. Tipik deplasman spektrum eğrisi değerleri

T	$\xi = \xi_1$	$\xi = \xi_2$	$\xi = \xi_3$		$\xi = \xi_m$
T_1	$x_{\max} \rightarrow S_{d1}$	$x_{\max} \rightarrow S_{d1}$	$x_{\max} \rightarrow S_{d1}$		
T_2	$x_{\max} \rightarrow S_{d2}$				
.....					
T_n	$x_{\max} \rightarrow S_{dn}$				$x_{\max} \rightarrow S_{dn}$

Tablodaki değerler, spektrum eğrileri olarak şekil 8.3.'te görülmektedir.



Şekil 8.3. Deprem deplasman spektrum eğrileri

Ülkemizde yapılan projelerde, yönetmeliklerin belirlediği üzere deprem hesapları eşdeğer statik yük kabulüne göre yapılmaktadır. Türkiye Deprem Yönetmeliği (TDY) olarak bilinen bu yöntemde deprem hesaplarında spektrum analizi kullanılmaktadır. TDY'de önerilen hesapta ivme spektrumun şekil 8.4.'teki gibi olduğu varsayılmaktadır.

C_0 : Deprem bölge katsayısı

S_{ak} : k nıncı modun ivme spektrum değeri

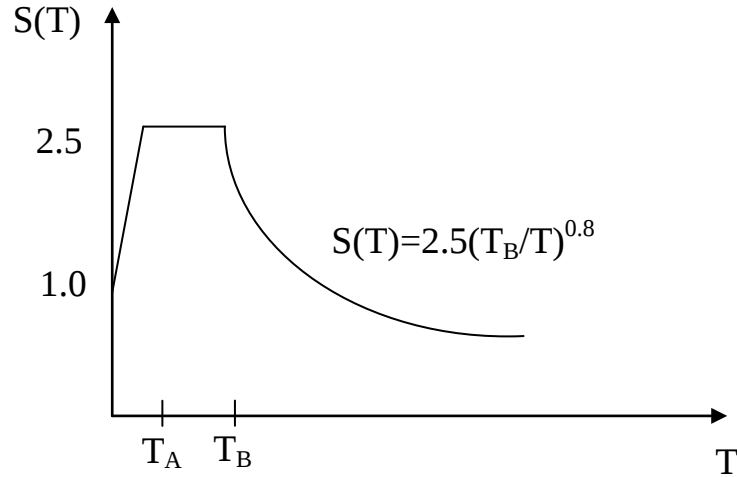
T_k : k nıncı modun periyodu

T_o : Zemin hakim periyodu

K : Yapı tipi katsayısı

I : Yapı önem katsayısı

g : Yerçekimi ivmesi



Şekil 8.4. TDY'nin Önerdiği Deprem İvme Spektrum Eğrisi

8.4. Çok Serbestlik Dereceli Sistemlerin Deprem Analizi

Tek serbestlik dereceli sistemlerde deprem söz konusu olduğu zaman, olayı idare eden denklem (8.4) bağıntısı ile verilmektedir. Eğer sistem çok serbestlik dereceli ise, olayı idare eden denklem takımı, matris formunda aşağıda gösterildiği şekildedir.

$$\underline{M} \ddot{\underline{X}} + \underline{C} \dot{\underline{X}} + \underline{K} \underline{X} = -\underline{M} \underline{r} \ddot{X}_g(t) \quad (8.14)$$

Denklemde görülen $\underline{r}$ vektörü statik etki katsayıları vektörü olarak tanımlanmaktadır. Bu vektöre, Pseudostatik vektörde denilmekte ve şöyle tanımlanmaktadır; Pseudostatik vektör, deprem hareketi yönünde birim yer hareketi nedeni ile, yapının rijit olması halinde, düğüm noktalarında aynı doğrultuda meydana gelecek yer değiştirmeleri göstermektedir. Pseudostatik vektörü $\underline{r}$, sistem serbestlik derecesi kadar mertebeli bir vektör olup, deprem hareketi ile aynı yöndeki serbestlik derecelerinin karşılığına 1, diğerlerine 0 konarak elde edilmektedir.

Sistem denkleminin sağında görülen ifade,

$$\underline{P}(t) = -\underline{M} \underline{r} \ddot{X}_g(t) \quad (8.15)$$

şeklinde yazılırsa sistem denklemi bilinen forma dönüşmektedir.

$$\underline{M} \ddot{\underline{X}} + \underline{C} \dot{\underline{X}} + \underline{K} \underline{X} = \underline{P}(t) \quad (8.16)$$

Sistem denklem takımı mod süperpozisyon metodu ile çözülmek istenirse,

$$\underline{X} = \underline{\Phi} \underline{Y} \quad (8.17)$$

dönüşümü yapıp yerine yazıldıktan sonra denklemin her iki tarafı soldan $\underline{\Phi}^T$ ile çarpılır ise sistem serbestlik derecesi kadar, girişimsiz diferansiyel denklem takımı elde edilmektedir. Tipik olarak bunlardan biri aşağıda gösterilmektedir.

$$\ddot{Y}_i + 2\xi_i \omega_i \dot{Y}_i + \omega_i^2 Y_i = \frac{P_i}{m_i} \quad i = 1, 2, \dots, n \quad (8.18)$$

Yukarıdaki denklemde eşitliğin sağında bulunan P_i yük ifadesi,

$$P_i = \underline{\Phi}_i^T \underline{P}(t) = -\underline{\phi}_i^T \underline{M} \underline{r} \ddot{x}_g(t) \quad (8.19)$$

Şeklindedir ve ayrıca,

$$L_i = \underline{\phi}_i^T \underline{M} \underline{r} \quad (8.20)$$

olacak şekilde dönüşüm yapılırsa yük ifadesi, aşağıda görülen formda ifade edilir.

$$P_i = -L_i \ddot{x}_g(t) \quad (8.21)$$

Bu durumda (8.18) denklemindeki yük ifadesi,

$$\frac{P_i}{m_i} = -\alpha_i \ddot{x}_g(t) \quad ; \quad \alpha_i = \frac{L_i}{m_i} \quad (8.22)$$

haline gelmektedir. (8.22) ifadesi (8.18) de yerine yazılırsa denklem şu hale dönüşmektedir.

$$\ddot{Y}_i + 2\xi_i \omega_i \dot{Y}_i + \omega_i^2 Y_i = -\alpha_i \ddot{x}_g(t) \quad i = 1, 2, \dots, n \quad (8.23)$$

Y_i : Modal deplasman

ω_i : Serbest titreşim frekansı

ξ_i : Sistem sönüm oranı

α_i : katıatı faktörü

Bilindiği gibi tek serbestlik dereceli sistemlerde, deplasman spektrumu, $x_{\max} = S_d(T, \xi)$ idi. Öyleyse, (8.23) denklemi için mutlak maksimum deplasman (deplasman spektrumu),

$$(Y_i)_{\max} = \max|Y_i| = \alpha_i S_d(T_i, \xi_i) \quad (8.24)$$

olarak elde edilir. Benzer olarak diğer spektrumlarda (hız ve ivme) şu şekildedir:

$$(Y_i)_{\max} = \alpha_i S_{di} = \frac{\alpha_i}{\omega_i} S_{vi} = \frac{\alpha_i}{\omega_i^2} S_{ai} \quad (8.25)$$

Görüldüğü üzere, eğer sistemin deprem spektrumlarından herhangi biri biliniyor ise, maksimum modal deplasman (8.25) bağıntısı yardımı ile kolayca bulunabilmektedir. Modal deplasmanlar bulunduktan sonra, sistemin gerçek deplasmanları hesaplanır.

$$\underline{X} = \underline{\Phi}\underline{Y} = \underline{\phi}_1 Y_1 + \underline{\phi}_2 Y_2 + \dots + \underline{\phi}_n Y_n \quad (8.26)$$

Eşitliğin sağında görülen birinci terim birinci modun katkısını, ikinci terim ikinci modun katkısını, n'inci terim n'inci modun katkısını göstermektedir. $(\underline{\phi}_i Y_i \rightarrow i'ninci \text{ modun katkı})$. i'ninci modun, deplasmana maksimum katkısı spektrumlar cinsinden aşağıdaki gibidir.

$$\underline{\phi}_i (Y_i)_{\max} = \underline{\phi}_i \frac{\alpha_i S_{vi}}{\omega_i} \quad (8.27)$$

Herhangi bir modun sistem deplasmanına maksimum katkısı b_i olarak gösterilirse, tüm modlar için maksimum katkılar aynı t zamanda meydana gelmemektedir. Bundan dolayı sistemin maksimum deplasman değerinin, modların

maksimum katkılarının (b_i) toplamı şeklinde olmayacağı görülmektedir. Sistemin maksimum deplasmanını bulmak için yaklaşık bazı yöntemler kullanılmaktadır. Bu yöntemlerden bir tanesi kareler toplamının karekökü (SRSS) yöntemidir. Bu yöntem kullanılarak sistemin maksimum deplasmanı, modların katkıları cinsinden,

$$(X_j)_{\max} \cong \sqrt{(b_1^2 + b_2^2 + \dots + b_n^2)} \quad (8.28)$$

şeklinde hesaplanmaktadır. Bu şekilde, sistemin bütün deplasmanları bulunduğundan sonra istendiği takdirde eleman kuvvetleri, kesit tesirleri vb. hesaplanabilmektedir.

Örnek 8.1. Örnek 6.3'te çözülen sistemin spektrum analizi ile deplasmanlarının maksimum değerlerinin bulunması istenmektedir.

Aynı örnek çözülmüş olup sistem için hız spektrum değerleri aşağıdaki gibi verilmektedir.

$$\omega_1 = 2.65 \rightarrow T_1 = 2.368 \rightarrow S_{v1} = 18.90 \text{ cm/s}$$

$$\omega_2 = 7.44 \rightarrow T_2 = 0.845 \rightarrow S_{v2} = 15.24 \text{ cm/s}$$

$$\omega_3 = 10.74 \rightarrow T_3 = 0.585 \rightarrow S_{v3} = 13.72 \text{ cm/s}$$

Sistem için Pseudostatik vektörü aşağıdaki gibi elde edilmektedir.

$$\underline{r} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Çözüm sonucunda elde edilen veriler aşağıdaki tabloda görülmektedir.

Çizelge 8.2. Örnek 8.1.'e ait sayısal veriler

		$(X_1)_{\max}$ cm	$(X_2)_{\max}$ cm	$(X_3)_{\max}$ cm
1. Mod	Mengi	3.87	6.97	8.69
	MSRSP.MAT	3.8682	6.9703	8.6919
2. Mod	Mengi	0.72	0.32	-0.57
	MSRSP.MAT	0.7159	0.3186	-0.5741
3. Mod	Mengi	0.14	-0.17	0.08
	MSRSP.MAT	0.1374	-0.1713	0.0762
Tüm Modlar	Mengi	3.939	6.979	8.709
	MSRSP.MAT	3.9363	6.9797	8.7112

Tablo incelendiği zaman en büyük katkı birinci moddan gelmektedir, diğer modların katkıları ihmal edilecek kadar küçüktür.

9. SONUÇLAR VE ÖNERİLER

Yapı sistemlerinin dinamik analizi mod süperpozisyon yöntemi kullanılarak bu tez çerçevesinde incelenmiştir. Diferansiyel denklem sistemleri mod süperpozisyon yöntemi ile girişimsiz hale getirilip, denklem sistemlerinin çözümü için; Newmark metodu, Fourier transform metodu ve spektrum analiz yöntemleri kullanılmıştır. Bu üç çözüm yöntemini arasındaki farklar, birbirlerine göre üstünlükleri ve kullanıldığı alanlar incelenmiştir.

Bunların yanı sıra serbest titreşim analizi yapan yöntemler de irdelenmiş olup, kullanıldığı yerler belirlenmeye çalışılmıştır. Yukarıda sözü edilen konularda elde edilen sonuçlar aşağıda maddeler halinde sıralanmıştır.

Transform yöntemlerinin, serbest titreşim analizi sırasında, özdeğer problemi çözümü için pek uygun olmadığına sonucuna varılmıştır. Yapı mekaniği problemleri çoğunlukla çok serbestlik dereceli sistemler olduğundan ve transformasyon yöntemlerinin bütün serbest titreşim karakteristiklerini (serbest titreşim frekansları ve mod şekilleri) bulduğundan dolayı çok fazla zaman almakta ve gerekmediği halde bütün özdeğerleri hesaplamaktadır.

Yapı mekaniği problemlerinin çok serbestlik dereceli sistemler olması ve mod süperpozisyon metodunun özelliğinden dolayı bütün serbest titreşim karakteristiklerinin bulunması yerine, yalnızca en küçük olanların göz önüne alınması yeterli olmaktadır. Bu sebepten ötürü alt uzaylarla iterasyon ve Rayleigh oranı ile iterasyon yönteminin diğerlerine göre daha üstün olduğu sonucu ortaya çıkmıştır. Ayrıca band genişliği küçük sistemler için determinant arama metodunda kullanılabileceği fakat band genişliği yüksek sistemler için alt uzaylarla iterasyon yönteminin daha uygun olduğu ortaya çıkmaktadır.

Tutarlı kütle (consistent mass) kabulü yapılan sistemler için alt uzaylarla iterasyon yöntemi ve Rayleigh oranı ile iterasyon yönteminin daha uygun olduğu sonucuna varılmıştır. Bunun yanı sıra toplanmış kütle (lumped mass) kabulü yapılan sistemlerde ise sistem kütle matrisi genellikle tekil olmaktadır. Bu yüzden toplanmış kütle kabulü yapılan sistemlerde, Rayleigh oranı ile iterasyon yönteminin sturm

teoremi ile birlikte kullanılarak serbest titreşim karakteristiklerinin bulunmasının gerekliliği ortaya çıkmıştır.

Zorlanmış titreşim analizi esnasında, mod süperpozisyon metodu ile girişimsiz halde, sistemin davranışı ikinci mertrebeden adi diferansiyel denklem takımı tarafından idare edilmektedir. Girişimsiz hale getirilmiş olan denklem takımı, Fourier transform metodu ile cebrik denklem takımları ile çözülebilmektedir. Bundan ötürü Fourier transform metodu ile çözüm daha basittir. Bunun yanı sıra, Newmark metodunun tersine Fourier transform metodunda herhangi bir t zamanındaki değeri bulmak için önceden bulunan değerlere ihtiyaç duyulmamaktadır. Ayrıca Newmark metodunda Δt zamanının seçiminin çok önemli olduğu sonucu ortaya çıkmaktadır.

Zorlanmış titreşim analizi sırasında, Newmark yöntemi tüm yükleme tipleri için kullanılmakla beraber, Fourier transform metodunun kullanılabilmesi için hem sistemin sönümlü olması hemde sisteme etkiyen yükün zaman geçtikçe sıfıra gitmesi gerekmektedir. Bu sebepten ötürü, Fourier transform metodunun sadece, zamanla sönümlenen yük ve davranış için uygun olduğu sonucu ortaya çıkmaktadır.

Zorlanmış titreşim analizi sırasında, kesme tipi çerçevelerde mod süperpozisyon yöntemi kullanılırken, yapının tüm modlarından gelen katkıların hesaplanması yerine sadece en küçük birkaç moddan gelen katkının hesaplanmasının, çözülen problemlerde hatalı sonuç vermediği anlaşılmıştır.

Deprem analizi sırasında yapının davranışı incelenirken, mod süperpozisyon yöntemi ile spektrum analizi kullanılması halinde, yine tüm modlar yerine sadece en küçük birkaç modun kullanılmasının hatalı sonuç vermediği ve çok az işlemle yapının davranışının maksimum değerlerinin bulunduğu gözlenmiştir.

KAYNAKLAR

- BATHE, K. J., ve WILSON, E. L., 1976. Numerical Methods in Finite Element Analysis. Prentice-Hall, New Jersey, 528s.
- BATHE, K. J., 1982. Finite Element Procedures in Engineering Analysis. Prentice-Hall, New Jersey, 735s.
- CELEP, Z., ve KUMBASAR, N., 1992. Örneklerle Yapı Dinamiği ve Deprem Mühendisliğine Giriş. Sema Matbaacılık, İstanbul, 422s.
- CLOUGH, R. W., ve PENZIEN, J., 1993. Dynamics of Structures. McGraw-Hill, Singapore, 738s.
- CRAIG, R. R., 1981. Structural Dynamics. John Wiley & Sons, Singapore, 527s.
- DHATT, G., ve TOUZOT, G., 1985. The Finite Element Method Displayed. John Wiley & Sons, Norwich, 509s.
- GHALI, A., ve NEVILLE, A. M., 1984. Structural Analysis. Bizim Büro Basımevi, Ankara, 779s.
- JAMES, M. L., ve SMITH, G. M., Ark., 1989. Vibration of Mechanical and Structural Systems. Harper and Row, New York, 652s.
- KIRAL, E., GÜRKÖK, A., ve MENGİ, Y., 1981. Computing Methods in Engineering Volume I. O.D.T.Ü. Basımevi, Ankara, 330s.
- MENGİ, Y., Structural Dynamics Ders Notları (yayınlanmamış).
- MENGİ, Y., Soil-Structure Interaction Analysis Ders Notları (yayınlanmamış).
- TEZCAN, S., 1970. Çubuk Sistemlerin Elektronik Hesap Makinaları ile Çözümü. Arı Kitabevi Matbası, Ankara, 406s.
- WEAVER, W., ve JOHNSTON, P.R., 1984. Finite Elements For Structural Analysis. Prentice-Hall, New Jersey, 403s.
- WOLFRAM, S., 1991. Mathematica. Addison – Wesley Publishing Co., California, 961s.
- ZAKOUT, U., 1992. A Study of Eigensolution Methods in Structural Engineering. Master Tezi, O.D.T.Ü, Ankara, 124s. (yayınlanmamış).

ÖZGEÇMİŞ

1984 yılında Diyarbakır’da doğdu. İlkokulu ve ortaokulu tamamlandıktan sonra 2002 yılında Diyarbakır Nevzat Ayaz Anadolu Lisesi’den mezun oldu. 2003 Eylülünde Çukurova Üniversitesi Mühendislik- Mimarlık Fakültesi İnşaat Mühendisliği Bölümünde lisans eğitime başlayıp, 2007 Haziranında mezun olarak lisans eğitimini tamamladı. 2007 Eylülde Çukurova Üniversitesi Fen Bilimleri Enstitüsü İnşaat Mühendisliği ana bilim dalında yüksek lisans eğitime başladı. Çeşitli özel şirketlerde saha ve kontrol mühendisi olarak görev yaptıktan sonra 2009 Nisanında Gaziantep Vakıflar Bölge Müdürlüğü Yapı İşleri ve Sanat Eserleri Şube Müdürlüğü’ne İnşaat Mühendisi olarak atandı ve halen bu görevini sürdürmektedir.

EKLER

EK-1. BİLGİSAYAR PROGRAMLARI

E1.1. Mathematica Programları

Mathematica programı ile hazırlanmış olan programlar, kendi düzlemi içinde yüklü yapı sistemlerinin serbest ve zorlanmış titreşim analizlerini yapmaktadır. Serbest titreşim analizi sırasında sistem sönümsüz kabul edilmekte, zorlanmış titreşim analizi sırasında ise sistemin sönümü hesaplara katılmaktadır. Programların çalışması sırasında gerekli bilgilerin çoğu data dosyalarından alınmakla beraber bazı bilgiler ise ekrandan direk olarak verilmektedir.

Hazırlanan 6 adet program aynı işleve sahip olmalarıyla beraber ana hatları ile iki konuda birbirlerinden ayrılırlar. Bunlardan ilki, sistemin serbest titreşim analizi iki farklı metotla yapılmaktadır (Alt uzaylarla iterasyon metodu, Rayleigh oranı ile iterasyon metodu). İkinci farklılık ise, zorlanmış titreşim analizinde mod süper pozisyon yöntemi ile sistem denklemleri ayrıklaştırıldıktan sonra bu denklemlerin çözümü üç farklı metotla yapılmaktadır. (Newmark metodu, Fourier transform metodu, Spektrum analizi).

E1.2. Program İçerikleri ve Listeleri

Yukarıda bahsı geçen 6 adet programın içerikleri ve program listeleri aşağıda görülmektedir.

E1.2.1. MSSN.MAT Programı

Programın işlem basamakları ve listesi aşağıdaki gibidir.

- 1-) Sistem ile ilgili bilgiler data dosyasından alınır.
- 2-) Önce lokal, sonra global koordinat takımında eleman rijitlik ve kütle matrisleri ile yük vektörü oluşturulmaktadır.

- 3-) Kodlama tekniđi yardımı ile sistem rijitlik ve kütle matrisleri ile yük vektörü teşkil edilmektedir.
- 4-) Bilinen serbestlik dereceleri yardımı ile indirgenmiş sistem matrisleri elde edilmektedir.
- 5-) İndirgenmiş sistem matrisleri kullanılarak sistemin serbest titreşim analizi yapılmaktadır. Serbest titreşim analizinde önce alt uzaylarla iterasyon metodu ile özdeğer problemi çözülerek sistemin serbest titreşim frekansları bulunmakta ve sonra ters iterasyon metodu ile özvektörler (mod şekil fonksiyonları) bulunmaktadır.
- 6-) Sturm teoremi yardımı ile bulunan özdeğerlerin sayısı kontrol edilir.
- 7-) Serbest titreşim analizinden sonra, mod süperpozisyon yöntemi ile zorlanmış titreşim analizi yapılmaktadır. Mod süperpozisyon ile girişimsiz hale gelen diferansiyel denklem takımı Newmark metodu ile çözülmekte ve her moda karşılık gelen modal deplasmanlar hesaplanmaktadır.
- 8-) Modal deplasmanlar bulunduktan sonra modların süperpozisyonu yapılarak sistemin esas deplasmanları bulunmaktadır.
- 9-) Son olarak istenen elemanların global koordinat takımında eleman uç kuvvetleri rijitlik matrisi metodu ile hesaplanmaktadır.

MSSN.MAT Program Listesi

```
Clear["Global`*"];
Print["*****"]
Print["***                      MSSN.MAT                      ***"]
Print["***                      Mod-Superpozisyon + Newmark Metodu          ***"]
Print["*** SUBSPACE/JACOBI/STURM CHECK/INVERSE ITERATION ***"]
Print["***                      Düzlemi icinde yuklu duzlemsel cerceveler      ***"]
Print["*****"]

dosya=Input["Data dosyasi adi (Tirnak icinde) = ?"]      ;
file=OpenRead[dosya]                                    ;
Read[file, Hold[Expression]]                             ;
Read[file, Hold[Expression]]                             ;
topelem=Read[file, Number]                               ;
nd=Table[{0.,0.,0.,0.,0.,0.},{topelem}]                  ;
Read[file, Hold[Expression]]                             ;
Do[nd[[i,j]]=Read[file, Number], {i, topelem}, {j, 6}]   ;
Read[file, Hold[Expression]]                             ;
bilsay=Read[file, Number]                               ;
listb=Table[0., {bilsay}]                               ;
Read[file, Hold[Expression]]                             ;
Do[listb[[i]]=Read[file, Number], {i, bilsay}]           ;
tsd=Max[nd]                                              ;
listu=Union[Flatten[nd]]                                ;
lista=Complement[listu, listb]                          ;
bsd=Length[listb]                                       ;
asd=tsd-bsd                                             ;
eds=2                                                  ;
esd=6                                                  ;
(*****)
Print["Elemanlar ile ilgili bilgiler okunuyor"]
Read[file, Hold[Expression]]                             ;
Do [elem=i                                              ;
  Ly[elem]=Read[file, Number]                          ;
  Aty[elem]=Read[file, Number]                          ;
  Aly[elem]=Read[file, Number]                          ;
  Emy[elem]=Read[file, Number]                          ;
  qy[elem]=Read[file, Number]                          ;
  ty[elem]=Read[file, Number]                          ;
  Pdegery[elem]=Read[file, Number]                      ;
  ay[elem]=Read[file, Number]                          ;
  Roy[elem]=Read[file, Number]                          ;
  by[elem]=Ly[elem]-ay[elem]                            ;
  , {i, 1, topelem}]                                    ;
(*****)
```

```

Print["Eleman ve Sistem Matrisleri Olusturuluyor"]
Do[SF[II]=0,{II,1,tsd}] ;
Do[SK[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}] ;
Do[SM[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}] ;

elem=1 ;
(Label [basla] ;
  Print["Eleman no=",elem] ;
  L=Ly[elem] ;
  At=Aty[elem] ;
  Al=Aly[elem] ;
  Em=Emy[elem] ;
  q=qy[elem] ;
  t=ty[elem] ;
  Ro=Roy[elem] ;
  Pdeger=Pdegery[elem] ;
  aa=ay[elem] ;
  bb=by[elem] ;

T[elem]={ {Cos[t],Sin[t],0,0,0,0},
           {-Sin[t],Cos[t],0,0,0,0},
           {0, 0, 1,0,0,0},
           {0,0,0,Cos[t],Sin[t],0},
           {0,0,0,-Sin[t],Cos[t],0},
           {0,0,0, 0, 0, 1}} ;

ke={ {Al*L^2/At,0,0,-Al*L^2/At,0,0},
      {0, 12, 6*L,0,-12,6*L},
      {0, 6*L,4*L^2,0,-6*L,2*L^2},
      {-Al*L^2/At,0,0,Al*L^2/At,0,0},
      {0, -12, -6*L, 0, 12,-6*L},
      {0,6*L,2*L^2,0,-6*L, 4*L^2}}*Em*At/L^3 ;
k[elem]=Transpose[T[elem]].ke.T[elem] ;
kt[elem]=k[elem] ;

me={ {140,0,0, 70,0,0},
      {0, 156,22*L,0,54,-13*L},
      {0,0,22*L,4*L^2,0,13*L,-3*L^2},
      {70,0,0,140,0,0},
      {0,54,13*L,0,156,-22*L},
      {0,-13*L,-3*L^2,0,-22*L,4*L^2}}*Ro*Al*L/420 ;
m[elem]=Transpose[T[elem]].me.T[elem] ;
mt[elem]=m[elem] ;

f3=Pdeger*aa*bb^2/L^2; f6=-Pdeger*bb*aa^2/L^2 ;
feP={ {0},{Pdeger*bb/L+(f3+f6)/L},{f3},{0},{Pdeger*aa/L-(f3+f6)/L},{f6}} ;

```

```

feq={{0},{q*L/2},{q*L^2/12},{0},{q*L/2},{-q*L^2/12}} ;
fe=feP+feq ;
f[elem]=Transpose[T[elem]].fe ;

Do[
II=nd[[elem,id]] ;
JJ=nd[[elem,jd]] ;
SK[II,JJ]=SK[II,JJ]+k[elem][[id,jd]] ;
SM[II,JJ]=SM[II,JJ]+m[elem][[id,jd]] ;
SF[II]=SF[II]+f[elem][[id,1]},{jd,1,esd},{id,1,esd}] ;

elem=elem+1 ;
If [elem<=topelem,Goto[basla],Continue] ) ;
(*****)
Read[file,Hold[Expression]] ;
dys=Read[file,Number] ;
Do[
d=Read[file,Number] ;
Pdeger=Read[file,Number] ;
SF[d]=SF[d]+Pdeger,
{dys}] ;
Read[file,Hold[Expression]] ;
tks=Read[file,Number] ;
Do[
sd=Read[file,Number] ;
Mdeger=Read[file,Number] ;
SM[sd,sd]=SM[sd,sd]+Mdeger,
{tks}] ;

Close[file] ;
(***) Sistem Matrislerinde Indirgeme Uygulanıyor (***)
RSF=Table[{0},{asd}] ;
RSK=IdentityMatrix[asd] ;
RSM=IdentityMatrix[asd] ;
Do[
RSK[[II,JJ]]=0 ;
RSM[[II,JJ]]=0,{II,1,asd},{JJ,1,asd}] ;
Do[
ii=lista[[id]] ;
jj=lista[[jd]] ;
RSK[[id,jd]]=SK[ii,jj] ;
RSM[[id,jd]]=SM[ii,jj] ;
RSF[[id,1]]=SF[ii]},{jd,1,asd},
{id,1,asd}]

```

```

Print["*****"]
Print["***** Alt Uzaylarda Iterasyon *****"]
Print["*****          +          *****"]
Print["*****          Genel Jacobi Metodu ile          *****"]
Print["*****          Ozel Deger Hesabi          *****"]
Print["*****"]

```

```

Clear[m,f];Clear[SK,SM,SF]
k=RSK;m=RSM ;
k1[1]=k; m1[1]=m ;
n=Dimensions[k1[1]][[1]] ;
Print["Sistem Serbestlik Derecesi=",n] ;
mod=Input["Istenen Mod Sayisi="] ;
q=mod ;
listmod=Table[0.,{q}] ;
list=Table[listmod,{n}] ;
Do[list[[i,j]]=Random[ ],{j,1,q},{i,1,n}] ;
x[1]=list ;
y[1]=m1[1].x[1] ;

iter=10 ;
Do[Print["Iterasyon No=",iii] ;
  yt=Transpose[y[iii]] ;
  Do[a[i]=LinearSolve[k1[1],yt[[i]],{i,q}] ;
  xb[iii+1]=Transpose[Table[a[i],{i,q}]] ;
  k1[iii+1]=Transpose[xb[iii+1]].y[iii] ;
  yb[iii+1]=m1[1].xb[iii+1] ;
  m1[iii+1]=Transpose[xb[iii+1]].yb[iii+1] ;
  (** Jacobi Metodu ile Alt Uzayda Ozdeger Cozumu **) ;
  k=k1[iii+1] ;
  m=m1[iii+1] ;
  n=Dimensions[k][[1]] ;
  ts=5 ;
  fi=IdentityMatrix[n] ;
  kk=IdentityMatrix[n] ;
  k0=k ;
  m0=m ;
  Do[
    Do[kk[[i,i]]=k[[i,i]] m[[i,j]]-m[[i,i]] k[[i,j]] ;
    kk[[j,j]]=k[[j,j]] m[[i,j]]-m[[j,j]] k[[i,j]] ;
    kb=k[[i,i]] m[[j,j]]-k[[j,j]] m[[i,i]] ;
    If[kb==0,sign=1,sign=Sign[kb]] ;
    xx=kb/2.+sign*Sqrt[(kb/2.)^2+kk[[i,i]]*kk[[j,j]]] ;
    gama=-kk[[i,i]]/xx ; alfa=kk[[j,j]]/xx ;
    p=IdentityMatrix[n] ;
    p[[i,i]]=1. ;
  ]

```

```

p[[i,j]]=alfa ;
p[[j,i]]=gama ;
p[[j,j]]=1. ;
fi=fi.p ;
pt=Transpose[p] ;
k=pt.k.p ;m=pt.m.p,{i,1,n-1},{j,i+1,n} , {ii,1,ts} ;
(*****) ;
mi=IdentityMatrix[n] ;
Do[mi[[i,i]]=1/Sqrt[m[[i,i]]], {i,n}] ;
fi=fi.mi ;
y[iii+1]=yb[iii+1].fi ,{iii,1,iter}] ;
x[iter+1]=xb[iter+1].fi ;
lamda=IdentityMatrix[n] ;
Do[lamda[[i,i]]=k[[i,i]]/m[[i,i]],{i,n}] ;
Print["*****"]
Print["**** Ozel Degerler ****"]
Print["*****"]
lam=Table[1,{i,n}] ;
Do[lam[[i]]=lamda[[i,i]],{i,n}] ;
lam=Sort[lam] ;
Do[Print[Chop[lam[[i]]],{i,n}] ;

Print["*****"]
Print["**** Sturm Dizisi Kontrolü ****"]
Print["*****"]
n=Dimensions[k1[1]] [[1]] ;
n1=Dimensions[k][[1]] ;
listlam=Table[lamda[[i,i]],{i,n1}] ;
mu=Max[listlam] ;
mu=mu+mu/100 ;
k=k1[1]-mu m1[1] ;
g=IdentityMatrix[n] ;
l=IdentityMatrix[n] ;
d=IdentityMatrix[n] ;
d[[1,1]]=k[[1,1]] ;
(***) Kolon Yükseklikleri Hesaplanıyor (***)
mm[1]=1 ;
Do[
Do[If[k[[i,j]]!=0., (mm[j]=i;Return[mm[j]]),Continue],{i,1,n}] ;
,{j,2,n}]

(**** L.D.U seklinde carpanlara ayirma ****)
Do[
g[[mm[j],j]]=k[[mm[j],j]] ;
Do[g[[i,j]]=k[[i,j]]-Sum[l[[r,i]]*g[[r,j]],{r,Max[mm[i],mm[j]],(i-1)}] ;
,{i,(mm[j]+1),(j-1)}] ;

```

```

Do[l[[i,j]]=g[[i,j]]/d[[i,i]],{i,mm[j],(j-1)}} ;
d[[j,j]]=k[[j,j]]-Sum[l[[r,j]]*g[[r,j]],{r,mm[j],(j-1)}} ;
, {j,2,n}]

say=0 ;
Do[If[d[[i,i]]<0, say=say+1,Continue],{i,1,n}] ;
If[say==q,Continue,(Print["Arada ozel deger var  !!"];Pause[5])] ;
sortlist=Sort[listlam]
Print["*****"]
Print["**** Ters Iterasyon ile Ozvektorler Hesaplaniyor ****"]
Print["*****"]
Do[
mu=sortlist[[nn]] ;
Print["Mu=",mu] ;
v1[1]=Table[Random[ ],{n}] ;
iter=5 ;
ks=RSK- mu RSM ;
Do[
mx=RSM.v1[i] ;
xb=LinearSolve[ks,mx] ;
payda=Sqrt[xb.RSM.xb] ;
v1[i+1]=xb/payda ;
ro[i+1]=(xb.RSK.xb)/(xb.RSM.xb)+mu ;
, {i,1,iter}] ;
omeg[nn]=N[ro[iter]-mu] ;
v[nn]=N[Chop[v1[iter]]] ;
a[nn]=v[nn] ;
, {nn,1,mod}]

Do[omeg[i]=Sqrt[omeg[i]],{i,mod}]
Do[Print[omeg[i]/2./N[Pi]],{i,mod}]
Do[RSFB[i]=a[i].RSF,{i,mod}]

Print["*****"]
Print["**** Newmark Yontemi ile Dif. Denk. Takimi Cozumu ****"]
Print["**** Y"+c Y'+k Y=F/m ****"]
Print["*****"]
Clear[f,m]
(*
Print["Harmonik yukleme"]
P0=2.0
Omega=62.9
ft[t_]:=P0*Cos[Omega*t] *)

Print["Adim Tipi Yukleme"]
P0=10. ;
ft[t_]:=P0 ;

```

```

(*)
Print["Ucgen Yukleme"]
t1=1.
t2=t1/2.
P0=1.0
ft[t_]:=Which[(t>=0.&& t<t2),P0*t/t2,(t>=t2&& t<=t1),
               P0*(t-t1)/(t2-t1),t>t1,0.0]      *)

dt=0.005                                     ;
tson=0.2                                     ;

(* pdata=Table[{t,ft[t]},{t,0.,2*tson,dt}]
p1=ListPlot[N[pdata],PlotJoined->True,AxesLabel->{"Zaman","Kuvvet"},
             PlotStyle->RGBColor[1,0,0],GridLines->Automatic,
             PlotRange->{-3.*P0,3.*P0},PlotLabel->"Zaman Uzayi"] *)

z=Input["Sonum Orani= ?"]                    ;

Do[ksi[i]=z, {i,mod}]                        ;

Do[Print["Mod=",imod]                        ;
m=1                                           ;
c=2*ksi[imod]*omeg[imod]                     ;
kk=omeg[imod]^2                             ;
ud0=0. (*baslangic deplasmani*)             ;
ud1=0. (*baslangic hizi*)                   ;
alfa=0.50;beta=0.25                         ;
t=0.                                          ;
nsay=1                                       ;
ud2=(1/m)*(ft[t]*RSFB[imod]-kk*ud0-c*ud1)   ;
dt2=dt*dt                                   ;
a0=1./beta/dt2                              ;
a1=alfa/dt/beta                             ;
a2=a0*dt                                    ;
a3=0.5/beta-1.                             ;
a4=alfa/beta-1.                             ;
a5=dt*(alfa/2./beta-1.)                    ;
a6=dt*(1.-alfa)                            ;
a7=alfa*dt                                  ;
bk=kk+a0*m+a1*c                             ;
yf[imod,nsay]=ud0                           ;
yf2[imod,nsay]=ud2[[1]]                    ;
(Label[newmark]                             ;
mmm=m*(a0*ud0+a2*ud1+a3*ud2)               ;
ccc=c*(a1*ud0+a4*ud1+a5*ud2)               ;
bf=ft[t]*RSFB[imod]+mmm+ccc                ;

```

```

u=(1./bk)*bf ;
ud2p=a0*(u-ud0)-a2*ud1-a3*ud2 ;
ud1=ud1+a6*ud2+a7*ud2p ;
ud2=ud2p ;
ud0=u ;
t=t+dt ;
nsay=nsay+1 ;
yf[imod,nsay]=u[[1]] ;
yf2[imod,nsay]=ud2[[1]] ;
If[t<=tson,Goto[newmark],Continue)) ;
      ,{imod,1,mod}}

Print["*****"]
Print["*** Modların Superpozisyonu Yapılıyor ***"]
Print["*****"]
Do[fx[i_,t_]:=Sum[a[j][[i]]*yf[j,Round[(t/dt)+1]],{j,mod}],{i,asd}] ;
Do[fx2[i_,t_]:=Sum[a[j][[i]]*yf2[j,Round[(t/dt)+1]],{j,mod}],{i,asd}] ;
t=. ;
Do[dept[i]=Table[{t,fx[i,t]},{t,0.,tson,dt}],{i,asd}] ;

Print["*****"]
Print["*** Deplasman ve ivme değerleri hesaplanıyor ***"]
Print["*****"]
Do[dep[i]=Table[fx[i,t],{t,0.,tson,dt}],{i,asd}] ;
Do[dep2[i]=Table[fx2[i,t],{t,0.,tson,dt}],{i,asd}] ;
dof=Input["Çizilmesi İstenen Serbestlik Derecesi ="] ;
pos=Position[lista,dof][[1,1]] ;
plotdep=ListPlot[dept[pos],PlotJoined->True,PlotStyle->RGBColor[0,0,1],
  PlotRange->All,Frame->True,FrameLabel->{"Zaman","Deplasman"},
  GridLines->Automatic,PlotLabel->"Deplasman Ergisi"]
d=N[Max[Abs[dep[pos]]]] ;
Print["Deplasmanın max. değeri =",d] ;
post=Position[Abs[dep[pos]],d] ;
tmax=dept[pos][[post[[1,1]],1]] ;
Print[" Bu andaki zaman,tmax =",tmax] ;
Print["*****"]
Print["*** Eleman Üç Kuvvetleri Hesabı ***"]
Print["*****"]
topadim=tson/dt ;
elem=Input["Eleman Numarası ="] ;
tt=0. ;
msay=1 ;
Do[
  Do[ii=listb[[i]];up[ii]=0.,{i,bsd}] ;
  Do[ii=listb[[i]];uc[ii,adim]=0.,{i,bsd}] ;

```

```

Do[ii=listb[[i]];up2[ii]=0.,{i,bsd}] ;
Do[ii=listb[[i]];uc2[ii,adim]=0.,{i,bsd}] ;
Do[ii=lista[[i]];uc[ii,adim]=dep[i][[adim]],{i,asd}] ;
Do[ii=lista[[i]];uc2[ii,adim]=dep2[i][[adim]],{i,asd}] ;
Do[up[i]=uc[nd[[elem,i]],adim],{i,1,esd}] ;
Do[up2[i]=uc2[nd[[elem,i]],adim],{i,1,esd}] ;
Do[pp[i]=Sum[kt[elem][[i,j]]*up[j],{j,esd}],{i,esd}] ;
Do[ppa[i]=Sum[mt[elem][[i,j]]*up2[j],{j,esd}],{i,esd}] ;
(*** Sistem Koordinatlarinda Eleman Kuvvetleri ***)
ptk[msay]=pp[5]+ppa[5] ;
tt=tt+dt ;
msay=msay+1 , {adim, topadim+1}] ;

ptt=Table[{tt,ptk[Round[(tt/dt)+1]]},{tt,0.,tson,dt}] ;
ptm=Table[ptk[Round[(tt/dt)+1]],{tt,0.,tson,dt}] ;
plot=ListPlot[ptt,PlotJoined->True,PlotStyle->RGBColor[1,0,0],
PlotRange->All,PlotLabel->{elem,". elemana ait"},Frame->
True,GridLines->Automatic,FrameLabel->{"Zaman","Kesme"}]
d=Max[Abs[ptm]] ;
d=N[d] ;
Print["Kesme Maksimum Degeri =", d ] ;
pos=Position[Abs[ptm],d] ;
tmax=ptt[[pos[[1,1]],1]] ;
Print["Bu Andaki Zaman ,t =" , tmax] ;

```

E1.2.2. MSRN.MAT Programı

Programın işlem basamakları ve listesi aşağıdaki gibidir.

- 1-) Sistem ile ilgili bilgiler data dosyasından alınır.
- 2-) Önce lokal, sonra global koordinat takımında eleman rijitlik ve kütle matrisleri ile yük vektörü oluşturulmaktadır.
- 3-) Kodlama tekniği yardımı ile sistem rijitlik ve kütle matrisleri ile yük vektörü teşkil edilmektedir.
- 4-) Bilinen serbestlik dereceleri yardımı ile indirgenmiş sistem matrisleri elde edilmektedir.
- 5-) Sistemin serbest titreşim analizi Rayleigh oranı ile iterasyon metodu kullanılarak yapılmaktadır. Ayrıca bu metotla birlikte Gram-Schmidt ortogonalizasyonu kullanılmaktadır.
- 6-) Sturm teoremi yardımı ile bulunan özdeğerlerin sayısı kontrol edilir.
- 7-) Serbest titreşim analizinden sonra, mod süperpozisyon yöntemi ile zorlanmış titreşim analizi yapılmaktadır. Mod süperpozisyon ile girişimsiz hale gelen diferansiyel denklem takımı Newmark metodu ile çözülmekte ve her moda karşılık gelen modal deplasmanlar hesaplanmaktadır.
- 8-) Modal deplasmanlar bulunduktan sonra modların süperpozisyonu yapılarak sistemin esas deplasmanları bulunmektedir.
- 9-) Son olarak istenen elemanların global koordinat takımında eleman uç kuvvetleri rijitlik matrisi metodu ile hesaplanmaktadır.

MSRN.MAT Program Listesi

```
Clear["Global`*"];
Print["*****"]
Print["***          MSRN.MAT          ***"]
Print["***   Mod-Superpozisyon + Newmark Metodu   ***"]
Print["***          Rayleigh & Gram-Schmidt & Shift          ***"]
Print["***   Duzlemi icinde yuklu duzlemsel cerceveler   ***"]
Print["*****"]
dosya=Input [ "Data dosyasi adi ( Tirnak icinde ) = ?" ]
file=OpenRead [ dosya ]
Read[file,Hold[Expression]]
Read[file,Hold[Expression]]
topelem=Read[file,Number]
nd=Table[{0.,0.,0.,0.,0.,0.}, {topelem}]
Read[file,Hold[Expression]]
Do[nd[[i,j]]=Read[file,Number],{i,topelem},{j,6}]
Read[file,Hold[Expression]]
bilsay=Read[file,Number]
listb=Table[0.,{bilsay}]
Read[file,Hold[Expression]]
Do[listb[[i]]=Read[file,Number],{i,bilsay}]
tsd=Max[nd]
listu=Union[Flatten[nd]]
lista=Complement[listu,listb]
bsd=Length[listb]
asd=tsd-bsd
eds=2
esd=6
(*****)
Read[file,Hold[Expression]]
For [elem=1, elem <=topelem, elem++,
    Ly[elem]=Read[file, Number]          ;
    Aty[elem]=Read[file, Number]         ;
    Aly[elem]=Read[file, Number]         ;
    Emy[elem]=Read[file, Number]         ;
    qy[elem]=Read[file, Number]          ;
    ty[elem]=Read[file, Number]          ;
    Pdegery[elem]=Read[file, Number]     ;
    ay[elem]=Read[file, Number]          ;
    Roy[elem]=Read[file, Number]         ;
    by[elem]=Ly[elem]-ay[elem]   ]

(*****)
Print[" Sistem Matrisleri Olusturuluyor"]
Do[SF[II]=0,{II,1,tsd}]
Do[SK[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]
```

Do[SM[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]

elem=1

```
(Label [basla] ;
  Print["Eleman no=",elem] ;
  L=Ly[elem] ;
  At=Aty[elem] ;
  Al=Aly[elem] ;
  Em=Emy[elem] ;
  q=qy[elem] ;
  t=ty[elem] ;
  Ro=Roy[elem] ;
  Pdeger=Pdegery[elem] ;
  aa=ay[elem] ;
  bb=by[elem] ;
```

```
T[elem]={ {Cos[t],Sin[t],0,0,0,0},
           {-Sin[t],Cos[t],0,0,0,0},
           {0, 0, 1,0,0,0},
           {0,0,0,Cos[t],Sin[t],0},
           {0,0,0,-Sin[t],Cos[t],0},
           {0,0,0, 0, 0, 1}} ;
```

```
ke={ {Al*L^2/At,0,0,-Al*L^2/At,0,0},
      {0, 12, 6*L,0,-12,6*L},
      {0, 6*L,4*L^2,0,-6*L,2*L^2},
      {-Al*L^2/At,0,0,Al*L^2/At,0,0},
      {0, -12, -6*L, 0, 12,-6*L},
      {0,6*L,2*L^2,0,-6*L, 4*L^2}}*Em*At/L^3 ;
k[elem]=Transpose[T[elem]].ke.T[elem] ;
kt[elem]=k[elem] ;
```

```
me={ {140,0,0, 70,0,0},
      {0, 156,22*L,0,54,-13*L},
      {0,0,22*L,4*L^2,0,13*L,-3*L^2},
      {70,0,0,140,0,0},
      {0,54,13*L,0,156,-22*L},
      {0,-13*L,-3*L^2,0,-22*L,4*L^2}}*Ro*Al*L/420 ;
m[elem]=Transpose[T[elem]].me.T[elem] ;
mt[elem]=m[elem] ;
```

```
f3=Pdeger*aa*bb^2/L^2; f6=-Pdeger*bb*aa^2/L^2 ;
feP={ {0},{Pdeger*bb/L+(f3+f6)/L},{f3},{0},{Pdeger*aa/L-(f3+f6)/L},{f6}} ;
feq={ {0},{q*L/2},{q*L^2/12},{0},{q*L/2},{-q*L^2/12}} ;
fe=feP+feq ;
f[elem]=Transpose[T[elem]].fe ;
```

```

Do[
II=nd[[elem,id]]
JJ=nd[[elem,jd]]
SK[II,JJ]=SK[II,JJ]+k[elem][[id,jd]]
SM[II,JJ]=SM[II,JJ]+m[elem][[id,jd]]
SF[II]=SF[II]+f[elem][[id,1]],{jd,1,esd},{id,1,esd}]

elem=elem+1
If [elem<=topelem,Goto[basla],Continue]
(*****)
Read[file,Hold[Expression]]
dys=Read[file,Number]
Do[
d=Read[file,Number]
Pdeger=Read[file,Number]
SF[d]=SF[d]+Pdeger,
{dys}]
Read[file,Hold[Expression]]
tks=Read[file,Number]
Do[
sd=Read[file,Number]
Mdeger=Read[file,Number]
SM[sd,sd]=SM[sd,sd]+Mdeger,
{tks}]
Close[file]
(*** Sistem Matrislerinde Indirgeme Uygulanıyor ***)
RSF=Table[{0},{asd}]
RSK=IdentityMatrix[asd]
RSM=IdentityMatrix[asd]
Do[
RSK[[II,JJ]]=0
RSM[[II,JJ]]=0,{II,1,asd},{JJ,1,asd}]
Do[
ii=lista[[id]]
jj=lista[[jd]]
RSK[[id,jd]]=SK[ii,jj]
RSM[[id,jd]]=SM[ii,jj]
RSF[[id,1]]=SF[ii,{jd,1,asd},
{id,1,asd}]

Print["*****"]
Print["***** Rayleigh & Gram-Schmidt *****"]
Print["***** Ozel Deger Hesabi *****"]
Print["*****"]

```

```

n=Dimensions[RSK][[1]]
Print["Toplam Serbestlik Derecesi=",n]
mod=Input["Istenen Mod Sayisi="]
mod1=mod-1
sayac=0
(Label [Rayleigh] ;mod1+=1
sayac+=1
v[1]=Table[Random [ ],{n}]
v1=v[1]
ro[1]= 0.0
tol=10^(-6)
v[i+1]=v[1]
Print["Deneme Sayisi = " ,sayac]
Do[
fi[eigs]=v[i+1]
alfa[eigs]=(fi[eigs].RSM.v1)
v[1]=v1-Sum[alfa[ii] fi[ii] , {ii,2,eigs}]
i=0
say=0
( Label[basla2]
i=i+1
ks=RSK-ro[i]*RSM
mx=RSM.v[i]
xb=LinearSolve[ks,mx]
payda=Sqrt [(xb.RSM.xb)]
v[i+1]=xb/payda
ro[i+1]=(xb.ks.xb)/(xb.RSM.xb)+ro[i]
kontrol=Abs[ro[i+1]-ro[i]]/Abs[ro[i+1]]
If [kontrol<tol,Continue,Goto [basla2]] )
omeg[eigs]=N[ro[i+1]]
Print[" Ozel deger (" ,eigs,")= " ,N[ro[i+1]]]
a[eigs]=v[i+1] , {eigs,1,mod1}]

Print["*****"]
Print["***** Sturm Dizisi Kontrolü *****"]
Print["*****"]
listlam=Table[omeg[i],{i,1,mod1}]
lislml=Sort[listlam]
mu=lislml[[mod]]
mu=mu+mu/100
kkk=RSK-mu RSM
g=IdentityMatrix[n]
l=IdentityMatrix[n]
d=IdentityMatrix[n]

```

```

d[[1,1]]=kkk[[1,1]] ;
(***) Kolon Yükseklikleri Hesaplanıyor (***)
mm[1]=1
Do[
Do[If[kkk[[i,j]]!=0., (mm[j]=i;Return[mm[j]]),Continue],{i,1,n}] ;
, {j,2,n}] ;
(**** L.D.U seklinde carpanlara ayirma ****)
Do[
g[[mm[j],j]]=kkk[[mm[j],j]] ;
Do[g[[i,j]]=kkk[[i,j]]-Sum[l[[r,i]]*g[[r,j]],{r,Max[mm[i],mm[j]],(i-1)}]
, {i,(mm[j]+1),(j-1)}] ;
Do[l[[i,j]]=g[[i,j]]/d[[i,i]],{i,mm[j],(j-1)}] ;
d[[j,j]]=kkk[[j,j]]-Sum[l[[r,j]]*g[[r,j]],{r,mm[j],(j-1)}] , {j,2,n}] ;
say=0 ;
Do[If[d[[i,i]]<0, say=say+1,Continue],{i,1,n}] ;
If[say==mod,Continue,Goto[Rayleigh]] )

Print[" Siraya Konmus Ozel Degerler "]
list=Table[omeg[i], {i,mod}]
Print[Sort [list]]
Do[omeg[i]=Sqrt[omeg[i]],{i,mod}]
Print["Serbest Titreşim Frekanslari (w,Rad/Sn)"]
Print["*****"]
Do[Print[omeg[i]/2./N[Pi]],{i,mod}]
Do[RSFB[i]=a[i].RSF,{i,mod}]

Print["*****"]
Print["*** Newmark Yontemi ile Dif. Denk. Takimi Cozumu ***"]
Print["*** Y''+c Y'+k Y=F/m ***"]
Print["*****"]
Clear[f,m]
(*
Print["Harmonik yukleme"]
P0=2.0
Omega=62.9
ft[t_]:=P0*Cos[Omega*t] *)

Print["Adim Tipi Yukleme"]
P0=10.
ft[t_]:=P0
(*
Print["Ucgen Yukleme"]
t1=1.
t2=t1/2.
P0=1.0
ft[t_]:=Which[(t>=0.&&t<t2),P0*t/t2,(t>=t2&&t<=t1),

```

```
P0*(t-t1)/(t2-t1),t>t1,0.0] *)
dt=0.01
tson=3.
```

```
(* pdata=Table[{t,ft[t]},{t,0.,2*tson,dt}]
p1=ListPlot[N[pdata],PlotJoined->True,AxesLabel->{"Zaman","Kuvvet"},
PlotStyle->RGBColor[1,0,0],GridLines->Automatic,
PlotRange->{-3.*P0,3.*P0},PlotLabel->"Zaman Uzayi"] *)
```

```
z=Input["Sonum Orani= ?"] ;
Do[ksi[i]=z, {i,mod}] ;
Do[Print["Mod=",imod] ;
m=1 ;
c=2*ksi[imod]*omeg[imod] ;
kk=omeg[imod]^2 ;
ud0=0. (*baslangic deplasmani*) ;
ud1=0. (*baslangic hizi*) ;
alfa=0.50;beta=0.25 ;
t=0. ;
nsay=1 ;
ud2=(1/m)*(ft[t]*RSFB[imod]-kk*ud0-c*ud1) ;
dt2=dt*dt ;
a0=1./beta/dt2 ;
a1=alfa/dt/beta ;
a2=a0*dt ;
a3=0.5/beta-1. ;
a4=alfa/beta-1. ;
a5=dt*(alfa/2./beta-1.) ;
a6=dt*(1.-alfa) ;
a7=alfa*dt ;
bk=kk+a0*m+a1*c ;
yf[imod,nsay]=ud0 ;
yf2[imod,nsay]=ud2[[1]] ;
(Label[newmark] ;
mmm=m*(a0*ud0+a2*ud1+a3*ud2) ;
ccc=c*(a1*ud0+a4*ud1+a5*ud2) ;
bf=ft[t]*RSFB[imod]+mmm+ccc ;
u=(1./bk)*bf ;
ud2p=a0*(u-ud0)-a2*ud1-a3*ud2 ;
ud1=ud1+a6*ud2+a7*ud2p ;
ud2=ud2p ;
ud0=u ;
t=t+dt ;
nsay=nsay+1 ;
yf[imod,nsay]=u[[1]] ;
yf2[imod,nsay]=ud2[[1]] ;
```

```

If[t<=tson,Goto[newmark],Continue])                                     ;
    ,{imod,1,mod}}
Print["*****"]
Print["***   Modların Superpozisyonu Yapılıyor   ***"]
Print["*****"]
Do[fx[i_,t_]:=Sum[a[j][[i]]*yf[j,Round[(t/dt)+1]],{j,mod}],{i,asd}]    ;
Do[fx2[i_,t_]:=Sum[a[j][[i]]*yf2[j,Round[(t/dt)+1]],{j,mod}],{i,asd}]  ;
t=.                                                                    ;
Do[dept[i]=Table[{t,fx[i,t]},{t,0.,tson,dt}],{i,asd}]                ;

Print["*****"]
Print["***   Deplasman ve ivme değerleri hesaplanıyor   ***"]
Print["*****"]
Do[dep[i]=Table[fx[i,t],{t,0.,tson,dt}],{i,asd}]                      ;
Do[dep2[i]=Table[fx2[i,t],{t,0.,tson,dt}],{i,asd}]                    ;
dof=Input["Çizilmesi İstenen Serbestlik Derecesi ="]                  ;
pos=Position[lista,dof][[1,1]]                                         ;
plotdep=ListPlot[dept[pos],PlotJoined->True,PlotStyle->RGBColor[0,0,1],
    PlotRange->All,Frame->True,FrameLabel->{"Zaman","Deplasman"},
    GridLines->Automatic,PlotLabel->"Deplasman Ergisi"]
d=N[Max[Abs[dep[pos]]]]                                                ;
Print["Deplasmanın max. değeri =",d]                                    ;
post=Position[Abs[dep[pos]],d]                                          ;
tmax=dept[pos][[post[[1,1]],1]]                                         ;
Print[" Bu andaki zaman,tmax =",tmax]                                    ;
Print["*****"]
Print["***   Eleman Üç Kuvvetleri Hesabı   ***"]
Print["*****"]
topadim=tson/dt                                                         ;
elem=Input["Eleman Numarası ="]                                         ;
tt=0.                                                                    ;
msay=1                                                                    ;
Do[
Do[ii=listb[[i]];up[ii]=0.,{i,bsd}]                                     ;
Do[ii=listb[[i]];uc[ii,adim]=0.,{i,bsd}]                                 ;
Do[ii=listb[[i]];up2[ii]=0.,{i,bsd}]                                     ;
Do[ii=listb[[i]];uc2[ii,adim]=0.,{i,bsd}]                                 ;
Do[ii=lista[[i]];uc[ii,adim]=dep[i][[adim]],{i,asd}]                   ;
Do[ii=lista[[i]];uc2[ii,adim]=dep2[i][[adim]],{i,asd}]                 ;
Do[up[i]=uc[nd[[elem,i]],adim],{i,1,esd}]                             ;
Do[up2[i]=uc2[nd[[elem,i]],adim],{i,1,esd}]                             ;
Do[pp[i]=Sum[kt[elem][[i,j]]*up[j],{j,esd}],{i,esd}]                 ;
Do[ppa[i]=Sum[mt[elem][[i,j]]*up2[j],{j,esd}],{i,esd}]               ;
(*** Sistem Koordinatlarında Eleman Kuvvetleri ***)
ptk[msay]=pp[5]+ppa[5]                                                 ;
tt=tt+dt                                                                ;

```

```

msay=msay+1      , {adim, topadim+1}]      ;
ptt=Table[{tt,ptk[Round[(tt/dt)+1]]},{tt,0.,tson,dt}]      ;
ptm=Table[ptk[Round[(tt/dt)+1]],{tt,0.,tson,dt}]      ;
plot=ListPlot[ptt,PlotJoined->True,PlotStyle->RGBColor[1,0,0],
      PlotRange->All,PlotLabel->{elem,". elemana ait"},Frame->
      True,GridLines->Automatic,FrameLabel->{"Zaman","Kesme"}]
d=Max[Abs[ptm]]      ;
d=N[d]      ;
Print["Kesme Maksimum Degeri =", d ]      ;
pos=Position[Abs[ptm],d]      ;
tmax=ptt[[pos[[1,1]],1]]      ;
Print["Bu Andaki Zaman  ,t =" , tmax]      ;

```

E1.2.3. MSSFT.MAT Programı

Programın işlem basamakları ve listesi aşağıdaki gibidir.

- 1-) Sistem ile ilgili bilgiler data dosyasından alınır.
- 2-) Önce lokal, sonra global koordinat takımında eleman rijitlik ve kütle matrisleri ile yük vektörü oluşturulmaktadır.
- 3-) Kodlama tekniği yardımı ile sistem rijitlik ve kütle matrisleri ile yük vektörü teşkil edilmektedir.
- 4-) Bilinen serbestlik dereceleri yardımı ile indirgenmiş sistem matrisleri elde edilmektedir.
- 5-) İndirgenmiş sistem matrisleri kullanılarak sistemin serbest titreşim analizi yapılmaktadır. Serbest titreşim analizinde önce alt uzaylarla iterasyon metodu ile özdeğer problemi çözülerek sistemin serbest titreşim frekansları bulunmakta ve sonra ters iterasyon metodu ile özvektörler (mod şekil fonksiyonları) bulunmaktadır.
- 6-) Sturm teoremi yardımı ile bulunan özdeğerlerin sayısı kontrol edilir.
- 7-) Serbest titreşim analizinden sonra, mod süperpozisyon yöntemi ile zorlanmış titreşim analizi yapılmaktadır. Mod süperpozisyon ile girişimsiz hale gelen diferansiyel denklem takımı elde edildikten sonra, her mod için fourier uzayında ayrık fourier transform metodu ile modal deplasmanlar bulunmaktadır. Daha sonra ters fourier transform algoritması ile zaman uzayında modal deplasmanlar elde edilmektedir.
- 8-) Modal deplasmanlar bulunduktan sonra modların süperpozisyonu yapılarak sistemin esas deplasmanları bulunmaktadır.
- 9-) Son olarak istenen elemanların global koordinat takımında eleman uç kuvvetleri rijitlik matrisi metodu ile hesaplanmaktadır.

MSSFT.MAT Program Listesi

```
Print["*****"]
Print["***                      MSSFT.MAT                      ***"]
Print["***                      Mod-Superpozisyon + Fourier Transform Metodu                      ***"]
Print["*** SUBSPACE/JACOBI/STURM CHECK/INVERSE ITERATION ***"]
Print["***          Düzlemi içinde yuklu düzlemsel çerçeveler          ***"]
Print["*****"]
dosya=Input [ "Data dosyasi adi ( Tirnak içinde ) = ?" ]
file=OpenRead [ dosya ]
Read[file,Hold[Expression]]
Read[file,Hold[Expression]]
topelem=Read[file,Number]
nd=Table[{0.,0.,0.,0.,0.,0.}, {topelem}]
Read[file,Hold[Expression]]
Do[nd[[i,j]]=Read[file,Number],{i,topelem},{j,6}]
Read[file,Hold[Expression]]
bilsay=Read[file,Number]
listb=Table[0.,{bilsay}]
Read[file,Hold[Expression]]
Do[listb[[i]]=Read[file,Number],{i,bilsay}]
tsd=Max[nd]
listu=Union[Flatten[nd]]
lista=Complement[listu,listb]
bsd=Length[listb]
asd=tsd-bsd
eds=2
esd=6
(*****)
Print["Elemanlar ile ilgili bilgiler okunuyor"]
Read[file,Hold[Expression]]
Do [elem=i                                     ;
    Ly[elem]=Read[ file, Number]               ;
    Aty[elem]=Read[ file, Number]              ;
    Aly[elem]=Read[ file, Number]              ;
    Emy[elem]=Read[ file, Number]              ;
    qy[elem]=Read[ file, Number]               ;
    ty[elem]=Read[ file, Number]               ;
    Pdegery[elem]=Read[ file, Number]          ;
    ay[elem]=Read[ file, Number]               ;
    Roy[elem]=Read[ file, Number]              ;
    by[elem]=Ly[elem]-ay[elem]                 ;
    ,{i,1,topelem}]
(*****)
```

```

Print["Eleman ve Sistem Matrisleri Olusturuluyor"]
Do[SF[II]=0,{II,1,tsd}]
Do[SK[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]
Do[SM[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]

elem=1
(Label [basla] ;
  Print["Eleman no=",elem] ;
  L=Ly[elem] ;
  At=Aty[elem] ;
  Al=Aly[elem] ;
  Em=Emy[elem] ;
  q=qy[elem] ;
  t=ty[elem] ;
  Ro=Roy[elem] ;
  Pdeger=Pdegery[elem] ;
  aa=ay[elem] ;
  bb=by[elem] ;

T[elem]={ {Cos[t],Sin[t],0,0,0,0},
           {-Sin[t],Cos[t],0,0,0,0},
           {0, 0, 1,0,0,0},
           {0,0,0,Cos[t],Sin[t],0},
           {0,0,0,-Sin[t],Cos[t],0},
           {0,0,0, 0, 0, 1}} ;

ke={ {Al*L^2/At,0,0,-Al*L^2/At,0,0},
      {0, 12, 6*L,0,-12,6*L},
      {0, 6*L,4*L^2,0,-6*L,2*L^2},
      {-Al*L^2/At,0,0,Al*L^2/At,0,0},
      {0, -12, -6*L, 0, 12,-6*L},
      {0,6*L,2*L^2,0,-6*L, 4*L^2}}*Em*At/L^3 ;
k[elem]=Transpose[T[elem]].ke.T[elem] ;
kt[elem]=k[elem] ;

me={ {140,0,0, 70,0,0},
      {0, 156,22*L,0,54,-13*L},
      {0,022*L,4*L^2,0,13*L,-3*L^2},
      {70,0,0,140,0,0},
      {0,54,13*L,0,156,-22*L},
      {0,-13*L,-3*L^2,0,-22*L,4*L^2}}*Ro*Al*L/420 ;
m[elem]=Transpose[T[elem]].me.T[elem] ;
mt[elem]=m[elem] ;
f3=Pdeger*aa*bb^2/L^2; f6=-Pdeger*bb*aa^2/L^2 ;
feP={ {0},{Pdeger*bb/L+(f3+f6)/L},{f3},{0},{Pdeger*aa/L-(f3+f6)/L},{f6}} ;
feq={ {0},{q*L/2},{q*L^2/12},{0},{q*L/2},{-q*L^2/12}} ;

```

```

fe=feP+feq ;
f[elem]=Transpose[T[elem]].fe ;

Do[
II=nd[[elem,id]] ;
JJ=nd[[elem,jd]] ;
SK[II,JJ]=SK[II,JJ]+k[elem][[id,jd]] ;
SM[II,JJ]=SM[II,JJ]+m[elem][[id,jd]] ;
SF[II]=SF[II]+f[elem][[id,1]],{jd,1,esd},{id,1,esd}] ;
elem=elem+1 ;
If [elem<=topelem,Goto[basla],Continue] )
(*****)
Read[file,Hold[Expression]] ;
dys=Read[file,Number] ;
Do[
d=Read[file,Number] ;
Pdeger=Read[file,Number] ;
SF[d]=SF[d]+Pdeger,
{dys}]
Read[file,Hold[Expression]] ;
tks=Read[file,Number] ;
Do[
sd=Read[file,Number] ;
Mdeger=Read[file,Number] ;
SM[sd,sd]=SM[sd,sd]+Mdeger,
{tks}]
Close[file]
(*** Sistem Matrislerinde Indirgeme Uygulaniyor ***)
RSF=Table[{0},{asd}]
RSK=IdentityMatrix[asd]
RSM=IdentityMatrix[asd]
Do[
RSK[[II,JJ]]=0 ;
RSM[[II,JJ]]=0,{II,1,asd},{JJ,1,asd}]
Do[
ii=lista[[id]] ;
jj=lista[[jd]] ;
RSK[[id,jd]]=SK[ii,jj] ;
RSM[[id,jd]]=SM[ii,jj] ;
RSF[[id,1]]=SF[ii,{jd,1,asd},
{id,1,asd}]

```

```

Print["*****"]
Print["***** Alt Uzaylarda Iterasyon *****"]
Print["*****          +          *****"]
Print["***** Genel Jacobi Metodu ile *****"]
Print["***** Ozel Deger Hesabi *****"]
Print["*****"]
Clear[m,f];Clear[SK,SM,SF]
k=RSK;m=RSM
k1[1]=k; m1[1]=m
n=Dimensions[k1[1]] [[1]]
Print["Sistem Serbestlik Derecesi=",n]
mod=Input["Istenen Mod Sayisi="]
q=mod
listmod=Table[0.,{q}]
list=Table[listmod,{n}]
Do[list[[i,i]]=1. ,{i,1,q}]
x[1]=list
y[1]=m1[1].x[1]

iter=10
Do[Print["Iterasyon No=",iii] ;
  yt=Transpose[y[iii]] ;
  Do[ a[i]=LinearSolve[k1[1],yt[[i]]],{i,q}] ;
  xb[iii+1]=Transpose[Table[a[i],{i,q}] ] ;
  k1[iii+1]=Transpose[xb[iii+1]].y[iii] ;
  yb[iii+1]=m1[1].xb[iii+1] ;
  m1[iii+1]=Transpose[xb[iii+1]].yb[iii+1] ;
  (*** Jacobi Metodu ile Alt Uzayda Ozdeger Cozumu ***) ;
  k=k1[iii+1]
  m=m1[iii+1]
  n=Dimensions[k][[1]] ;
  ts=5 ;
  fi=IdentityMatrix[n] ;
  kk=IdentityMatrix[n] ;
  k0=k ;
  m0=m ;
  Do[
    Do[kk[[i,i]]=k[[i,i]] m[[i,j]]-m[[i,i]] k[[i,j]] ;
    kk[[j,j]]=k[[j,j]] m[[i,j]]-m[[j,j]] k[[i,j]] ;
    kb=k[[i,i]] m[[j,j]]-k[[j,j]] m[[i,i]] ;
    If[kb==0,sign=1,sign=Sign[kb]] ;
    xx=kb/2.+ sign*Sqrt[(kb/2.)^2+kk[[i,i]]*kk[[j,j]]] ;
    gama=-kk[[i,i]]/xx ; alfa=kk[[j,j]]/xx ;
    p=IdentityMatrix[n] ;
    p[[i,i]]=1. ;
    p[[i,j]]=alfa ;

```

```

p[[j,i]]=gama ;
p[[j,j]]=1. ;
fi=fi.p ;
pt=Transpose[p] ;
k=pt.k.p ;m=pt.m.p,{i,1,n-1},{j,i+1,n} , {ii,1,ts} ;
(*****) ;
mi=IdentityMatrix[n] ;
Do[mi[[i,i]]=1/Sqrt[m[[i,i]]], {i,n}] ;
fi=fi.mi ;
y[iii+1]=yb[iii+1].fi ,{iii,1,iter}]

x[iter+1]=xb[iter+1].fi
lamda=IdentityMatrix[n]
Do[lamda[[i,i]]=k[[i,i]]/m[[i,i]],{i,n}]
Print["*****"]
Print["***** Ozel Degerler *****"]
Print["*****"]
lam=Table[1,{i,n}]
Do[lam[[i]]=lamda[[i,i]],{i,n}]
lam=Sort[lam]
Do[Print[Chop[lam[[i]]]],{i,n}]

Print["*****"]
Print["***** Sturm Dizisi Kontrolü *****"]
Print["*****"]
n=Dimensions[k1[1]] [[1]]
n1=Dimensions[k] [[1]]
listlam=Table[lamda[[i,i]],{i,n1}]
mu=Max[listlam]
mu=mu+mu/100
k=k1[1]-mu m1[1]
g=IdentityMatrix[n]
l=IdentityMatrix[n]
d=IdentityMatrix[n]
d[[1,1]]=k[[1,1]]
(***) Kolon Yükseklikleri Hesaplanıyor (***)
mm[1]=1
Do[
Do[If[k[[i,j]]!=0., (mm[j]=i;Return[mm[j]]),Continue],{i,1,n}] ;
,{j,2,n}]

```

```

(**** L.D.U seklinde carpanlara ayirma ****)
Do[
g[[mm[j],j]]=k[[mm[j],j]]
;

Do[g[[i,j]]=k[[i,j]]-Sum[l[[r,i]]*g[[r,j]],{r,Max[mm[i],mm[j]],(i-1)}]
, {i,(mm[j]+1),(j-1)}]
;
Do[l[[i,j]]=g[[i,j]]/d[[i,i]],{i,mm[j],(j-1)}]
;
d[[j,j]]=k[[j,j]]-Sum[l[[r,j]]*g[[r,j]],{r,mm[j],(j-1)}]
;
, {j,2,n}]

say=0
Do[If[d[[i,i]]<0, say=say+1,Continue],{i,1,n}]
If[say==q,Continue,(Print["Arada ozel deger var !!"];Pause[5])]
sortlist=Sort[listlam]

Print["*****"]
Print["**** Ters Iterasyon ile Ozvektorler Hesaplaniyor ****"]
Print["*****"]
Do[
mu=sortlist[[nn]]
;
Print["Mu=",mu]
;
v1[1]=Table[Random[ ],{n}]
;
iter=5
ks=RSK - mu RSM
;
Do[
mx=RSM.v1[i]
;
xb=LinearSolve[ks,mx]
;
payda=Sqrt[xb.RSM.xb]
;
v1[i+1]=xb/payda
;
ro[i+1]=(xb.RSK.xb)/(xb.RSM.xb)+mu
;
, {i,1,iter}]
;
omeg[nn]=N[ro[iter]-mu]
;
v[nn]=N[Chop[v1[iter]]]
;
a[nn]=v[nn]
;
, {nn,1,mod}]

Do[omeg[i]=Sqrt[omeg[i]],{i,mod}]
Do[Print[omeg[i]/2./N[Pi]],{i,mod}]
Do[RSFB[i]=a[i].RSF,{i,mod}]

Print["*****"]
Print["**** Fourier Transform Yontemi ile ****"]
Print["**** Dif. Denk. Takimi Cozumu ****"]
Print["**** Y''+c Y'+k Y=F/m ****"]
Print["*****"]
Clear[f,m]

```

```

z=Input["Sonum Orani = ?"]
Do[ksi[i]=z, {i,mod}]

Print["Ucgen Yukleme"]
t1=1.
t2=t1/2.
F0=1.0
ft[t_]:=Which[(t>=0.&& t<t2),F0*t/t2,(t>=t2&& t<=t1),
               F0*(t-t1)/(t2-t1),t>t1,0.0]

m=10
n=2^m
dt=0.01
tson=3.
T=n*dt

pdata=Table[{t,ft[t]},{t,0.,T,dt}]
p1=ListPlot[N[pdata],PlotJoined->True,AxesLabel->{"Zaman","Kuvvet"},
             PlotStyle->RGBColor[1,0,0],GridLines->Automatic,
             PlotRange->{0., 2 F0},PlotLabel->"Zaman Uzayi"]

pdata=Table[ft[t],{t,0.,T,dt}]
Print[" Yuklemenin Fourier Transformu Hesaplaniyor "]
pf=Conjugate [Fourier [N [pd]]]*Sqrt[n]*dt

p21=ListPlot[Re[pf],PlotJoined->True,PlotRange->All,Frame->True,FrameLabel
             ->{"Frekans","Kuvvet"},PlotStyle->RGBColor[0,0,1],GridLines->
             Automatic, PlotLabel->"Fourier transform uzayi (Reel kisim)"]
p22=ListPlot[Im[pf],PlotJoined->True,PlotRange->All,Frame->True,FrameLabel
             ->{"Frekans","Kuvvet"},PlotStyle->RGBColor[1,0,1],GridLines->
             Automatic, PlotLabel->"Fourier transform uzayi (Imajiner Kisim)"]

Print[" Frekans uzayinda her mod icin cozum yapiliyor "]
k1=n/2-1
k2=n/2+1
df=1./T

Do[Print["Mod=", imod] ;
   c=2*ksi[imod]*omeg[imod] ;
   kk=omeg[imod]^2 ;
   f=0.0 ;
   xf=Table[0.,{n}] ;

```

```

Do[w=2.*Pi*f ;
    Hw=1./(kk+c*I*w-w^2) ;
    xf[[i]]=N[ Hw*RSFB[imod][[1]]*pf[[i]]] ;
    Yf[imod,i]=xf[[i]] ;
    f=f+df ;
    , {i,1,k2}] ;
Do[Yf[imod,k2+i]=Conjugate[Yf[imod,k2-i]],{i,1,k1}] ;
Yf[imod,k2]=Re[Yf[imod,k2]] , {imod,mod}] ;

Do[listYf[imod]=Table[Yf[imod,i],{i,n}],{imod,mod}]
Print[" Ters Fourier Transmu Hesaplaniyor "],
Do[
ix[imod]=Sqrt[n]*Fourier[listYf[imod]]*df ;
rix[imod]=N[Re[ix[imod]]] , {imod,mod}] ;

Print["*****"]
Print["*** Modların Superpozisyonu Yapılıyor ***"]
Print["*****"]
Do[fx[i]=Sum[a[imod][[i]]*rix[imod], {imod,mod}],{i,asd}]
Print["*****"]
Print["*** Deplasman ve ivme değerleri hesaplanıyor ***"]
Print["*****"]
n1=tson/dt
Do[dept[icomp]=Table[{i*dt,fx[icomp][[i]]},{i,1,n1}],{icomp,asd}]
Do[dep[icomp]=Table[fx[icomp][[i]],{i,1,n1}],{icomp,asd}]

dof=Input["Cizilmesi İstenen Serbestlik Derecesi ="]
pos=Position[lista,dof][[1,1]]
plotdep=ListPlot[dept[pos],PlotJoined->True,PlotStyle->RGBColor[1,0,0],
    Frame->True,FrameLabel->{"Zaman","Deplasman"},
    PlotRange->All,PlotLabel->"Deplasman Ergisi"]
GridLines->Automatic]

d=Max[Abs[dep[pos]]]
d=N[d]
Print["Deplasmanın max. değeri =",d ]
post=Position[Abs[dep[pos]],d]
tmax=dept[pos][[post[[1,1]],1]]
Print[" Bu andaki zaman,tmax =",tmax]

Print["*****"]
Print["*** Eleman Üç Kuvvetleri Hesabı ***"]
Print["*****"]
topadim=tson/dt
elem=Input["Eleman Numarası =" ]
tt=0.

```

```

Do[
Do[ii=listb[[i]];up[ii]=0.,{i,bsd}] ;
Do[ii=listb[[i]];uc[ii,adim]=0.,{i,bsd}] ;
Do[ii=lista[[i]];uc[ii,adim]=dep[i][[adim]],{i,asd}] ;
Do[up[i]=uc[nd[[elem,i]],adim],{i,1,esd}] ;
Do[p[i]=Sum[kt[elem][[i,j]]*up[j],{j,esd}],{i,esd}] ;
(***) Sistem Koordinatlarinda Eleman Kuvvetleri (***)
      pt[tt]=N[p[1]] ;
      tt+=dt , {adim, topadim}] ;

ptt=Table[{tt,pt[tt]},{tt,0.,tson-dt,dt}]
ptm=Table[pt[tt],{tt,0.,tson-dt,dt}]
plot=ListPlot[ptt,PlotJoined->True,PlotStyle->RGBColor[1,0,0],
      PlotRange->All,GridLines->Automatic,PlotLabel->{elem,". elemana
ait"},
      Frame->True,,FrameLabel->{"Zaman","Kesme"}]
d=Max[Abs[ptm]]
d=N[d]
Print["Kesme Kuvvetinin Maksimum Degeri =", d ]
pos=Position[Abs[ptm],d]
tmax=ptt[[pos[[1,1]],1]]
Print["Bu Andaki Zaman ,t =" , tmax]

```

E1.2.4. MSRFT.MAT Programı

Programın işlem basamakları ve listesi aşağıdaki gibidir.

- 1-) Sistem ile ilgili bilgiler data dosyasından alınır.
- 2-) Önce lokal, sonra global koordinat takımında eleman rijitlik ve kütle matrisleri ile yük vektörü oluşturulmaktadır.
- 3-) Kodlama tekniği yardımı ile sistem rijitlik ve kütle matrisleri ile yük vektörü teşkil edilmektedir.
- 4-) Bilinen serbestlik dereceleri yardımı ile indirgenmiş sistem matrisleri elde edilmektedir.
- 5-) Sistemin serbest titreşim analizi Rayleigh oranı ile iterasyon metodu kullanılarak yapılır. Aynı zamanda bu metotla birlikte Gram-Schmidt ortogonalizasyonu kullanılmaktadır.
- 6-) Sturm teoremi yardımı ile bulunan özdeğerlerin sayısı kontrol edilir.
- 7-) Serbest titreşim analizinden sonra, mod süperpozisyon yöntemi ile zorlanmış titreşim analizi yapılmaktadır. Mod süperpozisyon ile girişimsiz hale gelen diferansiyel denklem takımı elde edildikten sonra, her mod için fourier uzayında ayrık fourier transform metodu ile modal deplasmanlar bulunmaktadır. Daha sonra ters fourier transform algoritması ile zaman uzayında modal deplasmanlar elde edilmektedir.
- 8-) Modal deplasmanlar bulunduktan sonra modların süperpozisyonu yapılarak sistemin esas deplasmanları bulunmaktadır.
- 9-) Son olarak istenen elemanların global koordinat takımında eleman uç kuvvetleri rijitlik matrisi metodu ile hesaplanmaktadır.

MSRFT.MAT Program Listesi

```
Print["*****"]
Print["***          MSRFT.MAT          ***"]
Print["***          Mod-Superpozisyon + Fourier Transform Metodu          ***"]
Print["***          Rayleigh & Gram-Schmidt & Shift          ***"]
Print["***          Duzlemi icinde yuklu duzlemsel cerceveler          ***"]
Print["*****"]
dosya=Input [ "Data dosyasi adi ( Tirnak icinde ) = ?" ]
file=OpenRead [ dosya ]
Read[file,Hold[Expression]]
Read[file,Hold[Expression]]
topelem=Read[file,Number]
nd=Table[{0.,0.,0.,0.,0.,0.}, {topelem}]
Read[file,Hold[Expression]]
Do[nd[[i,j]]=Read[file,Number],{i,topelem},{j,6}]
Read[file,Hold[Expression]]
bilsay=Read[file,Number]
listb=Table[0.,{bilsay}]
Read[file,Hold[Expression]]
Do[listb[[i]]=Read[file,Number],{i,bilsay}]
tsd=Max[nd]
listu=Union[Flatten[nd]]
lista=Complement[listu,listb]
bsd=Length[listb]
asd=tsd-bsd
eds=2
esd=6
(*****)

Read[file,Hold[Expression]]
For [elem=1, elem <=topelem, elem++,
  Ly[elem]=Read[file, Number]          ;
  Aty[elem]=Read[file, Number]         ;
  Aly[elem]=Read[file, Number]         ;
  Emy[elem]=Read[file, Number]         ;
  qy[elem]=Read[file, Number]          ;
  ty[elem]=Read[file, Number]          ;
  Pdegery[elem]=Read[file, Number]     ;
  ay[elem]=Read[file, Number]          ;
  Roy[elem]=Read[file, Number]         ;
  by[elem]=Ly[elem]-ay[elem]]
(*****)
```

```

Print[" Sistem Matrisleri Olusturuluyor"]
Do[SF[II]=0,{II,1,tsd}]
Do[SK[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]
Do[SM[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]
elem=1
(Label [basla] ;
  Print["Eleman no=",elem] ;
  L=Ly[elem] ;
  At=Aty[elem] ;
  Al=Aly[elem] ;
  Em=Emy[elem] ;
  q=qy[elem] ;
  t=ty[elem] ;
  Ro=Roy[elem] ;
  Pdeger=Pdegery[elem] ;
  aa=ay[elem] ;
  bb=by[elem] ;

T[elem]={ {Cos[t],Sin[t],0,0,0,0},
           {-Sin[t],Cos[t],0,0,0,0},
           {0, 0, 1,0,0,0},
           {0,0,0,Cos[t],Sin[t],0},
           {0,0,0,-Sin[t],Cos[t],0},
           {0,0,0, 0, 0, 1}} ;

ke={ {Al*L^2/At,0,0,-Al*L^2/At,0,0},
      {0, 12, 6*L,0,-12,6*L},
      {0, 6*L,4*L^2,0,-6*L,2*L^2},
      {-Al*L^2/At,0,0,Al*L^2/At,0,0},
      {0, -12, -6*L, 0, 12,-6*L},
      {0,6*L,2*L^2,0,-6*L, 4*L^2}}*Em*At/L^3 ;
k[elem]=Transpose[T[elem]].ke.T[elem] ;
kt[elem]=k[elem] ;

me={ {140,0,0, 70,0,0},
      {0, 156,22*L,0,54,-13*L},
      {0,0,22*L,4*L^2,0,13*L,-3*L^2},
      {70,0,0,140,0,0},
      {0,54,13*L,0,156,-22*L},
      {0,-13*L,-3*L^2,0,-22*L,4*L^2}}*Ro*Al*L/420 ;
m[elem]=Transpose[T[elem]].me.T[elem] ;
mt[elem]=m[elem] ;

f3=Pdeger*aa*bb^2/L^2; f6=-Pdeger*bb*aa^2/L^2 ;
feP={ {0},{Pdeger*bb/L+(f3+f6)/L},{f3},{0},{Pdeger*aa/L-(f3+f6)/L},{f6}} ;
feq={ {0},{q*L/2},{q*L^2/12},{0},{q*L/2},{-q*L^2/12}} ;

```

```

fe=feP+feq ;
f[elem]=Transpose[T[elem]].fe ;

Do[
II=nd[[elem,id]] ;
JJ=nd[[elem,jd]] ;
SK[II,JJ]=SK[II,JJ]+k[elem][[id,jd]] ;
SM[II,JJ]=SM[II,JJ]+m[elem][[id,jd]] ;
SF[II]=SF[II]+f[elem][[id,1]},{jd,1,esd},{id,1,esd}] ;

elem=elem+1 ;
If [elem<=topelem,Goto[basla],Continue] )
(*****)
Read[file,Hold[Expression]] ;
dys=Read[file,Number] ;
Do[
d=Read[file,Number] ;
Pdeger=Read[file,Number] ;
SF[d]=SF[d]+Pdeger,
{dys}]
Read[file,Hold[Expression]] ;
tk=Read[file,Number] ;
Do[
sd=Read[file,Number] ;
Mdeger=Read[file,Number] ;
SM[sd,sd]=SM[sd,sd]+Mdeger,
{tk}]
Close[file]
(***) Sistem Matrislerinde Indirgeme Uygulaniyor (***)
RSF=Table[{0},{asd}]
RSK=IdentityMatrix[asd]
RSM=IdentityMatrix[asd]
Do[
RSK[[II,JJ]]=0 ;
RSM[[II,JJ]]=0,{II,1,asd},{JJ,1,asd}]
Do[
ii=lista[[id]] ;
jj=lista[[jd]] ;
RSK[[id,jd]]=SK[ii,jj] ;
RSM[[id,jd]]=SM[ii,jj] ;
RSF[[id,1]]=SF[ii]},{jd,1,asd},
{id,1,asd}]

```

```

Print["*****"]
Print["***** Rayleigh & Gram-Schmidt *****"]
Print["***** Ozel Deger Hesabi *****"]
Print["*****"]

n=Dimensions[RSK][[1]]
Print["Toplam Serbestlik Derecesi=",n]
mod=Input["Istenen Mod Sayisi="]
mod1=mod-1
sayac=0
(Label [Rayleigh] ; mod1+=1 ;
sayac+=1 ;
v[1]=Table[Random [ ], {n}] ;
v1=v[1] ;
ro[1]= 0.0 ;
tol=10^ (-6) ;
v[i+1]=v[1] ;
Print["Deneme Sayisi = " , sayac] ;
Do[
fi[eigs]=v[i+1] ;
alfa[eigs]=(fi[eigs].RSM.v1) ;
v[1]=v1-Sum[alfa[ii] fi[ii] , {ii,2,eigs}] ;
i=0 ;
say=0 ;
( Label[basla2] ;
i=i+1 ;
ks=RSK-ro[i]*RSM ;
mx=RSM.v[i] ;
xb=LinearSolve[ks,mx] ;
payda=Sqrt [(xb.RSM.xb)] ;
v[i+1]=xb/payda ;
ro[i+1]=(xb.ks.xb)/(xb.RSM.xb)+ro[i] ;
kontrol=Abs[ro[i+1]-ro[i]]/Abs[ro[i+1]] ;
If [kontrol<tol,Continue,Goto [basla2]] ) ;
omeg[eigs]=N[ro[i+1]] ;
Print[" Ozel deger (" ,eigs,")= " ,N[ro[i+1]]] ;
a[eigs]=v[i+1] , {eigs,1,mod1}] ;

Print["*****"] ;
Print["***** Sturm Dizisi Kontrolü *****"] ;
Print["*****"] ;
listlam=Table[omeg[i],{i,1,mod1}] ;
lislml1=Sort[listlam] ;
mu=lislml1[[mod]] ;
mu=mu+mu/100 ;
kkk=RSK-mu RSM ;

```

```

g=IdentityMatrix[n] ;
l=IdentityMatrix[n] ;
d=IdentityMatrix[n] ;
d[[1,1]]=kkk[[1,1]] ;
(***) Kolon Yükseklikleri Hesaplanıyor (***)
mm[1]=1
Do[
Do[If[kkk[[i,j]]!=0., (mm[j]=i;Return[mm[j]]),Continue],{i,1,n}] ;
, {j,2,n}] ;

(**** L.D.U seklinde carpanlara ayirma ****)
Do[
g[[mm[j],j]]=kkk[[mm[j],j]] ;
Do[g[[i,j]]=kkk[[i,j]]-Sum[l[[r,i]]*g[[r,j]],{r,Max[mm[i],mm[j]],(i-1)}]
, {i,(mm[j]+1),(j-1)}] ;

Do[l[[i,j]]=g[[i,j]]/d[[i,i]],{i,mm[j],(j-1)}] ;
d[[j,j]]=kkk[[j,j]]-Sum[l[[r,j]]*g[[r,j]],{r,mm[j],(j-1)}] , {j,2,n}] ;
say=0 ;
Do[If[d[[i,i]]<0, say=say+1,Continue],{i,1,n}] ;
If[say==mod,Continue,Goto[Rayleigh]] )

Print[" Siraya Konmus Ozel Degerler " )
list=Table[omeg[i], {i,mod}]
Print[Sort [list]]
Do[omeg[i]=Sqrt[omeg[i]],{i,mod}]
Print["Serbest Titreşim Frekanslari (w,Rad/Sn)"]
Print["*****"]
Do[Print[omeg[i]/2./N[Pi]],{i,mod}]
Do[RSFB[i]=a[i].RSF,{i,mod}]

Print["*****"]
Print["**** Fourier Transform Yontemi ile ****"]
Print["**** Dif. Denk. Takimi Cozumu ****"]
Print["**** Y''+c Y'+k Y=F/m ****"]
Print["*****"]
Clear[f,m]

z=Input["Sonum Orani = ?"]
Do[ksi[i]=z, {i,mod}]

Print["Ucgen Yukleme"]
t1=1.
t2=t1/2.
F0=1.0
ft[t_]:=Which[(t>=0.&&t<t2),F0*t/t2,(t>=t2&&t<=t1),
F0*(t-t1)/(t2-t1),t>t1,0.0]

```

```

m=10
n=2^m
dt=0.01
tson=3.
T=n*dt

pdata=Table[{t,ft[t]},{t,0.,T,dt}]
p1=ListPlot[N[pdata],PlotJoined->True,AxesLabel->{"Zaman","Kuvvet"},
            PlotStyle->RGBColor[1,0,0],GridLines->Automatic,
            PlotRange->{0., 2 FO},PlotLabel->"Zaman Uzayi"]

pdata=Table[ft[t],{t,0.,T,dt}]
Print[" Yuklemenin Fourier Transformu Hesaplaniyor "]
pf=Conjugate [Fourier [N [pd]]]*Sqrt[n]*dt

p21=ListPlot[Re[pf],PlotJoined->True,PlotRange->All,Frame->True,FrameLabel
            ->{"Frekans","Kuvvet"},PlotStyle->RGBColor[0,0,1],GridLines->
            Auotomatic, PlotLabel->"Fourier transform uzayi (Reel kisim)"]
p22=ListPlot[Im[pf],PlotJoined->True,PlotRange->All,Frame->True,FrameLabel
            ->{"Frekans","Kuvvet"},PlotStyle->RGBColor[1,0,1],GridLines->
            Auotomatic, PlotLabel->"Fourier transform uzayi (Imajiner Kisim)"]

Print[" Frekans uzayinda her mod icin cozum yapiliyor "]
k1=n/2-1
k2=n/2+1
df=1./T

Do[Print["Mod=", imod] ;
c=2*ksi[imod]*omeg[imod] ;
kk=omeg[imod]^2 ;
f=0.0 ;
xf=Table[0.,{n}] ;

Do[w=2.*Pi*f ;
Hw=1./(kk+c*I*w-w^2) ;
xf[[i]]=N[ Hw*RSFB[imod][[1]]*pf[[i]]] ;
Yf[imod,i]=xf[[i]] ;
f=f+df ;
, {i,1,k2}] ;
Do[Yf[imod,k2+i]=Conjugate[Yf[imod,k2-i]],{i,1,k1}] ;
Yf[imod,k2]=Re[Yf[imod,k2]] , {imod,mod}] ;

Do[listYf[imod]=Table[Yf[imod,i],{i,n}],{imod,mod}]
Print[" Ters Fourier Transformu Hesaplaniyor "],

```

```

Do[
ix[imod]=Sqrt[n]*Fourier[listYf[imod]]*df ;
rix[imod]=N[Re[ix[imod]]] ,{imod,mod}] ;

Print["*****"]
Print["*** Modlarin Superpozisyonu Yapiliyor ***"]
Print["*****"]
Do[fx[i]=Sum[a[imod] [[i]]*rix[imod], {imod,mod}],{i,asd}]
Print["*****"]
Print["*** Deplasman ve ivme degerleri hesaplaniyor ***"]
Print["*****"]
n1=tson/dt
Do[dept[icomp]=Table[{i*dt,fx[icomp][[i]]},{i,1,n1}],{icomp,asd}]
Do[dep[icomp]=Table[fx[icomp] [[i]],{i,1,n1}],{icomp,asd}]

dof=Input["Cizilmesi Istenen Serbestlik Derecesi ="]
pos=Position[lista,dof][[1,1]]
plotdep=ListPlot[dept[pos],PlotJoined->True,PlotStyle->RGBColor[1,0,0],
Frame->True,FrameLabel->{"Zaman","Deplasman"},
PlotRange->All,PlotLabel->"Deplasman Ergisi"]
GridLines->Automatic]

d=Max[Abs[dep[pos]]]
d=N[d]
Print["Deplasmanın max. degeri =",d ]
post=Position[Abs[dep[pos]],d]
tmax=dept[pos][[post[[1,1]],1]]
Print[" Bu andaki zaman,tmax =",tmax]

Print["*****"]
Print["*** Eleman Uc Kuvvetleri Hesabi ***"]
Print["*****"]
topadim=tson/dt
elem=Input["Eleman Numarasi =" ]
tt=0.
Do[
Do[ii=listb[[i]];up[ii]=0.,{i,bsd}] ;
Do[ii=listb[[i]];uc[ii,adim]=0.,{i,bsd}] ;
Do[ii=lista[[i]];uc[ii,adim]=dep[i][[adim]],{i,asd}] ;
Do[up[i]=uc[nd[[elem,i]],adim],{i,1,esd}] ;
Do[p[i]=Sum[kt[elem][[i,j]]*up[j],{j,esd}],{i,esd}] ;
(***) Sistem Koordinatlarında Eleman Kuvvetleri (***)
pt[tt]=N[p[1]] ;
tt+=dt , {adim, topadim}] ;

```

```

ptt=Table[{tt,pt[tt]},{tt,0.,tson-dt,dt}]
ptm=Table[pt[tt],{tt,0.,tson-dt,dt}]
plot=ListPlot[ptt,PlotJoined->True,PlotStyle->RGBColor[1,0,0],
    PlotRange->All,GridLines->Automatic,PlotLabel->{elem,". elemana
ait"},
    Frame->True,,FrameLabel->{"Zaman","Kesme"}]
d=Max[Abs[ptm]]
d=N[d]
Print["Kesme Kuvvetinin Maksimum Degeri =", d ]
pos=Position[Abs[ptm],d]
tmax=ptt[[pos[[1,1]],1]]
Print["Bu Andaki Zaman ,t =" , tmax]

```

E1.2.5. MSSSP.MAT Programı

Programın işlem basamakları ve listesi aşağıdaki gibidir.

- 1-) Sistem ile ilgili bilgiler data dosyasından alınır.
- 2-) Önce lokal, sonra global koordinat takımında eleman rijitlik ve kütle matrisleri ile yük vektörü oluşturulmaktadır.
- 3-) Kodlama tekniği yardımı ile sistem rijitlik ve kütle matrisleri ile yük vektörü teşkil edilmektedir.
- 4-) Bilinen serbestlik dereceleri yardımı ile indirgenmiş sistem matrisleri elde edilmektedir.
- 5-) İndirgenmiş sistem matrisleri kullanılarak sistemin serbest titreşim analizi yapılmaktadır. Serbest titreşim analizinde önce alt uzaylarla iterasyon metodu ile özdeğer problemi çözülerek sistemin serbest titreşim frekansları bulunmakta ve sonra ters iterasyon metodu ile özvektörler (mod şekil fonksiyonları) bulunmaktadır.
- 6-) Sturm teoremi yardımı ile bulunan özdeğerlerin sayısı kontrol edilir.
- 7-) Sistemin Pseudostatik vektörü oluşturulur. Bunun için sisteme katılan serbestlik dereceleri ve Pseudostatik vektörün değeri ekrandan verilmektedir.
- 8-) Sistemin serbest titreşim periyotlarına karşılık gelen hız spektrum değerleri ekrandan verilmektedir.
- 9-) Hız spektrum değerleri yardımı ile sistemin mutlak maksimum modal deplasmanları hesaplanmaktadır.
- 10-) Her moddan gelen katkı bulunduktan sonra SRSS metodu ile sistemin esas deplasmanları bulunmaktadır.

MSSSP.MAT Program Listesi

```
Print["*****"]
Print["***                      MSSSP.MAT                      ***"]
Print["***          Mod-Superpozisyon + Spektrum Analizi          ***"]
Print["***  SUBSPACE/JACOBI/STURM CHECK/INVERSE ITERATION  ***"]
Print["*****"]
```

```
dosya=Input [ "Data dosyasi adi ( Tirnak icinde ) = ?" ]
```

```
file=OpenRead [ dosya ]
```

```
Read[file,Hold[Expression]]
```

```
Read[file,Hold[Expression]]
```

```
topelem=Read[file,Number]
```

```
nd=Table[{0.,0.,0.,0.,0.,0.}, {topelem}]
```

```
Read[file,Hold[Expression]]
```

```
Do[nd[[i,j]]=Read[file,Number],{i,topelem},{j,6}]
```

```
Read[file,Hold[Expression]]
```

```
bilsay=Read[file,Number]
```

```
listb=Table[0.,{bilsay}]
```

```
Read[file,Hold[Expression]]
```

```
Do[listb[[i]]=Read[file,Number],{i,bilsay}]
```

```
tsd=Max[nd]
```

```
listu=Union[Flatten[nd]]
```

```
lista=Complement[listu,listb]
```

```
bsd=Length[listb]
```

```
asd=tsd-bsd
```

```
eds=2
```

```
esd=6
```

```
(*****)
```

```
Print["Elemanlar ile ilgili bilgiler okunuyor"]
```

```
Read[file,Hold[Expression]]
```

```
Do [elem=i
```

```
;
```

```
  Ly[elem]=Read[file, Number]
```

```
;
```

```
  Aty[elem]=Read[file, Number]
```

```
;
```

```
  Aly[elem]=Read[file, Number]
```

```
;
```

```
  Emy[elem]=Read[file, Number]
```

```
;
```

```
  qy[elem]=Read[file, Number]
```

```
;
```

```
  ty[elem]=Read[file, Number]
```

```
;
```

```
  Pdegery[elem]=Read[file, Number]
```

```
;
```

```
  ay[elem]=Read[file, Number]
```

```
;
```

```
  Roy[elem]=Read[file, Number]
```

```
;
```

```
  by[elem]=Ly[elem]-ay[elem]
```

```
;
```

```
    ,{i,1,topelem}]
```

```
(*****)
```

```

Print["Eleman ve Sistem Matrisleri Olusturuluyor"]
Do[SF[II]=0,{II,1,tsd}]
Do[SK[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]
Do[SM[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]

elem=1
(Label [basla] ;
  Print["Eleman no=",elem] ;
  L=Ly[elem] ;
  At=Aty[elem] ;
  Al=Aly[elem] ;
  Em=Emy[elem] ;
  q=qy[elem] ;
  t=ty[elem] ;
  Ro=Roy[elem] ;
  Pdeger=Pdeger[elem] ;
  aa=ay[elem] ;
  bb=by[elem] ;

T[elem]={ {Cos[t],Sin[t],0,0,0,0},
           {-Sin[t],Cos[t],0,0,0,0},
           {0, 0, 1,0,0,0},
           {0,0,0,Cos[t],Sin[t],0},
           {0,0,0,-Sin[t],Cos[t],0},
           {0,0,0, 0, 0, 1}} ;

ke={ {Al*L^2/At,0,0,-Al*L^2/At,0,0},
      {0, 12, 6*L,0,-12,6*L},
      {0, 6*L,4*L^2,0,-6*L,2*L^2},
      {-Al*L^2/At,0,0,Al*L^2/At,0,0},
      {0, -12, -6*L, 0, 12,-6*L},
      {0,6*L,2*L^2,0,-6*L, 4*L^2}}*Em*At/L^3 ;
k[elem]=Transpose[T[elem]].ke.T[elem] ;
kt[elem]=k[elem] ;

me={ {140,0,0, 70,0,0},
      {0, 156,22*L,0,54,-13*L},
      {0,0,22*L,4*L^2,0,13*L,-3*L^2},
      {70,0,0,140,0,0},
      {0,54,13*L,0,156,-22*L},
      {0,-13*L,-3*L^2,0,-22*L,4*L^2}}*Ro*Al*L/420 ;
m[elem]=Transpose[T[elem]].me.T[elem] ;
mt[elem]=m[elem] ;

f3=Pdeger*aa*bb^2/L^2; f6=-Pdeger*bb*aa^2/L^2 ;
feP={ {0},{Pdeger*bb/L+(f3+f6)/L},{f3},{0},{Pdeger*aa/L-(f3+f6)/L},{f6}} ;

```

```

feq={{0},{q*L/2},{q*L^2/12},{0},{q*L/2},{-q*L^2/12}} ;
fe=feP+feq ;
f[elem]=Transpose[T[elem]].fe ;

Do[
II=nd[[elem,id]] ;
JJ=nd[[elem,jd]] ;
SK[II,JJ]=SK[II,JJ]+k[elem][[id,jd]] ;
SM[II,JJ]=SM[II,JJ]+m[elem][[id,jd]] ;
SF[II]=SF[II]+f[elem][[id,1]],{jd,1,esd},{id,1,esd}] ;
elem=elem+1 ;
If [elem<=topelem,Goto[basla],Continue] )
(*****)
Read[file,Hold[Expression]] ;
dys=Read[file,Number] ;
Do[
d=Read[file,Number] ;
Pdeger=Read[file,Number] ;
SF[d]=SF[d]+Pdeger,
{dys}]
Read[file,Hold[Expression]] ;
tks=Read[file,Number] ;
Do[
sd=Read[file,Number] ;
Mdeger=Read[file,Number] ;
SM[sd,sd]=SM[sd,sd]+Mdeger,
{tks}]
Close[file]
(***) Sistem Matrislerinde Indirgeme Uygulaniyor (***)
RSF=Table[{0},{asd}]
RSK=IdentityMatrix[asd]
RSM=IdentityMatrix[asd]
Do[
RSK[[II,JJ]]=0 ;
RSM[[II,JJ]]=0,{II,1,asd},{JJ,1,asd}]
Do[
ii=lista[[id]] ;
jj=lista[[jd]] ;
RSK[[id,jd]]=SK[ii,jj] ;
RSM[[id,jd]]=SM[ii,jj] ;
RSF[[id,1]]=SF[ii,{jd,1,asd},
{id,1,asd}]

```

```

Print["*****"]
Print["***** Alt Uzaylarda Iterasyon *****"]
Print["*****          +          *****"]
Print["***** Genel Jacobi Metodu ile *****"]
Print["***** Ozel Deger Hesabi *****"]
Print["*****"]
Clear[m,f];Clear[SK,SM,SF]
k=RSK;m=RSM
k1[1]=k; m1[1]=m
n=Dimensions[k1[1]] [[1]]
Print["Sistem Serbestlik Derecesi=",n]
mod=Input["Istenen Mod Sayisi="]
q=mod
listmod=Table[0.,{q}]
list=Table[listmod,{n}]
x[1]=list
y[1]=m1[1].x[1]

iter=10
Do[Print["Iterasyon No=",iii]                                ;
    yt=Transpose[y[iii]]                                      ;
    Do[ a[i]=LinearSolve[k1[1],yt[[i]]],{i,q}]                ;
    xb[iii+1]=Transpose[Table[a[i],{i,q}] ]                  ;
    k1[iii+1]=Transpose[xb[iii+1]].y[iii]                     ;
    yb[iii+1]=m1[1].xb[iii+1]                                 ;
    m1[iii+1]=Transpose[xb[iii+1]].yb[iii+1]                  ;
    (** Jacobi Metodu ile Alt Uzayda Ozdeger Cozumu **)      ;
    k=k1[iii+1]
    m=m1[iii+1]
    n=Dimensions[k][[1]]                                      ;
    ts=5                                                       ;
    fi=IdentityMatrix[n]                                       ;
    kk=IdentityMatrix[n]                                       ;
    k0=k                                                        ;
    m0=m                                                        ;
    Do[                                                         ;
        Do[kk[[i,i]]=k[[i,i]] m[[i,j]]-m[[i,i]] k[[i,j]]    ;
            kk[[j,j]]=k[[j,j]] m[[i,j]]-m[[j,j]] k[[i,j]]    ;
            kb=k[[i,i]] m[[j,j]]-k[[j,j]] m[[i,i]]           ;
            If[kb==0,sign=1,sign=Sign[kb]]                    ;
            xx=kb/2.+ sign*Sqrt[(kb/2.)^2+kk[[i,i]]*kk[[j,j]]] ;
            gama=-kk[[i,i]]/xx ; alfa=kk[[j,j]]/xx            ;
            p=IdentityMatrix[n]                                ;
            p[[i,i]]=1.                                         ;
            p[[i,j]]=alfa                                       ;
            p[[j,i]]=gama                                       ;

```

```

p[[j,j]]=1. ;
fi=fi.p ;
pt=Transpose[p] ;
k=pt.k.p ;m=pt.m.p,{i,1,n-1},{j,i+1,n} , {ii,1,ts} ;

(*****) ;
mi=IdentityMatrix[n] ;
Do[mi[[i,i]]=1/Sqrt[m[[i,i]]], {i,n}] ;
fi=fi.mi ;
y[iii+1]=yb[iii+1].fi ,{iii,1,iter}]

x[iter+1]=xb[iter+1].fi
lamda=IdentityMatrix[n]
Do[lamda[[i,i]]=k[[i,i]]/m[[i,i]],{i,n}]
Print["*****"]
Print["**** Ozel Degerler ****"]
Print["*****"]
lam=Table[1,{i,n}]
Do[lam[[i]]=lamda[[i,i]],{i,n}]
lam=Sort[lam]
Do[Print[Chop[lam[[i]]]],{i,n}]

Print["*****"]
Print["**** Sturm Dizisi Kontrolü ****"]
Print["*****"]
n=Dimensions[k1][1][1]
n1=Dimensions[k][1][1]
listlam=Table[lamda[[i,i]],{i,n1}]
mu=Max[listlam]
mu=mu+mu/100
k=k1[1]-mu m1[1]
g=IdentityMatrix[n]
l=IdentityMatrix[n]
d=IdentityMatrix[n]
d[[1,1]]=k[[1,1]]
(***) Kolon Yükseklikleri Hesaplanıyor (***)
mm[1]=1
Do[
Do[If[k[[i,j]]!=0., (mm[j]=i;Return[mm[j]]),Continue],{i,1,n}] ;
,{j,2,n}]

(**** L.D.U seklinde carpanlara ayirma ****)
Do[
g[[mm[j],j]]=k[[mm[j],j]] ;
Do[g[[i,j]]=k[[i,j]]-Sum[l[[r,i]]*g[[r,j]],{r,Max[mm[i],mm[j]],(i-1)}]
,{i,(mm[j]+1),(j-1)}] ;

```

```

Do[l[[i,j]]=g[[i,j]]/d[[i,i]],{i,mm[j],(j-1)}} ;
d[[j,j]]=k[[j,j]]-Sum[l[[r,j]]*g[[r,j]],{r,mm[j],(j-1)}} ;
, {j,2,n}]

say=0
Do[If[d[[i,i]]<0, say=say+1,Continue],{i,1,n}]
If[say==q,Continue,(Print["Arada ozel deger var !!"];Pause[5])]
sortlist=Sort[listlam]
Print["*****"]
Print["**** Ters Iterasyon ile Ozvektorler Hesaplaniyor ****"]
Print["*****"]
Do[
mu=sortlist[[nn]] ;
Print["Mu=",mu] ;
v1[1]=Table[Random[ ],{n}] ;
iter=5
ks=RSK - mu RSM ;
Do[
mx=RSM.v1[i] ;
xb=LinearSolve[ks,mx] ;
payda=Sqrt[xb.RSM.xb] ;
v1[i+1]=xb/payda ;
ro[i+1]=(xb.RSK.xb)/(xb.RSM.xb)+mu ;
, {i,1,iter}] ;
omeg[nn]=N[ro[iter]-mu] ;
v[nn]=N[Chop[v1[iter]]] ;
a[nn]=v[nn] ;
, {nn,1,mod}]

Do[omeg[i]=Sqrt[omeg[i]],{i,mod}]
Do[Print[omeg[i]/2./N[Pi]],{i,mod}]
Do[RSFB[i]=a[i].RSF,{i,mod}]

Print["*****"]
Print["**** Spektrum Analizi ile Deplasman ****"]
Print["**** ve Eleman Kuvvetlerinin Hesabi ****"]
Print["*****"]

(***) Pseudo-Statik Vektörü (PSV) Olusturuluyor (***)
dktsd=Input["Depreme katılan toplam d.o.f sayisini veriniz="]
rvek=Table[0.,{asd}]
Do[
dof=Input["Depreme katılan dof no'su=?"] ;
deg=Input["PSV 'un degeri=?"] ;
pos=Position[lista,dof][[1,1]] ;
rvek[[pos]]=deg ,{dktsd} ]

```

```

(*asd=mod*)
Do[alf[i]=(a[i].RSM.rvek),{i,mod}]
Do[period[i]=2*Pi/omeg[i], {i,mod}]
Do[Print[i," T = ", N[period[i]]] ;
Sv[i]=Input[" Bu period icin hiz spektrum degerini giriniz="]
, {i,mod}]

Do[Ymax[i]=(alf[i]/omeg[i])*Sv[i],{i,mod}]
Do[xmax[i]=a[i]*Ymax[i],{i,mod}]
Do[Xmax[j]=Sum[xmax[i][[j]]^2,{i,mod}], {j,1,asd}]
Do[Xmax[i]=Sqrt[Xmax[i]],{i,asd}]
!cls

Print[" ***** "]
Print[" ***** DOF-----> Umax ***** "]
Print[" ***** "]
Do[
Print[" ", i , ". modun katkisi " ] ;
Do[jj=lista[[ii]] ; umax[jj]=xmax[i] ;
Print[" ",jj,"----->",umax[jj] [[ii]] , {ii,asd}] ;
Print[" ##### "] ;
, {i,mod}]
Print[" Tum Modlarin Katkisi (SRSS) "]
Do[jj=lista[[i]] ; umax[jj]=Xmax[i] ;
Print[" ",jj," -----> ", umax[jj]], {i,asd}]
Print[" ##### "] ;

```

E1.2.6. MSRSP.MAT Programı

Programın işlem basamakları ve listesi aşağıdaki gibidir.

- 1-) Sistem ile ilgili bilgiler data dosyasından alınır.
- 2-) Önce lokal, sonra global koordinat takımında eleman rijitlik ve kütle matrisleri ile yük vektörü oluşturulmaktadır.
- 3-) Kodlama tekniği yardımı ile sistem rijitlik ve kütle matrisleri ile yük vektörü teşkil edilmektedir.
- 4-) Bilinen serbestlik dereceleri yardımı ile indirgenmiş sistem matrisleri elde edilmektedir.
- 5-) Sistemin serbest titreşim analizi Rayleigh oranı ile iterasyon metodu kullanılarak yapılır. Ayrıca bu metotla birlikte Gram-Schmidt ortogonalizasyonu kullanılmaktadır.
- 6-) Sturm teoremi yardımı ile bulunan özdeğerlerin sayısı kontrol edilir.
- 7-) Sistemin Pseudostatik vektörü oluşturulur. Bunun için sisteme katılan serbestlik dereceleri ve Pseudostatik vektörün değeri ekrandan verilmektedir.
- 8-) Sistemin serbest titreşim periyotlarına karşılık gelen hız spektrum değerleri ekrandan verilmektedir.
- 9-) Hız spektrum değerleri yardımı ile sistemin mutlak maksimum modal deplasmanları hesaplanmaktadır.
- 10-) Her moddan gelen katkı bulunduktan sonra SRSS metodu ile sistemin esas deplasmanları bulunmaktadır.

MSRSP.MAT Program Listesi

```
Print["*****"]
Print["***                MSRSP.MAT                ***"]
Print["***                Mod-Superpozisyon + Spektrum                ***"]
Print["***                Rayleigh & Gram-Schmidt & Shift                ***"]
Print["*****"]
```

```
dosya=Input [ "Data dosyasi adi ( Tirnak icinde ) = ?" ]
```

```
file=OpenRead [ dosya ]
```

```
Read[file,Hold[Expression]]
```

```
Read[file,Hold[Expression]]
```

```
topelem=Read[file,Number]
```

```
nd=Table[{0.,0.,0.,0.,0.,0.}, {topelem}]
```

```
Read[file,Hold[Expression]]
```

```
Do[nd[[i,j]]=Read[file,Number],{i,topelem},{j,6}]
```

```
Read[file,Hold[Expression]]
```

```
bilsay=Read[file,Number]
```

```
listb=Table[0.,{bilsay}]
```

```
Read[file,Hold[Expression]]
```

```
Do[listb[[i]]=Read[file,Number],{i,bilsay}]
```

```
tsd=Max[nd]
```

```
listu=Union[Flatten[nd]]
```

```
lista=Complement[listu,listb]
```

```
bsd=Length[listb]
```

```
asd=tsd-bsd
```

```
eds=2
```

```
esd=6
```

```
(*****)
```

```
Read[file,Hold[Expression]]
```

```
For [elem=1, elem <=topelem, elem++,
```

```
    Ly[elem]=Read[file, Number]
```

```
;
```

```
    Aty[elem]=Read[file, Number]
```

```
;
```

```
    Aly[elem]=Read[file, Number]
```

```
;
```

```
    Emy[elem]=Read[file, Number]
```

```
;
```

```
    qy[elem]=Read[file, Number]
```

```
;
```

```
    ty[elem]=Read[file, Number]
```

```
;
```

```
    Pdegery[elem]=Read[file, Number]
```

```
;
```

```
    ay[elem]=Read[file, Number]
```

```
;
```

```
    Roy[elem]=Read[file, Number]
```

```
;
```

```
    by[elem]=Ly[elem]-ay[elem] ]
```

```
(*****)
```

```

Print[" Sistem Matrisleri Olusturuluyor"]
Do[SF[II]=0,{II,1,tsd}]
Do[SK[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]
Do[SM[II,JJ]=0,{II,1,tsd},{JJ,1,tsd}]

elem=1
(Label [basla] ;
  Print["Eleman no=",elem] ;
  L=Ly[elem] ;
  At=Aty[elem] ;
  Al=Aly[elem] ;
  Em=Emy[elem] ;
  q=qy[elem] ;
  t=ty[elem] ;
  Ro=Roy[elem] ;
  Pdeger=Pdegery[elem] ;
  aa=ay[elem] ;
  bb=by[elem] ;

T[elem]={ {Cos[t],Sin[t],0,0,0,0},
           {-Sin[t],Cos[t],0,0,0,0},
           {0, 0, 1,0,0,0},
           {0,0,0,Cos[t],Sin[t],0},
           {0,0,0,-Sin[t],Cos[t],0},
           {0,0,0, 0, 0, 1}} ;

ke={ {Al*L^2/At,0,0,-Al*L^2/At,0,0},
      {0, 12, 6*L,0,-12,6*L},
      {0, 6*L,4*L^2,0,-6*L,2*L^2},
      {-Al*L^2/At,0,0,Al*L^2/At,0,0},
      {0, -12, -6*L, 0, 12,-6*L},
      {0,6*L,2*L^2,0,-6*L, 4*L^2}}*Em*At/L^3 ;
k[elem]=Transpose[T[elem]].ke.T[elem] ;
kt[elem]=k[elem] ;

me={ {140,0,0, 70,0,0},
      {0, 156,22*L,0,54,-13*L},
      {0,0,22*L,4*L^2,0,13*L,-3*L^2},
      {70,0,0,140,0,0},
      {0,54,13*L,0,156,-22*L},
      {0,-13*L,-3*L^2,0,-22*L,4*L^2}}*Ro*Al*L/420 ;
m[elem]=Transpose[T[elem]].me.T[elem] ;
mt[elem]=m[elem] ;

f3=Pdeger*aa*bb^2/L^2; f6=-Pdeger*bb*aa^2/L^2 ;
feP={ {0},{Pdeger*bb/L+(f3+f6)/L},{f3},{0},{Pdeger*aa/L-(f3+f6)/L},{f6}} ;

```

```

feq={{0},{q*L/2},{q*L^2/12},{0},{q*L/2},{-q*L^2/12}} ;
fe=feP+feq ;
f[elem]=Transpose[T[elem]].fe ;

Do[
II=nd[[elem,id]] ;
JJ=nd[[elem,jd]] ;
SK[II,JJ]=SK[II,JJ]+k[elem][[id,jd]] ;
SM[II,JJ]=SM[II,JJ]+m[elem][[id,jd]] ;
SF[II]=SF[II]+f[elem][[id,1]},{jd,1,esd},{id,1,esd}] ;

elem=elem+1 ;
If [elem<=topelem,Goto[basla],Continue] )
(*****)
Read[file,Hold[Expression]] ;
dys=Read[file,Number] ;
Do[
d=Read[file,Number] ;
Pdger=Read[file,Number] ;
SF[d]=SF[d]+Pdger,
{dys}]
Read[file,Hold[Expression]] ;
tks=Read[file,Number] ;
Do[
sd=Read[file,Number] ;
Mdger=Read[file,Number] ;
SM[sd,sd]=SM[sd,sd]+Mdger,
{tks}]
Close[file]
(***) Sistem Matrislerinde Indirgeme Uygulanıyor (***)
RSF=Table[{0},{asd}]
RSK=IdentityMatrix[asd]
RSM=IdentityMatrix[asd]
Do[
RSK[[II,JJ]]=0 ;
RSM[[II,JJ]]=0,{II,1,asd},{JJ,1,asd}]
Do[
ii=lista[[id]] ;
jj=lista[[jd]] ;
RSK[[id,jd]]=SK[ii,jj] ;
RSM[[id,jd]]=SM[ii,jj] ;
RSF[[id,1]]=SF[ii]},{jd,1,asd},
{id,1,asd}]

```

```

Print["*****"]
Print["***** Rayleigh & Gram-Schmidt  ***"]
Print["***** Ozel Deger Hesabi  *****"]
Print["*****"]

n=Dimensions[RSK][[1]]
Print["Toplam Serbestlik Derecesi=",n]
mod=Input["Istenen Mod Sayisi="]
mod1=mod-1
sayac=0
(Label [Rayleigh] ; mod1+=1 ;
sayac+=1 ;
v[1]=Table[Random [ ], {n} ;
v1=v[1] ;
ro[1]= 0.0 ;
tol=10^ (-6) ;
v[i+1]=v[1] ;
Print["Deneme Sayisi = " , sayac] ;
Do[
fi[eigs]=v[i+1] ;
alfa[eigs]=(fi[eigs].RSM.v1) ;
v[1]=v1-Sum[alfa[ii] fi[ii] , {ii,2,eigs}] ;
i=0 ;
say=0 ;
( Label[basla2] ;
i=i+1 ;
ks=RSK-ro[i]*RSM ;
mx=RSM.v[i] ;
xb=LinearSolve[ks,mx] ;
payda=Sqrt [(xb.RSM.xb)] ;
v[i+1]=xb/payda ;
ro[i+1]=(xb.ks.xb)/(xb.RSM.xb)+ro[i] ;
kontrol=Abs[ro[i+1]-ro[i]]/Abs[ro[i+1]] ;
If [kontrol<tol,Continue,Goto [basla2]] ) ;
omeg[eigs]=N[ro[i+1]] ;
Print[" Ozel deger (",eigs,")= " ,N[ro[i+1]]] ;
a[eigs]=v[i+1] , {eigs,1,mod1}} ;

Print["*****"] ;
Print["***** Sturm Dizisi Kontrolü  *****"] ;
Print["*****"] ;
listlam=Table[omeg[i],{i,1,mod1}] ;
lislml=Sort[listlam] ;
mu=lislml[[mod]] ;
mu=mu+mu/100 ;

```

```

kkk=RSK-mu RSM ;
g=IdentityMatrix[n] ;
l=IdentityMatrix[n] ;
d=IdentityMatrix[n] ;
d[[1,1]]=kkk[[1,1]] ;
(***) Kolon Yükseklikleri Hesaplanıyor (***)
mm[1]=1
Do[
Do[If[kkk[[i,j]]!=0., (mm[j]=i;Return[mm[j]]),Continue],{i,1,n}] ;
, {j,2,n}] ;

(**** L.D.U seklinde carpanlara ayirma ****)
Do[
g[[mm[j],j]]=kkk[[mm[j],j]] ;
Do[g[[i,j]]=kkk[[i,j]]-Sum[l[[r,i]]*g[[r,j]],{r,Max[mm[i],mm[j]],(i-1)}]
, {i,(mm[j]+1),(j-1)}] ;
Do[l[[i,j]]=g[[i,j]]/d[[i,i]],{i,mm[j],(j-1)}] ;
d[[j,j]]=kkk[[j,j]]-Sum[l[[r,j]]*g[[r,j]],{r,mm[j],(j-1)}] , {j,2,n}] ;
say=0 ;
Do[If[d[[i,i]]<0, say=say+1,Continue],{i,1,n}] ;
If[say==mod,Continue,Goto[Rayleigh]] )

Print[" Siraya Konmus Ozel Degerler " )
list=Table[omeg[i], {i,mod}]
Print[Sort [list]]
Do[omeg[i]=Sqrt[omeg[i]],{i,mod}]
Print["Serbest Titreşim Frekanslari (w,Rad/Sn)"]
Print["*****"]
Do[Print[omeg[i]/2./N[Pi]],{i,mod}]
Do[RSFB[i]=a[i].RSF,{i,mod}]

Print["*****"]
Print["**** Spektrum Analizi ile Deplasman ****"]
Print["**** ve Eleman Kuvvetlerinin Hesabi ****"]
Print["*****"]

(***) Pseudo-Statik Vektoru (PSV) Olusturuluyor (***)
dktsd=Input["Depreme katılan toplam d.o.f sayisini veriniz="]
rvek=Table[0.,{asd}]
Do[
dof=Input["Depreme katılan dof no'su=?"] ;
deg=Input["PSV 'un degeri=?"] ;
pos=Position[lista,dof][[1,1]] ;
rvek[[pos]]=deg ,{dktsd} ]

```

```

(*asd=mod*)
Do[alf[i]=(a[i].RSM.rvek),{i,mod}]
Do[period[i]=2*Pi/omeg[i], {i,mod}]
Do[Print[i," T = ", N[period[i]]] ;
Sv[i]=Input[" Bu period icin hiz spektrum degerini giriniz="]
, {i,mod}]

Do[Ymax[i]=(alf[i]/omeg[i])*Sv[i],{i,mod}]
Do[xmax[i]=a[i]*Ymax[i],{i,mod}]
Do[Xmax[j]=Sum[xmax[i][[j]]^2,{i,mod}], {j,1,asd}]
Do[Xmax[i]=Sqrt[Xmax[i]],{i,asd}]
!cls

Print[" ***** "]
Print[" ***** DOF-----> Umax ***** "]
Print[" ***** "]
Do[
Print[" ", i , ". modun katkisi " ] ;
Do[jj=lista[[i]] ; umax[jj]=xmax[i] ;
Print[" ",jj,"-----> ",umax[jj] [[i]] , {ii,asd}] ;
Print[" ##### "] ;
, {i,mod}]
Print[" Tum Modlarin Katkisi (SRSS) "]
Do[jj=lista[[i]] ; umax[jj]=Xmax[i] ;
Print[" ",jj," -----> ", umax[jj]], {i,asd}]
Print[" ##### "] ;

```

EK-2. ÖRNEKLERİN DATA DOSYALARI

E2.1. Örnek 5.9'un Data Dosyası

Polat Gulkan (Kesme Tipi 20 Katli Bina - With Roation)

Toplam eleman sayısı

60

Elemanların kod numaraları

1 2 3 4 5 6

4 5 6 7 8 9

7 8 9 10 11 12

10 11 12 13 14 15

13 14 15 16 17 18

16 17 18 19 20 21

19 20 21 22 23 24

22 23 24 25 26 27

25 26 27 28 29 30

28 29 30 31 32 33

31 32 33 34 35 36

34 35 36 37 38 39

37 38 39 40 41 42

40 41 42 43 44 45

43 44 45 46 47 48

46 47 48 49 50 51

49 50 51 52 53 54

52 53 54 55 56 57

55 56 57 58 59 60

58 59 60 61 62 63

64 65 66 67 68 69

67 68 69 70 71 72

70 71 72 73 74 75

73 74 75 76 77 78
76 77 78 79 80 81
79 80 81 82 83 84
82 83 84 85 86 87
85 86 87 88 89 90
88 89 90 91 92 93
91 92 93 94 95 96
94 95 96 97 98 99
97 98 99 100 101 102
100 101 102 103 104 105
103 104 105 106 107 108
106 107 108 109 110 111
109 110 111 112 113 114
112 113 114 115 116 117
115 116 117 118 119 120
118 119 120 121 122 123
121 122 123 124 125 126
4 5 6 67 68 69
7 8 9 70 71 72
10 11 12 73 74 75
13 14 15 76 77 78
16 17 18 79 80 81
19 20 21 82 83 84
22 23 24 85 86 87
25 26 27 88 89 90
28 29 30 91 92 93
31 32 33 94 95 96
34 35 36 97 98 99
37 38 39 100 101 102
40 41 42 103 104 105
43 44 45 106 107 108

46 47 48 109 110 111

49 50 51 112 113 114

52 53 54 115 116 117

55 56 57 118 119 120

58 59 60 121 122 123

61 62 63 124 125 126

Bilinen serbestlik derecesi sayisi

46

Bilinen serbestlik derecelerinin kodlari

1 2 3 64 65 66 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59
62 68 71 74 77 80 83 86 89 92 95 98 101 104 107 110 113 116 119 122
125

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.

[illegible]

6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.

Dugume etkiyen direk yuk (sayisi & serbestlik derecesi & degeri)

0

Toplanmis kutle (sayisi & serbestlik derecesi & degeri)

0

Polat Gulkan (Kesme Tipi 20 Katli Bina – Without Roation)

Toplam eleman sayisi

60

Elemanların kod numaralari

1 2 3 4 5 6

4 5 6 7 8 9

7 8 9 10 11 12

10 11 12 13 14 15

13 14 15 16 17 18

16 17 18 19 20 21

19 20 21 22 23 24

22 23 24 25 26 27

25 26 27 28 29 30

28 29 30 31 32 33

31 32 33 34 35 36

34 35 36 37 38 39

37 38 39 40 41 42

40 41 42 43 44 45

43 44 45 46 47 48

46 47 48 49 50 51

49 50 51 52 53 54

52 53 54 55 56 57

55 56 57 58 59 60

58 59 60 61 62 63

64 65 66 67 68 69

67 68 69 70 71 72

70 71 72 73 74 75

73 74 75 76 77 78

76 77 78 79 80 81

79 80 81 82 83 84

82 83 84 85 86 87

85 86 87 88 89 90
88 89 90 91 92 93
91 92 93 94 95 96
94 95 96 97 98 99
97 98 99 100 101 102
100 101 102 103 104 105
103 104 105 106 107 108
106 107 108 109 110 111
109 110 111 112 113 114
112 113 114 115 116 117
115 116 117 118 119 120
118 119 120 121 122 123
121 122 123 124 125 126
4 5 6 67 68 69
7 8 9 70 71 72
10 11 12 73 74 75
13 14 15 76 77 78
16 17 18 79 80 81
19 20 21 82 83 84
22 23 24 85 86 87
25 26 27 88 89 90
28 29 30 91 92 93
31 32 33 94 95 96
34 35 36 97 98 99
37 38 39 100 101 102
40 41 42 103 104 105
43 44 45 106 107 108
46 47 48 109 110 111
49 50 51 112 113 114
52 53 54 115 116 117
55 56 57 118 119 120

58 59 60 121 122 123

61 62 63 124 125 126

Bilinen serbestlik derecesi sayisi

86

Bilinen serbestlik derecelerinin kodlari

1 2 3 64 65 66 5 8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53 56 59

62 68 71 74 77 80 83 86 89 92 95 98 101 104 107 110 113 116 119 122

125

6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 69 72 75

78 81 84 87 90 93 96 99 102 105 108 111 114 117 120 123 126

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.
3.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	1.570796	0.	0.	2400.

[illegible]

6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.
6.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0	0.	0.	2400.

Dugume etkiyen direk yuk (sayisi & serbestlik derecesi & degeri)

0

Toplanmis kutle (sayisi & serbestlik derecesi & degeri)

0

E2.2. Örnek 6.1'in Data Dosyası

Polat Gulkan (Sayfa 71) & Clough (Sayfa 401)

Toplam eleman sayısı

10

Elemanların kod numaraları

1 2 3 4 5 6

4 5 6 7 8 9

7 8 9 10 11 12

10 11 12 13 14 15

13 14 15 16 17 18

16 17 18 19 20 21

19 20 21 22 23 24

22 23 24 25 26 27

25 26 27 28 29 30

28 29 30 31 32 33

Bilinen serbestlik derecesi sayısı

13

Bilinen serbestlik derecelerinin kodları

1 2 4 7 10 13 16 19 22 25 28 31 32

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.
1.	0.0108	0.36	$2 \cdot 10^{10}$.	0.	0.	0.	0.	2400.

Dugume etkiyen direk yuk (sayisi & serbestlik derecesi & degeri)

1

17 1.

Toplanmis kutle (sayisi & serbestlik derecesi & degeri)

0

E2.3. Örnek 6.2'nin Data Dosyası

R.Craig (sayfa 435)

Toplam eleman sayısı

3

Elemanların kod numaraları

1 2 3 4 5 6

4 5 6 7 8 9

7 8 9 10 11 12

Bilinen serbestlik derecesi sayısı

6

Bilinen serbestlik derecelerinin kodları

1 2 3 4 7 10

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
10.	0.0833	1.00	$1 \cdot 10^7$.	0.	0.	0.	0.	$2.591e-4$
10.	0.0833	1.00	$1 \cdot 10^7$.	0.	0.	0.	0.	$2.591e-4$
10.	0.0833	1.00	$1 \cdot 10^7$.	0.	0.	0.	0.	$2.591e-4$

Dugume etkiyen direk yük (sayısı & serbestlik derecesi & değeri)

1

11 -1.

Toplanmış kütle (sayısı & serbestlik derecesi & değeri)

1

11 0.02591

E2.4. Örnek 6.3'ün Data Dosyası

Yapı Dinamigi (Sayfa 239)

Toplam eleman sayısı

3

Elemanların kod numaraları

1 2 3 4 5 6

4 5 6 7 8 9

7 8 9 10 11 12

Bilinen serbestlik derecesi sayısı

9

Bilinen serbestlik derecelerinin kodları

1 2 3 5 6 8 9 11 12

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
300.	800000000.	1.	1.	0.	1.570796	0.	0.	0.
300.	800000000.	1.	1.	0.	1.570796	0.	0.	0.
300.	800000000.	1.	1.	0.	1.570796	0.	0.	0.

Dugume etkiyen direk yük (sayısı & serbestlik derecesi & değeri)

1

10 10.

Toplanmış kütle (sayısı & serbestlik derecesi & değeri)

3

4 1.

7 1.

10 1.

E2.5. Örnek 7.1'in Data Dosyası

Soil Structure Dersi (İki katli cerceve)

Toplam eleman sayısı

8

Elemanların kod numaraları

1 2 3 4 2 5

8 9 10 4 9 11

4 2 5 6 2 7

4 9 11 6 9 12

4 2 5 4 13 14

4 13 14 4 9 11

6 2 7 6 9 12

4 13 14 15 13 16

Bilinen serbestlik derecesi sayısı

6

Bilinen serbestlik derecelerinin kodları

1 2 3 8 9 10

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
1.	1.	1.	1.	0.	1.570796	0.	0.	0.
1.	1.	1.	1.	0.	1.570796	0.	0.	0.
1.	1.	1.	1.	0.	1.570796	0.	0.	0.
1.	1.	1.	1.	0.	1.570796	0.	0.	0.
0.5.	1.	1.	1.	0.	0	0.	0.	0.
0.5.	1.	1.	1.	0.	0	0.	0.	0.
1.	1.	1.	1.	0.	0	0.	0.	0.
0.5.	1.	1.	1.	0.	1.570796	0.	0.	0.

Dugume etkiyen direk yuk (sayisi & serbestlik derecesi & degeri)

1

6 1.

Toplanmis kutle (sayisi & serbestlik derecesi & degeri)

3

4 1.

6 1.

15 6.7547

E2.6. Örnek 8.1'in Data Dosyası

Yapı Dinamigi (Sayfa 239)

Toplam eleman sayısı

3

Elemanların kod numaraları

1 2 3 4 5 6

4 5 6 7 8 9

7 8 9 10 11 12

Bilinen serbestlik derecesi sayısı

9

Bilinen serbestlik derecelerinin kodları

1 2 3 5 6 8 9 11 12

(L)	(I)	(A)	(E)	(q)	(Aci)	(P)	(a)	(ro)
300.	800000000.	1.	1.	0.	1.570796	0.	0.	0.
300.	800000000.	1.	1.	0.	1.570796	0.	0.	0.
300.	800000000.	1.	1.	0.	1.570796	0.	0.	0.

Dugume etkiyen direk yuk (sayisi & serbestlik derecesi & degeri)

0

Toplanmış kutle (sayisi & serbestlik derecesi & degeri)

3

4 1.

7 1.

10 1.