

57450



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPILARIN DİNAMİK HESABINDA**  
**SPEKTRAL YÖNTEM**  
**VE**  
**SPEKTRUM ÇİZİMİ**

**İnş. Müh. Nuri Özhendekçi**

**F.B.E. İnşaat Mühendisliği Anabilim Dalı Yapı Programında**  
**hazırlanan**

**YÜKSEK LİSANS TEZİ**

**Tez Danışmanı: Prof. Zekeriya POLAT**

**İSTANBUL, 1996**

## İÇİNDEKİLER

SEMBOL LİSTESİ	III
ŞEKİL LİSTESİ	IV
TABLO LİSTESİ	V
KISALTMALAR LİSTESİ	VI
TEŞEKKÜR	VII
TÜRKÇE ÖZET	VIII
YABANCI DİLDE ÖZET (ABSTRACT)	IX

### 1. YAPILARIN DİNAMİK TİTREŞİM PARAMETRELERİNİ BELİRLEME YÖNTEMLERİ

1.1. GİRİŞ	1
1.2. NEWMARK YÖNTEMİ	1
1.3. STODOLA-VIANELLO YÖNTEMİ	2
1.4. RAYLEIGH YÖNTEMİ	2

### 2. ZAMAN ARTIMI (TIME HISTORY ANALYSIS) YÖNTEMİ

2.1. GİRİŞ	3
2.2. YAPININ TİTREŞİM MODELİNİN OLUŞTURULMASI	5
2.3. DİFERANSİYEL HAREKET DENKLEM SİSTEMİNİN ELDE EDİLMESİ	6
2.4. ZAMAN ARTIMI YÖNTEMİ İLE ÇÖZÜM	7
2.4.1. Denklem sisteminin indirgenmesi	7
2.4.2. Belirli bir yapının zaman artımı yöntemi ile çözümü	9
2.5. HASSASİYET ARAŞTIRMASI	17
2.5.1 Nümerik çözüm yönteminin özellikleri	17
2.5.2 Deneme sonuçları	19

3. SPEKTRAL ÇÖZÜM YÖNTEMİ	24
3.1 GİRİŞ	24
3.2 HAREKET DENKLEMİNİN ÇÖZÜMÜ	24
3.3 ÖRNEK YAPININ SPEKTRAL ÇÖZÜM YÖNTEMİ İLE HESABI	27
4. SONUÇ VE ÖNERİLER	31
PROGRAM - 1	33
PROGRAM - 2	48
KAYNAKLAR	59
ÖZGEÇMİŞ	60



## SEMBOL LİSTESİ

- $\{y\}$  (m):Deplasman ve hız vektörü  
[A]:Diferansiyel denklem sistemi (State-space) katsayı matrisi  
[B]:Diferansiyel denklem sisteminin ikinci taraf katsayı vektörü  
 $x_g(m)$ :Yer hareketi ötelemesi  
[m] ( $kNs^2/m$ ):Kütle matrisi  
[k] ( $kN/m$ ):Rijitlik matrisi  
[c] ( $kNs/m$ ):Sönüm matrisi  
[C]( $kNs/m$ ):Modal sönüm matrisi  
h(s):Nümerik integrasyonda adım aralığı  
T (m):Kesme hatâsı  
d(m):Hatâ  
R :Bağıl hatâ  
 $\omega(1/s)$ :Açısal frekans  
 $\xi$ :Sönüm oranı ( $=c/c_{cr}$ )  
Y :Genelleştirilmiş koordinatlardaki modal deplasman  
M :Genelleştirilmiş kütle  
C :Genelleştirilmiş sönüm  
K :Genelleştirilmiş rijitlik  
 $\{\phi\}$ :Modal vektör  
q :Katılım çarpanı

## ŞEKİL LİSTESİ

Şekil 2.1 Toplanmış kütleli yapı modeli	5
Şekil 2.2 Örnek yapının planı	9
Şekil 2.3 Erzincan Depremi Doğu-Batı bileşeni	13
Şekil 2.4 Erzincan Depremi Doğu-batı ivmesine 1.Kat cevabı	14
Şekil 2.5 Erzincan Depremi Doğu-batı ivmesine 5.Kat cevabı	15
Şekil 2.6 Erzincan Depremi Doğu-Batı ivmesine 10.Kat cevabı	16
Şekil 2.7 Kararsız çözüme örnek teşkil eden BSD sistemin cevabı	22
Şekil 2.8 Kararlı çözüme örnek teşkil eden BSD sistemin cevabı	23
Şekil 2.9 Spektral ivme	30



## TABLO LİSTESİ

Tablo 2.1 Zaman artımı yöntemi çözümünden bulunan kat kesme kuvvetleri	12
Tablo 2.2 $A=50$ için bağıl hatâ değerleri	20
Tablo 2.3 $A=500$ için bağıl hatâ değerleri	20
Tablo 2.4 Sinuzoidal hareket için bağıl hatâ değerleri	21
Tablo 3.1 Spektral analiz yöntemi çözümünden bulunan kat kesme kuvvetleri	29
Tablo 4.1 Kat kesme kuvvetlerindeki hata yüzdeleri	31



## KISALTMALAR LİSTESİ

BSD : Bir serbestlik dereceli sistem

ÇSD : Çok serbestlik dereceli sistem

SRSS : The square root of the sum of the squares (Kareleri toplamının karekökü)

CQC : Complete quadratic combination (Tam dörtgensel kombinasyon)

K : Nümerik çözümde “Kararsız davranış”



## TEŐEKKÜR

Tez konumun oluŐmasında ve ilerlemede benden hiçbir yardımı esirgemeyen Sayın Hocam Prof. Zekeriya Polat'a en içten teşekkürlerimi sunarım.

Nuri Özhendekçi





## ÖZET

Yapı mühendisliğinde, günümüzde, yapıların dinamik hesabı, uygulamada, özellikle yer sarsıntısına karşı hesaplarda, “Modların süperpozisyonu” adı ile anılan, aslında özel bir mod süperpozisyonu olan “spektral modal çözüm” yöntemi kullanılarak gerçekleştirilir. Ayrıca, hesaplarda, yapıya etkiyen sarsıntı iki ortogonal doğrultuda birbirinden bağımsız kabul edilir; bu kabul, yorucu rutinleri içeren bu tür hesapların projecilikte kullanılabilmesi için bir tür sadeleştirmedir.

Bu çalışmada; deprem kayıtlarından hareketle, bir serbestlik dereceli (BSD) sistem için spektral cevapların (responsların) çizimi programlandıktan sonra; yapı hesaplarının “modal spektral çözüm yöntemi” ile yapılması halinde yapılan hatânın miktarı belirlenmek istenmiştir. Bunun için, önce, dinamik hesap yöntemleri kısaca açıklandıktan sonra, kesin hesap yöntemi gözü ile bakılan “Zaman artımı yöntemi” (Time history analysis) ayrıntısı ile anlatılmış ve programlanmıştır.

Örnek deprem kaynak olarak “13 Mart 1992 Erzincan Depremi” doğu-batı ivme kayıtları dikkate alınmış; bu çalışmanın hacmi içinde sayısal örnek verebilmek için, nispeten kolay ve az rutin gerektiren düzlem çerçeve örnekler dikkate alınmıştır.

Geliştirilen program ile yapılan örnek çözümler göstermiştir ki; “Spektral çözüm yöntemi” ile yapılan çözümler, düzlem çerçevelerde bile, gerçek çözüme (Zaman artımı yöntemi ile bulunan sonuçlara) nispetle %20’ye varan hatalar içermektedir. Gelecekte, hesapların, uzun rutinler içerse de, bilgisayar yardımı ile, Zaman artımı yöntemi ile gerçekleştirilmesi beklenir.

## **ABSTRACT**

In structural engineering, dynamic analyses of structures are currently realized by spectral analyses methods. These methods consist of some superposition techniques to superimpose the modal spectral responses of structure. The design spectra representing a conventional earthquake is applied in two individual orthogonal directions for simplicity in design.

In the present work, it is aimed to find out the differences between the analyses realized by the “spectral analyses methods” and the “time history analysis method” which is generally accepted as the most accurate method. To do so, firstly a computer program is developed to draw spectral responses of a structure excited by an earthquake. Then, it is generalized to calculate the numerical responses of structure. Using the developed programs and the known spectral analyses techniques some samples of plane frame are analyzed and the obtained results are compared. The analyses and comparisons made, show that, the results provided by spectral analysis techniques are generally higher of about 20 % than the results obtained by time history analysis even for plane frame samples. Much more differences may be expected in case of three dimensional examples.

# 1 YAPILARIN DİNAMİK TİTREŞİM ÖZELLİKLERİNİN HESABINDA NÜMERİK YÖNTEMLER

## 1.1 GİRİŞ

Yapıların dinamik davranışında doğal frekans ve modları, belirleyici özellikleridir. Yapıların deprem hareketi altındaki davranışı da, büyük ölçüde bu özelliklerinin doğru tahminine bağlıdır.

Bir serbestlik dereceli sistemlerde (BSD), bu parametreler analitik olarak kolaylıkla bulunabilir. Buna mukabil çok serbestlik dereceli sistemlerde (ÇSD) doğrudan çözüm yapmak güç ve zaman alıcıdır. Bunun yerine, belirli yaklaşımlarla aynı sonuca götürecek sayısal çözüm yöntemlerin kullanılması düşünülebilir. Bu yöntemlerin başlıcaları, aşağıda özetlenmiştir.

## 1.2 NEWMARK YÖNTEMİ

Bu yöntemde yapılan temel kabuller, yapının toplanmış kütleli ve ortogonal modlara sahip olduğudur.

Çözüme, yapının dinamik cevabına (respons) en çok katkısı olan birinci modun tahmini ile başlanır. Tahmin edilen mod deplasmanlarına karşı gelen atalet kuvvetleri altında yapının gözönüne alınan serbestlik derecelerindeki deplasmanları hesaplanır. Bu işlem, pratik olarak, tahmin edilen ve hesaplanan mod şekli aynı oluncaya kadar iterasyon adımları şeklinde sürer. Yüksek modlarda, tahmin edilen mod şeklinden önceki modların bilinen payları çıkarılarak benzer iterasyonlar yapılır. [1,3]

### 1.3 STODOLA-VIANELLO YÖNTEMİ

Newmark yönteminin, deplasman hesabında gözönüne alınan serbestlik derecelerine karşı gelen rijitlik matrisini kullanmasına mukabil, Stodola-Vianello Yönteminde elastiklik (flexibility) matrisi kullanılır. Uygulanmasında, prensip olarak, Newmark Yönteminden önemli bir farkı yoktur. [1,3]

Her iki yöntemin de yakınsaklığı (konverjansı) oldukça yüksektir.

Newmark Yönteminin, yalnız çerçevelerden yada yalnız perdelerden (konsol) oluşan yapı sistemlerinde kullanılması pratiktir; buna mukabil, Stodola-Vianello Yöntemi daha geneldir.

### 1.4 RAYLEIGH YÖNTEMİ

Temelde, enerji prensiplerini esas alan bir yöntemdir. Fakat bu yöntemde de Newmark yöntem benzer şekilde yapının hesaplanacak moduna uyacak bir şekil fonksiyonunun tahmini gerekir. Tahmin edilen mod şekline karşı gelen atalet (kütle) kuvvetleri altında hesaplanan yapının deforme şekli tahmin edilen mod şekline yeterince yakın oluncaya dek iterasyona devam edilir. [1,3] Uygulamada çok kullanılan bir yöntemdir.

## 2 ZAMAN ARTIMI (TIME HISTORY ANALYSIS) YÖNTEMİ

### 2.1 GİRİŞ

Toplanmış kütleli olarak modellenen çok serbestlik dereceli (ÇSD) yapıların çözümünde, başlıca üç değişik yol takibedilir:

- (1) Doğrudan çözüm (integrasyon) yöntemi;
- (2) Modal çözüm yöntemi;
- (3) Spektral çözüm yöntemi.

Sonuncusu, modal çözüm yönteminin bir özel hâli olarak da sınıflandırılabilir.

Özellikle çok karmaşık bir yapısı olan yer sarsıntısı için doğrudan çözüm imkansız denecek ölçüde güçtür.

Modal çözüm yönteminde ise, doğal koordinatlar kullanılmasının, n-serbestlik dereceli bir titreşim hareketini n-adet bir serbestlik dereceli sistemin hareketine dönüştürmeye imkan verir; dış etki (yer sarsıntısı) çok karmaşık olsa da BSD sistem için çözümü, bilgisayarların da kullanılması ile güçlüğüne katlanılabilir rutin işlemler haline gelmektedir.

Bilindiği üzere,  $x_g(t)$  yer sarsıntısı için BSD sistemin hareket denklemi ile ÇSD sistemin hareket denklemleri :

$$\text{BSD sistem} \quad \ddot{y} + 2 \cdot \omega \cdot \xi \cdot \dot{y} + \omega^2 \cdot y = -\ddot{x}_g \quad (2.1)$$

$$\text{ÇSD sistem} \quad \ddot{Y}_i + 2 \cdot \omega_i \cdot \xi_i \cdot \dot{Y}_i + \omega_i^2 \cdot Y_i = -q_i \cdot \ddot{x}_g \quad (2.2)$$

dir.  $Y_i$  i-ci modun doğal koordinatını,  $q_i$  ise i-ci modun katılım çarpanını gösterir:

$$\{y_i\} = \{\phi_i\} \cdot Y_i \quad (2.3)$$

$$q_i = \frac{\{\phi_i\}^T \cdot [m] \cdot \{I\}}{\{\phi_i\}^T \cdot [m] \cdot \{\phi_i\}} \quad (2.4)$$

olarak bilinir.  $\{y_i\}$  i-ci moddaki gerçek (geometrik) koordinatları (ötelemeleri) gösterir;  $\{\phi_i\}$  i-ci moddur.

Görüldüğü üzere, BSD sistemin hareketi ile, ÇSD sistemin doğal koordinatlaedaki modal hareketi, yer sarsıntısı için, katılım çarpanı farkı ile aynıdır.

$\{y_i\}$ 'ler elde edildikten sonra bunların eşzamanlı toplamı yapılabilir:

$$\{y\} = \sum_{i=1}^r \{y_i\} = \sum_{i=1}^r \{\phi_i\} \cdot Y_i \quad (2.5)$$

Burada,  $r$ , dikkate alınması pratik olarak yeterli ilk mod sayısını gösterir.

(2.5) ile, yani,  $\{y_i\}$ 'lerin eş zamanlı nümerik değerlerinin toplanması ile yapılan hesaba, literatürde, "Zaman artımı yöntemi" (Time history analysis) denir.

Spektral Çözüm Yöntemi ise, projecilikte yaygın olan bir hesap yöntemidir. Sistemin modal özellikleri (frekans ve mod şekilleri) belirli bir deprem spektrumu veya bir yapın bölgesi için tasarım deprem spektrumu bilinirse, bilindiği üzere, modal doğal koordinatların maksimum (spektral) değerleri:

$$Y_{i,\max} = \frac{q_i \cdot Sa(\omega_i, \xi_i)}{\omega_i} \quad (2.6)$$

şeklinde elde edilebilir; (2.3) ile, buradan  $\{y_i\}_{\max}$  lar bulunup ilgili mod iç kuvvetlerine geçilebilir.

Burada sorun, eş zamanlı (instantaneous) olmayan bu modal maksimumların ne şekilde değerlendirilebileceği, bir başka deyimle, nasıl toplanabileceğidir. Bunun için, kaynaklarda [1,2,3,4] üç yöntem yer almaktadır; bunlar:

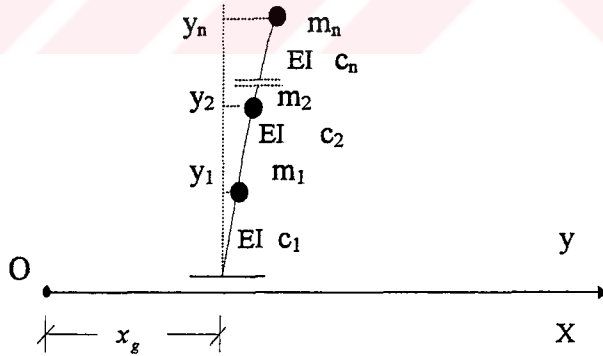
- Karelerinin toplamının karekökü (Square root of the sum of the squares, SRSS) yöntemi;
- CQC (Complete quadratic combination);
- Hint Yönetmeliği (1984) yöntemi ve benzerleri.

Bunlardan SRSS yöntemi hemen her deprem yönetmeliğinde bulunmaktadır. Modal frekansların birbirine yakınlığı ölçüsünde hâta oranı artar.

CQC yöntemi, California Standard (1990) ile uygulama alanı bulmuş nispeten genç bir yöntemdir; SRSS yönteminin, frekans ve sönüm oranına bağlılığını da dikkate alan bir genelleştirmesidir ve SRSS'nin bu bakımdan zaafını telafi etmektedir. Sönüm oranı,  $\xi_i = 0$  alınarak uygulandığında, SRSS ve CQC yöntemleri üst üste düşer.

## 2.2 YAPININ TİTREŞİM MODELİNİN OLUŞTURULMASI

Şekil 2.1'de, konsol olarak modellenen yapıya ait titreşim parametreleri görülmektedir.



Şekil 2.1

“O” noktası, uzayda bir noktadır; ve sabittir. Binanın zemini, OX eksenine doğrultusunda titreşmektedir. Bina, kütleleri kat hizalarında toplanmış ve viskoz sönümlü kabul edilmektedir.

## 2.3 DİFERANSİYEL HAREKET DENKLEM SİSTEMİNİN ELDE EDİLMESİ

Bölüm 2.1'de modellenen yapı için hareket denklem sistemi elde edilmeye çalışılırsa;

$$[m] = \begin{bmatrix} m_1 & 0 & \cdot & 0 \\ 0 & m_2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & m_n \end{bmatrix}_{n \times n} = \text{Kütle matrisi}$$

$$[k] = \begin{bmatrix} k_{11} & k_{12} & \cdot & k_{1n} \\ k_{21} & k_{22} & \cdot & k_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ k_{n1} & \cdot & \cdot & k_{nn} \end{bmatrix}_{n \times n} = \text{Rijitlik matrisi}$$

$$[c] = \begin{bmatrix} c_{11} & c_{12} & \cdot & c_{1n} \\ c_{21} & c_{22} & \cdot & c_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ c_{n1} & \cdot & \cdot & c_{nn} \end{bmatrix}_{n \times n} = \text{Sönüm matrisi}$$

$$\{y\} = \begin{Bmatrix} y_1 \\ y_2 \\ \cdot \\ y_n \end{Bmatrix}_{n \times 1} = \text{Deplasman vektörü}$$

olmak üzere;

$$[m] \cdot \{\ddot{x}\} + [c] \cdot \{\dot{y}\} + [k] \cdot \{y\} = \{0\}$$

şeklinde olur.

$$x_i = x_g + y_i \quad \ddot{x}_i = \ddot{x}_g + \ddot{y}_i$$

ifadelerinden,  $\ddot{x}_i$  yerine  $\ddot{x}_g$  ve  $\ddot{y}_i$  cinsinden değerleri konulursa;

$$[m] \cdot \{\ddot{y}\} + [c] \cdot \{\dot{y}\} + [k] \cdot \{y\} = -[m] \cdot \{I\} \cdot \ddot{x}_g \quad (2.7)$$



elde edilir. bu bir, ikinci dereceden diferansiyel denklem sistemidir.

## 2.4 ZAMAN ARTIMI YÖNTEMİ İLE ÇÖZÜM

### 2.4.1 Denklem sisteminin indirgenmesi

Bölüm 2.2'de modellenen bina için bulunan ikinci dereceden diferansiyel denklem sistemi, zaman artımı yönteminde kullanılmak üzere birinci dereceden bir diferansiyel denklem sistemine dönüştürülebilir. Bunun için ilk olarak ikinci tarafsız denklem sistemi ele alırsa:

$$[m] \cdot \{\ddot{y}\} + [c] \cdot \{\dot{y}\} + [k] \cdot \{y\} = \{0\} \quad (2.8)$$

yerine,

$$\begin{aligned} \ddot{y}_1 &= -\frac{k_{11}}{m_1} \cdot y_1 - \frac{k_{12}}{m_1} \cdot y_2 \cdots - \frac{k_{1n}}{m_1} \cdot y_n - \frac{c_{11}}{m_1} \cdot \dot{y}_1 - \frac{c_{12}}{m_1} \cdot \dot{y}_2 \cdots - \frac{c_{1n}}{m_1} \\ \ddot{y}_2 &= -\frac{k_{21}}{m_2} \cdot y_1 - \frac{k_{22}}{m_2} \cdot y_2 \cdots - \frac{k_{2n}}{m_2} \cdot y_n - \frac{c_{21}}{m_2} \cdot \dot{y}_1 - \frac{c_{22}}{m_2} \cdot \dot{y}_2 \cdots - \frac{c_{2n}}{m_2} \\ &\vdots \\ \ddot{y}_n &= -\frac{k_{n1}}{m_n} \cdot y_1 - \frac{k_{n2}}{m_n} \cdot y_2 \cdots - \frac{k_{nn}}{m_n} \cdot y_n - \frac{c_{n1}}{m_n} \cdot \dot{y}_1 - \frac{c_{n2}}{m_n} \cdot \dot{y}_2 \cdots - \frac{c_{nn}}{m_n} \end{aligned}$$

denklemeleri kullanılabilir. Burada  $\dot{y}_1 = y_{n+1}$  olacak şekilde değişken dönüşümü yapıp, matris formda yeniden yazılırsa;

$$\begin{Bmatrix} \dot{y}_1 \\ \vdots \\ \dot{y}_n \\ \dot{y}_{n+1} \\ \vdots \\ \dot{y}_{2n} \end{Bmatrix}_{2n \times 1} = \begin{bmatrix} 0 & \cdot & \cdot & 0 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & 0 & 0 & 0 & 0 & 1 \\ \frac{k_{11}}{m_1} & \frac{k_{12}}{m_1} & \cdot & \frac{k_{1n}}{m_1} & \frac{c_{11}}{m_1} & \frac{c_{12}}{m_1} & \cdot & \frac{c_{1n}}{m_1} \\ \frac{k_{21}}{m_2} & \frac{k_{22}}{m_2} & \cdot & \frac{k_{2n}}{m_2} & \frac{c_{21}}{m_2} & \frac{c_{22}}{m_2} & \cdot & \frac{c_{2n}}{m_2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{k_{n1}}{m_n} & \frac{k_{n2}}{m_n} & \cdot & \frac{k_{nn}}{m_n} & \frac{c_{n1}}{m_n} & \frac{c_{n2}}{m_n} & \cdot & \frac{c_{nn}}{m_n} \\ \frac{k_{n1}}{m_n} & \frac{k_{n2}}{m_n} & \cdot & \frac{k_{nn}}{m_n} & \frac{c_{n1}}{m_n} & \frac{c_{n2}}{m_n} & \cdot & \frac{c_{nn}}{m_n} \end{bmatrix}_{2n \times 2n} \begin{Bmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{2n} \end{Bmatrix}_{2n \times 1} \quad (2.9)$$

elde edilir. Görüldüğü gibi eşitlik, bilinmeyenlerin en yüksek mertebeden olan birinci dereceli olanlarına göre çözülmüştür. Eşitliğin sağında ise türevli terim yoktur. Denklem sayısı bilinmeyen sayısına eşittir. (2.9) ifadeleri, birinci dereceden bir diferansiyel denklem sistemidir. Bu dönüşüm sonucunda, denklem sisteminin mertebesi yarıya inerken denklem sayısı iki katına çıkmıştır.

Denklem sisteminin ikinci tarafını da hesaba katmak için gerekli matris tanımı yapılmak istenirse;

$$B = \begin{Bmatrix} 0 \\ \cdot \\ \cdot \\ 0^{(n)} \\ -1 \\ \cdot \\ \cdot \\ -1^{(2n)} \end{Bmatrix}_{2n \times 1} \quad (2.10)$$

katsayılar vektörü olmak üzere;

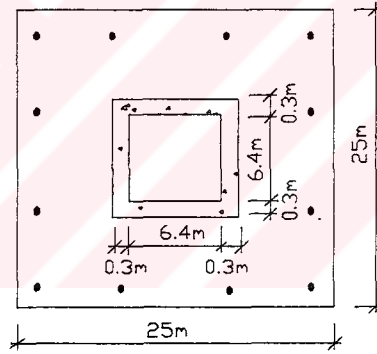
$$\{\dot{y}\} = [A] \cdot \{y\} + \{B\} \cdot \ddot{x}_{g_i} \quad (2.11)$$

şeklini alır.

Burada  $\{y\}_{2n \times 1}$  matrisinin ilk yarısındaki elemanlar, karşılık gelen serbestlik derecelerinin deplasmanları, ikinci yarısındakiler ise hızlarıdır. Benzer şekilde,  $\{\dot{y}\}_{2n \times 1}$  matrisinin ilk yarısındaki elemanlar, karşılık gelen serbestlik derecelerinin hızları, ikinci yarısındakiler ise ivmeleridir.  $[A]$  matrisi, birinci dereceden diferansiyel denklem sisteminin katsayılar matrisi olarak yukarıda bulunduğu gibidir.  $\ddot{x}_i$  ise zaman artımı yönteminin i-ci zaman aralığındaki sabit yer ivmesi değeridir.

#### 2.4.2 Belirli bir yapının Zaman artımı yöntemi ile çözümü

Buraya kadar verilen teorik formüller, burada, bir örnek yapı üzerinde açıklamalı olarak uygulanacaktır.



Şekil 2.2

Şekil 2.1’de modellenen yapının planı, şekil 2.2’deki gibi, elastiklik modülü C20 kalitesindeki betona karşı gelmek üzere  $E = 2.8E7 \text{ kN/m}^2$  olsun. Modal sönüm oranları, bütün modlar için eşit ve  $\xi = 0.05$  olarak kabul edilmiştir.

Örnek yapı kütlelerinin kat hizalarında toplandığı ve her katın  $600 \text{ kNs}^2/\text{m}$  kütlesi olduğu varsayılmıştır.

Yapının kütle ve rijitlikmatrisleri aşağıdaki gibidir:

$$[m] = \begin{bmatrix} 600 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 600 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 600 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & 600 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & 0 & 600 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & 600 & 0 & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & 600 & 0 & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 600 & 0 & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 600 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 600 \end{bmatrix}$$

$$[k] = \begin{bmatrix} -2436E8 & .1269E8 & -5509E6 & -3079E6 & -1719E6 & -9548E5 & -5226E5 & -2720E5 & -1159E5 & -3337E5 \\ .1269E8 & -2407E8 & .1285E8 & -4604E6 & -2570E6 & -1428E6 & -7815E5 & -4068E5 & -1733E5 & -4989E5 \\ & & -2398E8 & .1290E8 & -4313E6 & -2396E6 & -1312E6 & -6827E5 & -2909E5 & -8375E5 \\ & & & -2395E8 & .1292E8 & -4197E6 & 2298E6 & -1196E6 & -5095E5 & -1467E6 \\ & & & & -2394E8 & .1293E8 & -4082E6 & -2124E6 & -9050E5 & -2606E6 \\ & & & & & -2392E8 & .1294E8 & -3791E6 & -1615E6 & -4650E6 \\ & & & & & & -2390E8 & .1299E8 & -2886E6 & -8308E6 \\ & & & & & & & -2380E8 & .1316E8 & -1485E7 \\ & & & & & & & & -2352E8 & .1102E8 \\ & & & & & & & & & -7646E7 \end{bmatrix}$$

simetrik

Zaman artımı ile bire bir çözüm yapılacağı için yapının sönüm matrisinin elde edilmesi gereklidir. Sönüm matrisinden [C] faydalanarak yapı sönüm matrisi [c],

$$[C] = \begin{bmatrix} 2 \cdot \xi_1 \cdot \omega_1 \cdot M_1 & 0 & \cdot & 0 \\ 0 & 2 \cdot \xi_2 \cdot \omega_2 \cdot M_2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & 2 \cdot \xi_n \cdot \omega_n \cdot M_n \end{bmatrix} \quad (2.12)$$

İle elde edilebilir. Bu matris soldan  $[m] \cdot [\phi]$  ve sağdan  $[\phi]^T \cdot [m]$  matrisiyle çarpılırsa:

$$[m] \cdot [\phi] \cdot [C] \cdot [\phi]^T \cdot [m] = [m] \cdot [\phi] \cdot [\phi]^T \cdot [c] \cdot [\phi] \cdot [\phi]^T [m]$$

olur. Ortogonalite özelliklerinden yararlanarak;

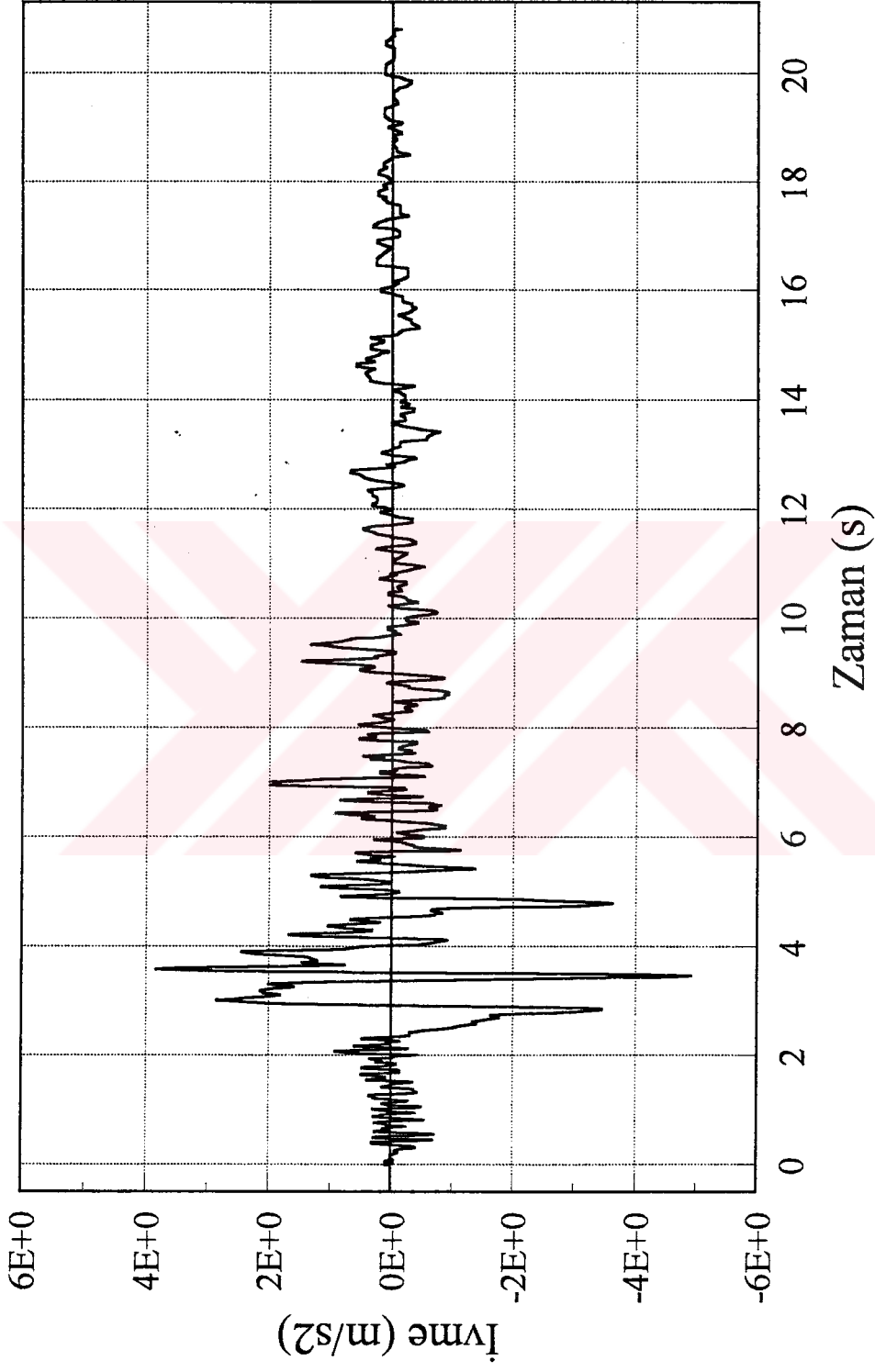
$$[c] = [m] \cdot [\phi] \cdot [C] \cdot [\phi]^T \cdot [m] \quad (2.13)$$

Şeklinde, sönüm matrisi hesaplanabilir.  $[\phi]$ , kütle matrisine göre orta normal formunda olmak üzere (2.13)'den [c] hesaplanırsa,

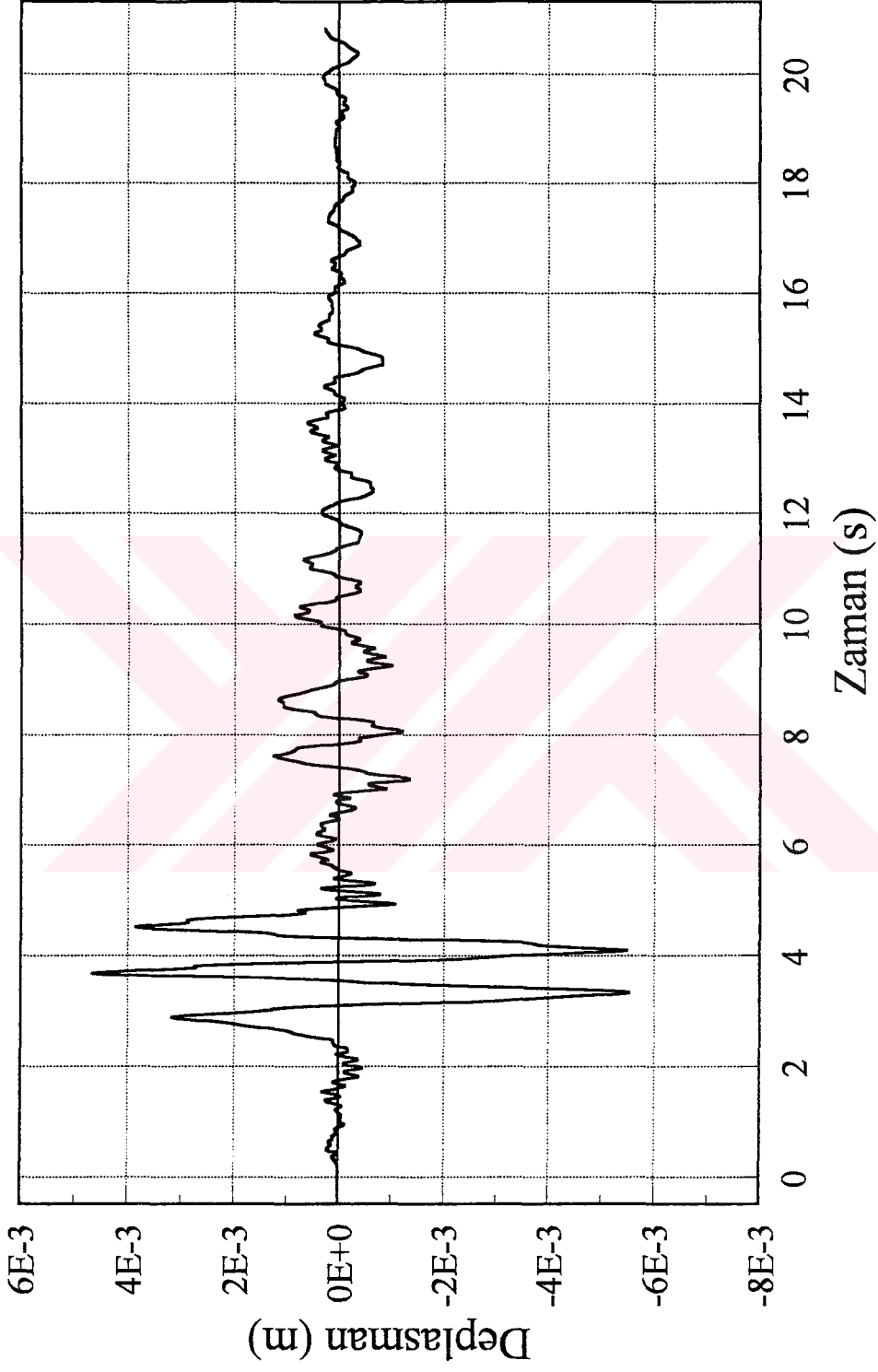


1 numaralı program ile,  $h=5E-4$  adım aralığı için, 5. derece ile yapılan zaman artımı çözümünde kat kesme kuvvetleri için Tablo 2.1 elde edilmiştir.

Kat	Kat Kesme Kuvveti (kN)
1	33060
2	31898
3	30528
4	28557
5	25916
6	22686
7	19680
8	16551
9	12606
10	7105

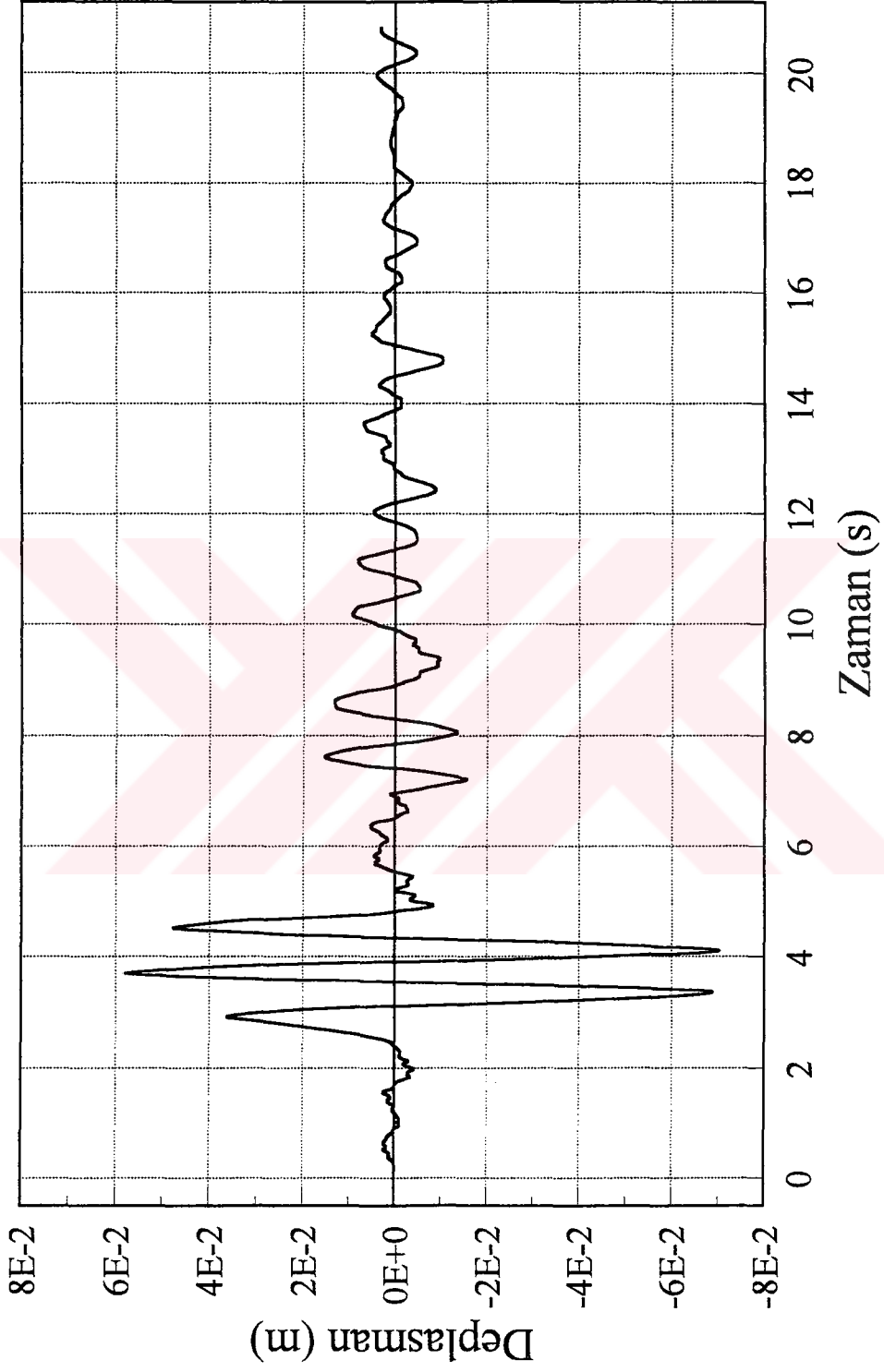


Şekil 2.3 - Erzincan Depremi Doğu-Batı bileşeni

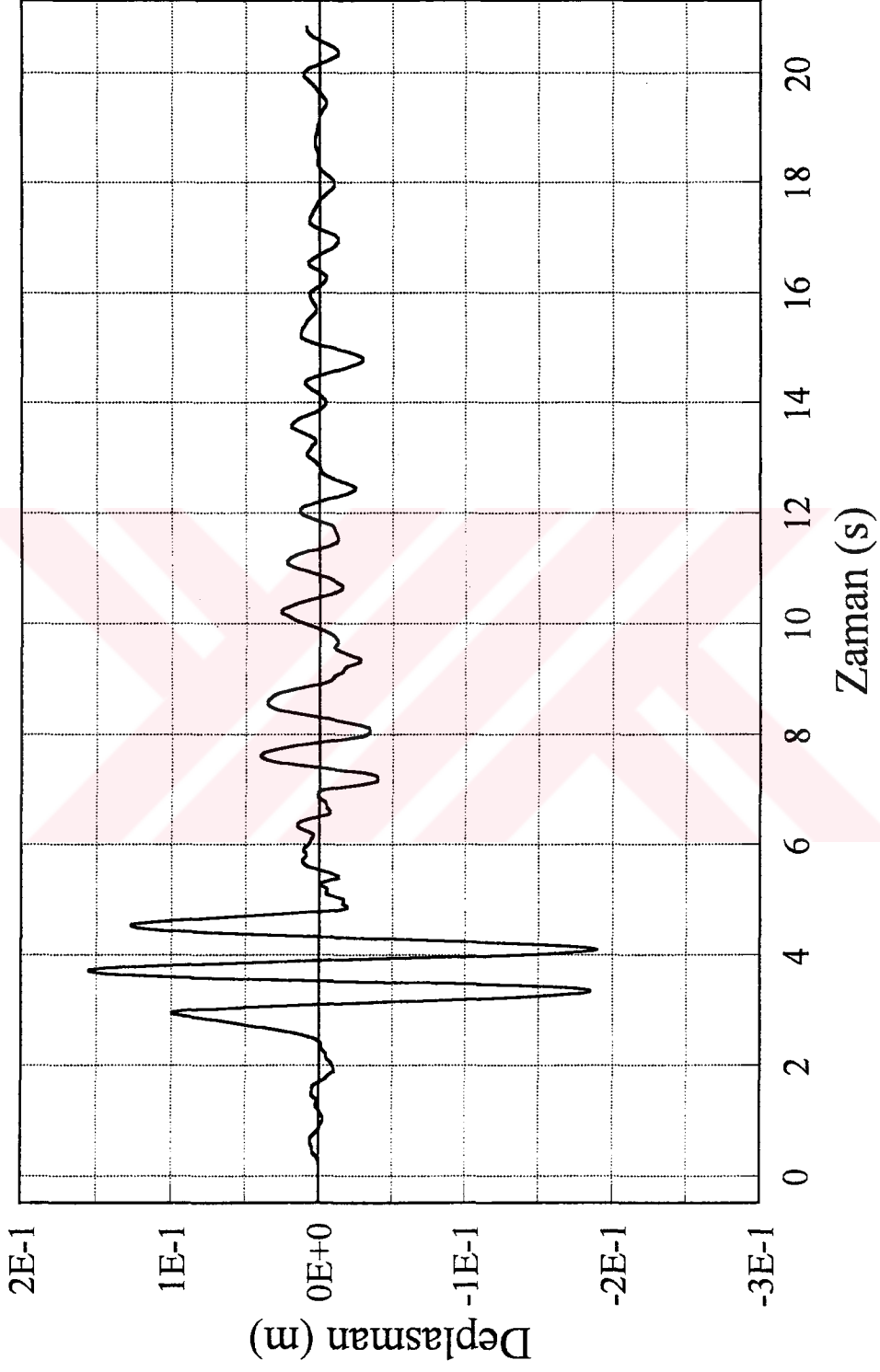


Şekil 2.4 - Erzincan Doğu Batı ivmesine 1.Kat cevabı





Şekil 2.5 - Erzincan Doğu Batı ivmesine 5.Kat cevabı



Şekil 2.6 - Erzincan Depremi Doğu-Batı ivmesine 10.Kat cevabı

## 2.5 HASSASİYET ARAŞTIRMASI

### 2.5.1 Nümerik çözüm yönteminin özellikleri

Ekte sunulan program, Adams-Bashford ve Adams-Moulton yöntemlerini, çeşitli derece ve adım aralıkları için birlikte kullanarak birinci dereceden bir diferansiyel denklem sistemini çözmektedir. Denemeler sırasında, seçilen adım aralığı ve yöntem derecesi, programdaki “karar” isimli süreçteki “h” ve “derece” değişkenleri değiştirilerek yapılmıştır. Program, en fazla 18. dereceye kadar yöntem katsayısı üretmektedir. Bunun nedeni, 18. dereceden büyük olan yöntem katsayılarının, kullanılan “extended” türündeki değişkenlerin basamak sayısı olan 19’u aşmasıdır. Dolayısıyla daha yüksek dereceden yöntem katsayısı üretmenin, hassasiyet açısından bir anlamı olmamaktadır.

Programın birinci dereceden diferansiyel denklem sistemlerini çözdüğü söylenmekle beraber “Türev” isimli süreçte yapılan değişikliklerle non-lineer çözümler içinde kullanılabilir. Bunun kaynağı, yöntemin analitik çözümün türüne bağlı olmaksızın türev fonksiyonunu enterpole edip entegre ederek çözüme gitmesidir.

$\dot{y} = f(x, y)$  diferansiyel denkleminin  $x_k$  noktasındaki gerçek çözümü  $y(x_k)$ , nümerik yöntemle elde edilen çözüm ise,  $y_k$  ile gösterilirse;

4. dereceden Adams-Moulton yöntemi:

$$y_{k+1} = y_k + \frac{1}{24} \cdot h \cdot (9 \cdot \dot{y}_{k+1} + 19 \cdot \dot{y}_k - 5 \cdot \dot{y}_{k-1} + \dot{y}_{k-2})$$

olarak bilinir. Yaklaşık değerler yerine gerçek değerler konursa;

$$y(x_{k+1}) = y(x_k) + \frac{1}{24} \cdot h \cdot (9 \cdot \dot{y}(x_{k+1}) + 19 \cdot \dot{y}(x_k) - 5 \cdot \dot{y}(x_{k-1}) + \dot{y}(x_{k-2})) + T$$

olur. Burada  $T$ , kesme hatâsı olarak adlandırılır.  $d_k = y(x_k) - y_k$  yerine konulursa,  $d$ 'ye bağlı,

$$\left(1 - \frac{9}{24} \cdot A \cdot h\right) \cdot d_{k+1} = \left(1 + \frac{19}{24} \cdot A \cdot h\right) \cdot d_k - \frac{5}{24} A \cdot h \cdot d_{k-1} + \frac{1}{24} \cdot A \cdot h \cdot d_{k-2} + T$$

fark denklemini elde edilir.

Başlangıç şartlarındaki hatânın çözüm boyunca etkisini incelemek için  $T$  kesme hatâsı ihmâl edilip fark denklemini çözülmüşse;

$$\left(1 - \frac{9}{24} \cdot A \cdot h\right) \cdot r^3 - \left(1 + \frac{19}{24} A \cdot h\right) \cdot r^2 + \frac{5}{24} A \cdot h \cdot r - \frac{1}{24} A \cdot h = 0$$

bulunur. Buradanda,  $r_1 \cong 1 + A \cdot h$  bulunur. Daha sonra denklem,

$$(24 - 9 \cdot A \cdot h) \cdot r^2 - 4 \cdot A \cdot h \cdot r + A \cdot h = 0$$

denklemine indirgenebilir.  $r_2$  ve  $r_3$ ,

$$\sqrt{A \cdot h / 24} \text{ veya } \mp \sqrt{-6 \cdot A \cdot h / 12} \text{ 'ye}$$

yaklaşırlar.

$$|r_2|, |r_3| < 1 + A \cdot h \cong e^{A \cdot h}$$

$$d_k \cong c_1 \cdot (1 + A \cdot h)^k + o \cdot e^{A \cdot k \cdot h}$$

Adams yöntemleriyle, birinci dereceden diferansiyel denklem sistemleri çözülmesi düşünüldüğünden,  $\dot{y} = A \cdot y; y(0) =$  olan denklem doğru çözüm kabul edilirse, bağıl hatâ, hatânın doğru çözüme bölünmesiyle;

$$R = \frac{c_1 \cdot e^{A \cdot k \cdot h} + o \cdot e^{A \cdot k \cdot h}}{e^{A \cdot x}} = c_1 + o$$

olarak bulunur.

Bunun anlamı, bağıl hatânın, çözümle birlikte aynı oranda büyüyeceğidir. Burada önemli olan bir nokta mevcuttur:  $\dot{y} = A \cdot y$ 'nin çözümü,  $y = c \cdot e^{A \cdot x}$  dir. Bu durumda bağıl hatâ, çözümle birlikte, yâni  $e^x$  ile orantılı olarak büyür. Fakat bir yapının dinamik davranışı, sinüzoidal bir harekettir; artı ve eksi değerler alır. Böylece bağıl hatâ, belirli sınırlar içinde kalabilir.

Gerçek çözüme asıl yakınsayan kök,  $r$  köküdür.  $r_2$  ve  $r_3$  kökleri, çözümün sonsuza gitmesine neden olabilirler. Hassas hesap yapmak için yöntemin yüksek derecelilerini kullanmanın bedeli, “fazlalık kök” veya “parazit kök” olarak adlandırılan  $r_2, r_3, \dots, r_n$  kökleriyle ödenir.

### 2.5.2 Deneme sonuçları

$\dot{y} = A \cdot y$ 'nin  $y(0) =$  için  $(0-1)s$  aralığındaki nümerik çözümlerinden aşağıdaki çözümler elde edilmiştir. Tablolarda, yatayda yöntem derecesi, düşeyde ise adım aralığı gösterilmiştir.

Tablo 2.2 Bağıl hatâ (A=50 için)						
	1	2	3	4	5	6
$10^{-2}$	$8.10^{-1}$	$5.10^{-1}$	$2.10^{-1}$	$1.10^{-1}$	$7.10^{-2}$	$5.10^{-2}$
$10^{-3}$	$2.10^{-2}$	$8.10^{-4}$	$5.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$
$10^{-4}$	$2.10^{-4}$	$8.10^{-7}$	$2.10^{-8}$	$2.10^{-8}$	$2.10^{-8}$	$2.10^{-8}$

Tablo 2.2 (Devamı) Bağıl hatâ (A=50 için)						
	7	8	9	12	13	16
$10^{-2}$	$4.10^{-2}$	$3.10^{-2}$	$3.10^{-2}$	$3.10^{-2}$	“K”	“K”
$10^{-3}$	$2.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$	“K”	“K”
$10^{-4}$	$2.10^{-8}$	$2.10^{-8}$	$2.10^{-8}$	$2.10^{-8}$	$2.10^{-8}$	$2.10^{-8}$

Tablo 2.3 Bağıl hatâ (A=500 için)						
	1	2	3	4	5	6
$10^{-2}$	1.0	1.0	1.0	1.0	1.0	1.0
$10^{-3}$	1.0	1.0	$8.10^{-1}$	$5.10^{-1}$	$2.10^{-1}$	$1.10^{-1}$
$10^{-4}$	$2.10^{-1}$	$8.10^{-3}$	$4.10^{-4}$	$4.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$

Tablo 2.3 (Devamı) Bağıl hatâ (A=500 için)						
	1	2	3	4	5	6
$10^{-2}$	1.0	1.0	1.0	1.0	1.0	1.0
$10^{-3}$	$5.10^{-2}$	$3.10^{-2}$	$3.10^{-2}$	“K”	“K”	“K”
$10^{-4}$	$2.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$	$2.10^{-5}$	“K”	“K”

Tablo 2.2 ve 2.3’de “K”, “kararsız davranış” anlamında kullanılmıştır. Nümerik yöntemin kararsız davranması, Bölüm 2.5.1’de açıklandığı gibi başlangıç veya kesme hatâlarının, çözüm boyunca etkilerini göstererek çözümleri sonsuza götürmesidir.

Yöntem derecesinin artması, çok az hassasiyet sağlarken parazit köklerin de artması nedeniyle kararsızlıkla sonuçlanabilmektedir; azalması ise kesme hatâlarından dolayı hassasiyeti düşürmektedir. Bundan sonraki hesaplarda kullanılmak üzere, “5 ci derece” seçilmiştir.

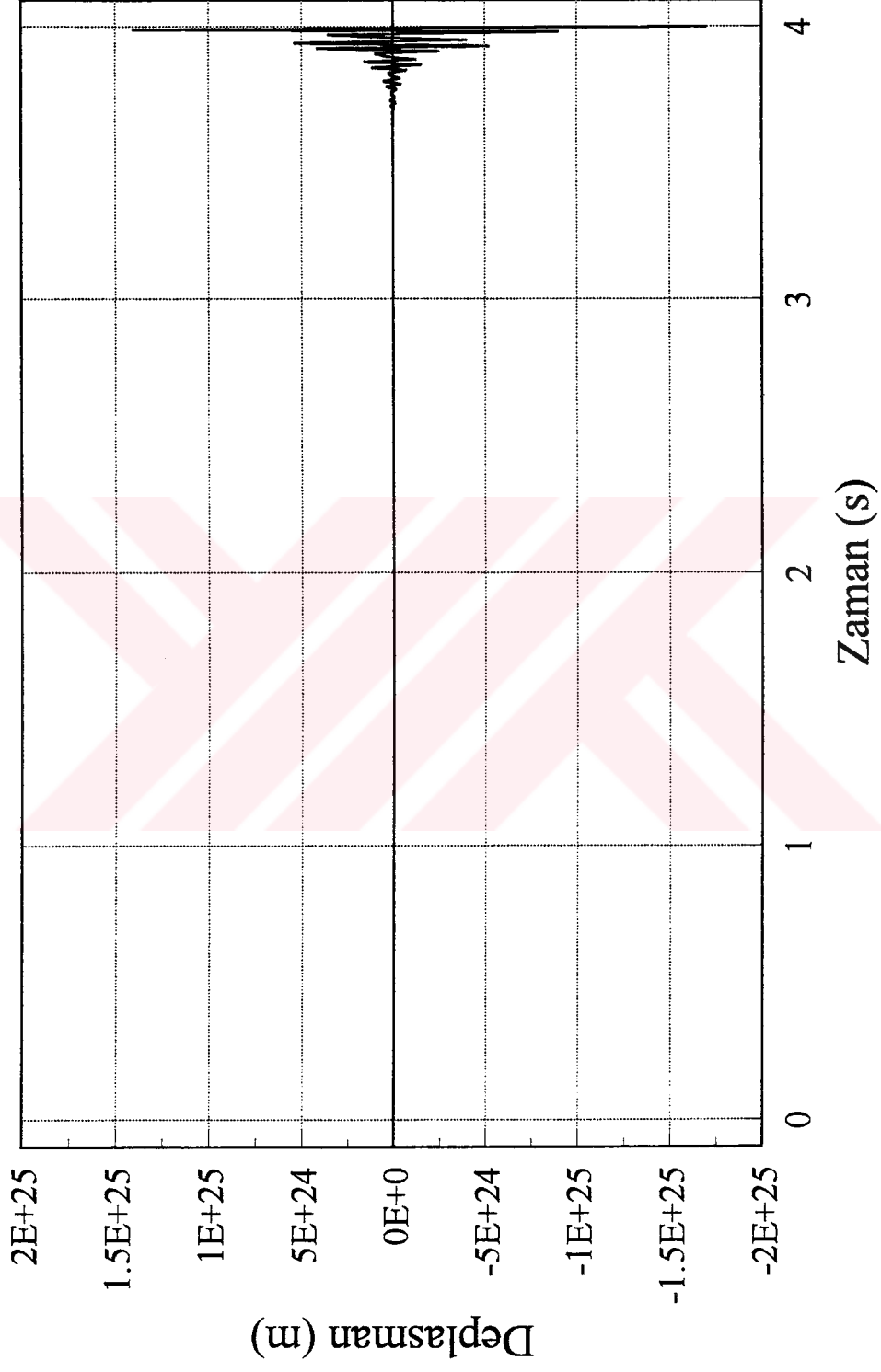
Tablo 2.2 ve 2.3 için gerçek çözüm,  $e^{Ax}$  dir. Oysa Bölüm 2.5.1’de de açıklandığı gibi dinamik davranış, sinüzoidal bir harekettir. Hatâ, çözüm boyunca daha yavaş büyür. Bu durum daha ayrıntılı incelendiğinde:

$$\ddot{y} + 2\xi\omega\dot{y} + \omega^2 y = 0$$

denklemini,  $\dot{y}(0)=0,01$ ,  $y(0)=0$  için  $(0-20)s$  aralığında  $\xi=0$  ve “5.derece“ için çözümlerse:

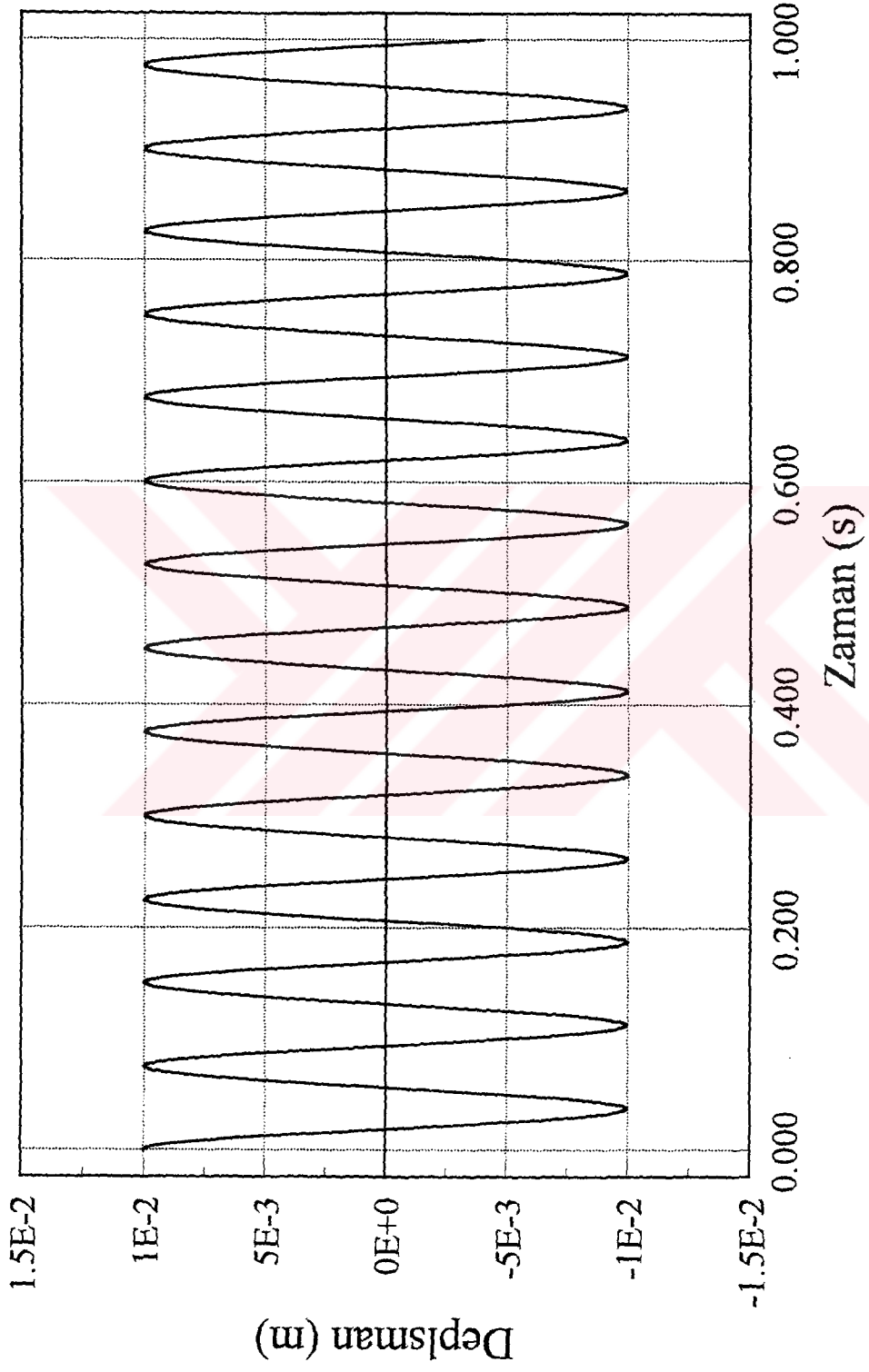
Adım aralığı (h)	$\omega^2$ (A)	Bağıl hata
$10^{-2}$	$10^2$	$4.10^{-6}$
$5.10^{-3}$	$2,7.10^3$	$2.10^{-2}$
$10^{-3}$	$5.10^3$	$10^{-6}$
$10^{-3}$	$3.10^4$	$4.10^{-3}$
$10^{-3}$	$5.10^4$	$3.10^{-2}$
$10^{-3}$	$5.10^5$	“K”
$10^{-4}$	$5.10^5$	$3.10^{-5}$

Tablo 2.4’deki değerler elde edilir.Kararsızlığa bir örnek olmak üzere,  $\omega^2 = 7000$ ,  $y(0)=0,0$  m için  $(0-1)s$  aralığında çözümlerse, Şek.2.7 elde edilir.Burada kullanılan adım aralığı  $h = 0,01s$ , çok büyük seçildiğinden yöntem, kararsız davranmıştır.  $h = 0,0025s$  ile yapılan çözümde ise Şek 2.8 elde edilmiştir.



Şekil 2.7 - BSD'li sistemin cevabı





Şekil 2.8 - BSD'li sistemin cevabı

### 3 SPEKTRAL ÇÖZÜM YÖNTEMİ

#### 3.1 GİRİŞ

Yukarıda, paragraf 2.1'de kısaca açıklandığı üzere; spektral çözüm yönteminde, öncelikle,  $n$  sayıda bağlantılı hareket denklemi yine  $n$  sayıda fakat bağlantısız hareket denklemine indirgenmektedir; bunun için doğal koordinatlar kullanılmaktadır. Böylece Bir Serbestlik Dereceli (BSD) sisteminkine analog denklemler elde edilmektedir. Elde edilen bu denklemlerin her biri için bulunan maksimum değerler o moda ait maksimumlar olmaktadır. Bu aşamadan sonra ise bulunan bu modal maksimum değerler, tercih edilen bir toplama (süperpozisyon, kombinasyon) yöntemiyle değerlendirilerek sonuç bulunmaktadır.

#### 3.2. HAREKET DENKLEMİNİN ÇÖZÜMÜ

(2.7) ve (2.8) ile verilen hareket denklemlerinin,

$$\{y\} = \{\phi\} \sin(\omega t + \theta) \quad (3.1)$$

şeklinde bir çözümü olduğu varsayılırsa; (2.8) ile,

$$-\omega^2 \cdot [m] \cdot \{\phi\} \sin(\omega t + \theta) + [k] \cdot \{\phi\} \cdot \sin(\omega t + \theta) = \{0\} \quad (3.2)$$

bulunur. Basitleştirilirse;  $\{\phi\}$  katsayı vektörünün sıfırdan farklı çözümü olabilmesi için,

$$|[k] - \omega^2 \cdot [m]| = 0 \quad (3.3)$$

olması gerekir; buna karakteristik denklem (frekans denklemi) denir. Bu denklemin,  $n$  tane pozitif,  $n$  tane de negatif kökü vardır. Bunların yalnızca pozitif olanlarının fiziksel anlamı bulunmaktadır; bunlar da yapı modal frekanslarıdır. Buradan genel çözümün, (3.1) gibi değil,

$$\{y\} = \sum_{j=1}^n \{\phi\}_j \cdot \sin(\omega_j t + \theta_j) \quad (3.4)$$

şeklinde olması gerektiği anlaşılmaktadır. Diğer bir gösterimle;

$$\{y\} = [\phi] \cdot \{Y\} \quad (3.4a)$$

demektir. Burada  $\{\theta\}_j$  modal vektör olarak adlandırılır ise  $\{Y\}$  genelleştirilmiş koordinatlar vektörü olarak adlandırılmaktadır. Sönümsüz bir sistemin genelleştirilmiş koordinatlardaki hareket denklemleri şöyledir:

$$\begin{aligned} [m] \cdot \{\ddot{y}\} + [k] \cdot \{y\} &= -[m] \cdot \{I\} \cdot \ddot{x}_g \\ [m] \cdot [\phi] \{\ddot{Y}\} + [k] \cdot [\phi] \cdot \{Y\} &= -[m] \cdot \{I\} \cdot \ddot{x}_g \end{aligned} \quad (3.5)$$

Denklemin her iki tarafı  $\{\phi\}_j^T$  ile çarpılıp ortagonallik özelliği kullanılırsa;

$$\begin{aligned} \{\phi\}_j^T \cdot [m] \cdot [\phi] \cdot \{\ddot{Y}\} + \{\phi\}_j^T \cdot [k] \cdot [\phi] \cdot \{Y\} &= -\{\phi\}_j^T \cdot [m] \cdot \{I\} \cdot \ddot{x}_g \\ \{\phi\}_j^T \cdot [m] \cdot \{\phi\}_j \cdot \{\ddot{Y}\} + \{\phi\}_j^T \cdot [k] \cdot \{\phi\}_j \cdot \{Y\} &= -\{\phi\}_j^T \cdot [m] \cdot \{I\} \cdot \ddot{x}_g \end{aligned}$$

bulunur.

$$\begin{aligned} M_j &= \{\phi\}_j^T \cdot [m] \cdot \{\phi\}_j \\ K_j &= \{\phi\}_j^T \cdot [k] \cdot \{\phi\}_j \\ q_j &= \left( -\{\phi\}_j^T \cdot [m] \cdot \{I\} \right) / M_j \end{aligned}$$

kısaltmalarıyla; j-ci moda ait hareket denklemi, BSD sistemin hareket denkleminde benzer olarak;

$$\begin{aligned} \ddot{Y}_j + \left( K_j / M_j \right) \cdot Y_j &= -q_j \cdot \ddot{x}_g \\ \ddot{Y}_j + \omega_j^2 \cdot Y_j &= -q_j \cdot \ddot{x}_g \end{aligned} \quad (3.6)$$

şeklinde elde edilmiş olur. Sönüm matrisinin de ortogonallik özelliğine sahip olduğu varsayılarak;

$$C_j = \{\phi\}_j^T \cdot [c] \cdot \{\phi\}_j$$

kısaltmasıyla; (3.6) genelleştirilerek,

$$\ddot{Y}_j + \left( C_j / M_j \right) \cdot \dot{Y}_j + \omega_j^2 \cdot Y_j = -q_j \cdot \ddot{x}_g$$

şeklinde yazılabilir.

Elde bir deprem ve BSD sistem için, sönüm oranını da dikkate alan ve yapının frekans bazında maksimum cevaplarını (spectral responses) gösteren bir spektrum (bu bir tasarım spektrumu da olabilir) var ise; yapının  $Y_{j,max}$  modal maksimumlarını elde etmek kolaydır; örneğe,  $S_a$  spektrumu ile,

$$Y_{j,max} = \frac{q_i \cdot S_a(\omega_j, \xi_i)}{\omega_j^2} \quad (3.7)$$

şeklinde; ve, geçek (geometrik) modal maksimum cevap (respons) ise,

$$\{y_j\}_{max} = \{\phi_j\}_{max} \cdot Y_{j,max} \quad (3.8)$$

şeklinde elde edilebilir. (3.8)'deki toplama işlemi kesin doğrudur. Bununla birlikte bütün modlar eşzamanlı olarak üst üste gelmezler. Modların toplanmasına ilişkin bazı yöntemler aşağıdaki gibidir.[1,2]

(1)  $Y_{j,max}$  ve  $\{y_j\}_{j,max}$  olasılıksal değişkenler olarak kabul edilmek üzere, toplama işlemi aşağıdaki gibi yapılır:

$$\{y\}_{SRSS} = \left[ \sum_{i=1}^r (\{\phi_i\} \cdot Y_i)^2 \right]^{1/2} \quad (3.9)$$

(r, dikkate alınması yeterli ilk mod sayısıdır.)

(2) Başka bir yöntem tam karesel toplama (Complete quadratic combination)

yöntemidir.  $U_k$  k-cı serbestlik derecesinin maksimum cevabı olmak üzere, formül aşağıdaki gibidir:

$$U_k = \left[ \sum_{i=1}^N \sum_{j=1}^N U_{ki} \cdot \rho_{ij} \cdot U_{kj} \right]^{1/2} \quad (3.10)$$

Burada  $U_{ki}$  ve  $U_{kj}$  k-cı serbestlik derecesinin i-ci ve j-ci modlardaki maksimum cevaplarıdır.  $\rho_{ij}$  modların karşılıklı etkilerini ifade eden katsayı olup aşağıdaki gibidir:

$$\rho_{ij} = \frac{8 \cdot \xi^2 (1+r) \cdot r^{3/2}}{(1-r^2)^2 + 4 \cdot \xi^2 \cdot r \cdot (1+r)^2} = \rho_{ji} \quad (3.11)$$

Burada  $r$ , dikkate alınması yeterli ilk mod sayısıdır. Sönüm oranının  $\xi = 0$  alınmasıyla SRSS ve CQC yöntemleri üst üste düşer.

(3) Hindistan yönetmeliğinde (1984) [2] toplama yöntemi için aşağıdaki formül verilmiştir:

$$V_i = (1-\gamma) \cdot \sum_{j=1}^r |V_{ij}| + \gamma \cdot \left[ \sum_{j=1}^r V_{ij}^2 \right]^{1/2} \quad (3.12)$$

şeklinde. Burada  $r$ , yukarıdaki ile aynı anlamdadır. Kat sayısına bağlı olan  $\gamma$ ,  $0 \leq \gamma \leq 1$  aralığında değişmektedir. Benzeri birleştirme (combination) ifadeleri diğer bazı yönetmeliklerde de yer alır. [2]

### 3.3 ÖRNEK YAPININ SPEKTRAL ÇÖZÜM YÖNTEMİ İLE HESABI

Paragraf 2.4.2'de belirlenen yapının spektral çözüm yöntemi ile çözümü için SAP90 (Educational Version) programı kullanılmıştır. Modal sönüm oranları birbirlerine eşit ve  $\xi = 0.05$  olarak kabul edilmiştir. Modların toplanmasında programın kullandığı yöntem, CQC dir. Programın giriş dosyası dökümü aşağıdaki gibidir.

```

SYSTEM
V=10
:
JOINTS
1 X=0 Y=0 Z=0
11 Z=35 G=1,11,1
:
RESTRAINTS
1 R=1,1,1,1,1,1
2 11 1 R=0,1,0,0,0,0
:

```

## MASSES

2 11 1 M=600,0,0,0,0,0 :[M]=kNs<sup>2</sup>/m/kat

:

FRAME :Çekirdek yapıda çevre kolonların etkisi ihmal edilmiştir.

NM=1

1 SH=B T=7,7,.3,.3 E=2.8E7

1 1 2 M=1 LP=2,0 G=9,1,1,1

:

## SPEC

A=0 S=1 D=0.05

0.00 4.9278

0.021969 4.9278

0.022820 4.9214

0.024377 4.9277

0.026901 4.9186

0.030929 4.9437

0.037601 4.9811

0.049878 5.0316

0.076860 5.2192

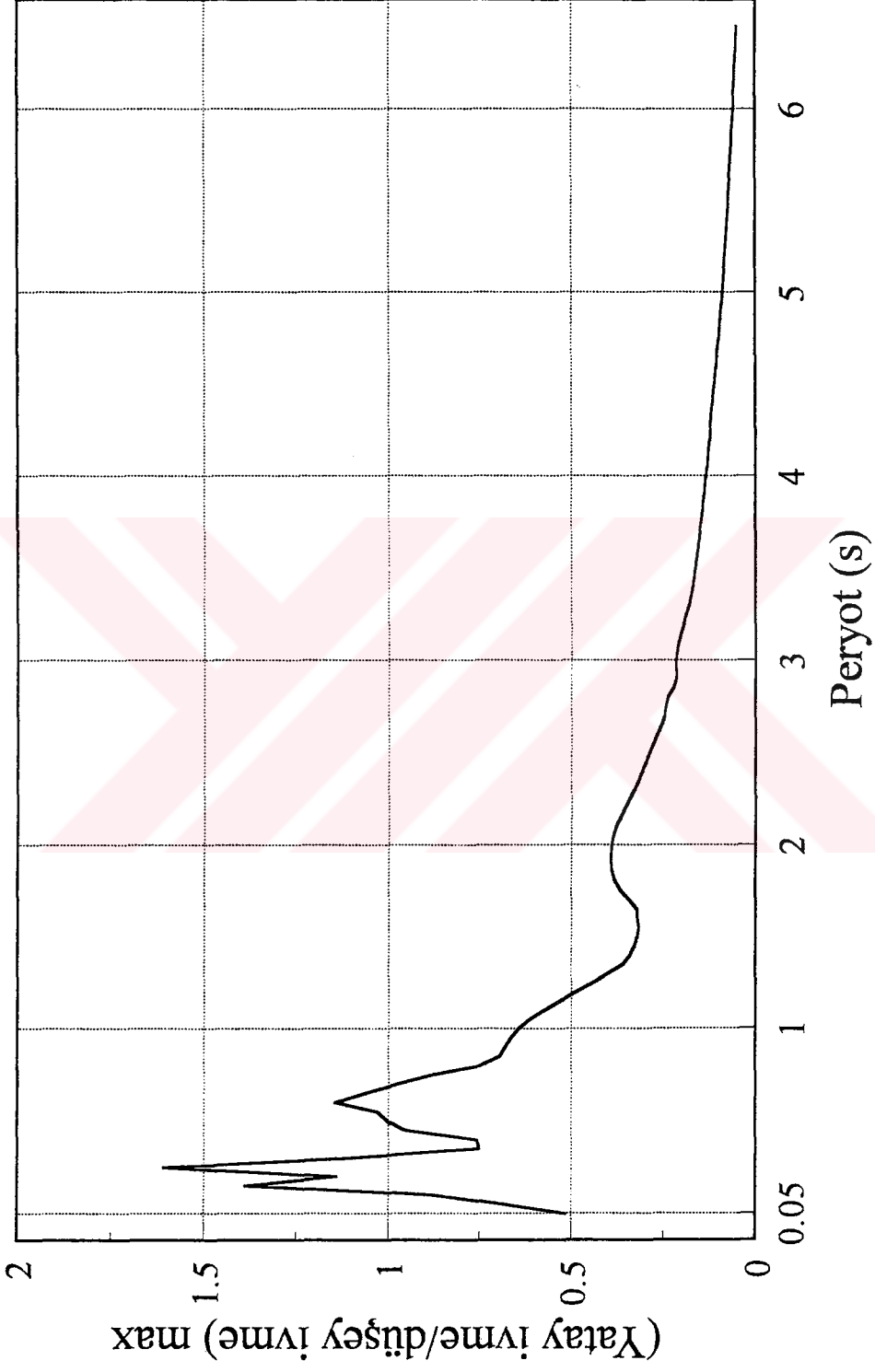
0.168696 8.8468

0.816121 8.2139

:

Spec data bloğundaki değerler, modal frekanslara karşılık gelen BSD sistemlerin Erzincan Depremi Doğu-Batı bileşenine verdikleri maksimum cevaplardır. Verilen 2. Program ile hesaplanan spektrum eğrisi, şekil 2.9'da görülmektedir. Fakat hassasiyeti artırmak için değerler grafikten okunmamış; her modal frekans için ayrı ayrı çözüm yapıp maksimumlar bulunmuştur. SAP90 Programının çalıştırılması sonucunda elde edilen kat kesme kuvvetleri, Tablo 3.1'de verilmiştir.

TABLO 3.1 Kat Kesme Kuvvetleri (kN)		
KAT	CQC	RMS
1	35192	35150
2	34653	34622
3	33481	33460
4	31714	31702
5	29447	29442
6	26733	26736
7	23500	23510
8	19528	19544
9	14496	14515
10	8069	8088



Şekil 2.9 - Spektral ivme (sönüm oranı=0.05)



#### 4. SONUÇ VE ÖNERİLER

Paragraf 2.4.2’de belirlenen yapının Erzincan Depremi Doğu-Batı bileşeni altında Spektral çözüm yöntemiyle çözümünden elde edilen kat kesme kuvvetleri Tablo 3.1’de verilmektedir. Aynı örnek için Zaman Artımı ile yapılan kesin çözümünden elde edilen değerler ise Tablo 2.1’de verilmektedir. Spektral çözüm yöntemiyle yapılan çözümdeki hata miktarları Zaman artımı ile yapılan çözümünden bulunan değerler doğru çözüm kabul edilmek üzere Tablo 4.1’de verilmiştir.

Kat	CQC yönteminde	RMS yönteminde
1	6.45	6.32
2	8.64	8.54
3	9.67	9.60
4	11.06	11.01
5	13.62	13.61
6	17.84	17.85
7	19.41	19.46
8	17.99	18.08
9	14.99	15.14
10	13.57	13.84

(1) Tablo 4.1’den izlendiği üzere spektral çözüm yöntemi (RMS ve CQC) gerçek değerlerden yüksek değerler vermektedir; Hatâ oranı, ayrıca, üst katlara gittikçe artmaktadır.

(2) Hatâ oranları, bu örnekte CQC için %15 civarında kalmaktadır. CQC ve RMS yöntemlerinin, Zaman artımı yöntemindeki zahmet gözönüne alınınca, pratik değeri olduğu açıktır. Bununla beraber, verilen örnek bir düzlem çerçeve (konsol) örneğidir; üç boyutlu örneklerde daha farklı sonuçlara ulaşmak mümkündür; bu bakımdan bu sonuçları, bu aşamada ihtiyatla karşılamak gerekir.

(3) CQC ve RMS yöntemlerinin, burulmalı sistemlerde çok farklı sonuçlar verdiği bilinir. [2] Burada, sonuçların yakınlığı, örneğin düzlem çerçeve alınmasından kaynaklanır. Çalışmanın, üç boyutlu örneklerle zenginleştirilmesi, daha güvenilir genellemeler yapmaya imkan verecektir.



## PROGRAM - 1



**- PROGRAMIN GİRDİLERİ:**

- 1) "A" isimli dosya (Bölüm 2'de tanımlandı)
- 2) "B" isimli dosya (Bölüm 2'de tanımlandı)
- 3) "Y0" isimli dosya (Bölüm 2'de tanımlandı)
- 4) "T" isimli dosya. Sütun vektör olarak tanımlıdır. Çözümün yapılacağı zaman eksenindeki değerleri içerir.
- 5) "U" isimli dosya. Sütun vektör olarak tanımlıdır. "T" vektörüyle aynı boyuttadır ve karşılık gelen ivme değerlerini içerir.

**- PROGRAMIN ÇIKTILARI:**

- 1) "MAXY" isimli dosya. Her katın bağıl maksimum deplasmanlarını ve karşılık gelen zaman değerlerini içerir.
- 2) "TT" isimli dosya. Hesap yapılan zaman adımlarını içerir.
- 3) "YY" isimli dosya. "TT" matrisiyle aynı satır sayısına serbestlik derecesi kadarda sütun sayısına sahip olan çözüm dosyasıdır.
- 4) "IVME" isimli dosya. Sütun matristir. Her serbestlik derecesi kadarki satırı bir zaman adımına karşılık gelir. Serbestlik derecelerinin ivmelerini içerir.

```

program Time-History(okuma,okumat,okumau,yazmay,yazmat,yazmai);
uses crt,graph;
type
  dizi19=array [1..19,1..19] of extended;
  dizi18=array [1..18,1..18] of extended;
var
  okuma,okumat,okumau,yazmay,yazmat,yazmai:text;
  a:array [1..50,1..50] of extended;
  b:array [1..50] of extended;
  y:array [1..50,1..19] of extended;
  ymax:array [1..50] of extended;
  tmax:array [1..50] of extended;
  f:array [1..50,1..19] of extended;
  fb:array [1..50] of extended;
  u,t:array [0..102] of extended;
  time,uu:extended;
  rc,boyut,derece,zz,yarisi,ilkde,son,atlama:integer;
  n,nt,ntt,i,j,adim:longint;
  ksi,w,h:extended;
  kb:dizi18;
  pas,km:dizi19;

procedure pascl(var pas:dizi19;n:integer);
var
  i,j,s:integer;

begin
  pas[1,1]:=1;pas[2,1]:=1;pas[2,2]:=-1;
  for i:=3 to n do
    begin
      s:=1;
      for j:=2 to i-1 do
        begin
          s:=-s;
          pas[i,j]:= (abs(pas[i-1,j-1])+abs(pas[i-1,j]))*s;
        end;
      pas[i,1]:=1;pas[i,i]:=-s;
    end;
end;

procedure katsayib(b:dizi19;var kb:dizi18;n:integer);
var

```

```

i,j,k:integer;
adb:array [0..18] of extended;
begin
for i:=1 to 18 do
begin
for j:=1 to i do
begin
kb[i,j]:=0;
end;
end;
adb[1]:=1.0;
adb[2]:=1.0/2.0;
adb[3]:=5.0/12.0;
adb[4]:=3.0/8.0;
adb[5]:=251.0/720.0;
adb[6]:=95.0/288.0;
adb[7]:=19087.0/60480.0;
adb[8]:=5257.0/17280.0;
adb[9]:=1070017.0/3628800.0;
adb[10]:=25713.0/89600.0;
adb[11]:=26842253.0/95800320.0;
adb[12]:=4777223.0/17418240.0;
adb[13]:=703604254357.0/2615348736000.0;
adb[14]:=106364763817.0/402361344000.0;
adb[15]:=1166309819657.0/4483454976000.0;
adb[16]:=25221445.0/98402304.0;
adb[17]:=8092989203533249.0/32011868528640000.0;
adb[18]:=85455477715379.0/34237292544000.0;
for i:=1 to n do
begin
for j:=1 to i do
begin
for k:=1 to j do
begin
kb[i,i-k+1]:=kb[i,i-k+1]+adb[j]*b[j,k];
end;
end;
end;
end;
end;

```

```

procedure katsayim(b:dizi19;var km:dizi19;n:integer);
var
i,j,k:integer;
adm:array [0..19] of extended;

```

```

begin
for i:=1 to 19 do
begin
for j:=1 to i do
begin
km[i,j]:=0;
end;
end;
adm[1]:=-1.0;
adm[2]:=1.0/2.0;
adm[3]:=1.0/12.0;
adm[4]:=1.0/24.0;
adm[5]:=19.0/720.0;
adm[6]:=3.0/160.0;
adm[7]:=863.0/60480.0;
adm[8]:=275.0/24192.0;
adm[9]:=33953.0/3628800.0;
adm[10]:=8183.0/1036800.0;
adm[11]:=3250433.0/479001600.0;
adm[12]:=4671.0/788480.0;
adm[13]:=13695779093.0/2615348736000.0;
adm[14]:=2224234463.0/475517952000.0;
adm[15]:=132282840127.0/31384184832000.0;
adm[16]:=2639651053.0/689762304000.0;
adm[17]:=111956703448001.0/32011868528640000.0;
adm[18]:=50188465.0/15613165568.0;
adm[19]:=2334028946344463.0/786014494949376000.0;
for i:=1 to n do
begin
for j:=1 to i do
begin
for k:=1 to j do
begin
km[i,i-k+1]:=km[i,i-k+1]-adm[j]*b[j,k];
end;
end;
end;
end;
end;

```

```

procedure dosyaac;
begin
assign(okumat,'c:\bilgi\t'); {T ZAMAN VEKTÖRÜ DOSYASI}
reset(okumat);
assign(okumau,'c:\bilgi\u'); {U ZAMAN VEKTÖRÜ DOSYASI}

```

```
reset(okumau);
assign(yazmay,'c:\bilgi\yy');
assign(yazmat,'c:\bilgi\tt');
assign(yazmai,'c:\bilgi\ivme');
rewrite(yazmay);
rewrite(yazmat);
rewrite(yazmai);
end;
```

```
procedure maxyaz;
var
  i:integer;
begin
  assign(yazmay,'c:\bilgi\maxy');
  rewrite(yazmay);
  for i:=1 to boyut do
    begin
      writeln(yazmay,tmax[i],ymax[i]);
    end;
end;
```

```
procedure cikis;
begin
  close(yazmay);
  close(yazmat);
  maxyaz;
  close(yazmay);
  close(yazmai);
  close(okumat);
  close(okumau);
  halt;
end;
{PROGRAMDAN ÇIKIŞ BURADANDIR}
```

```
procedure cik(d:string);
begin
  writeln(d);
  halt;
end;
```



```
procedure okumaby0(var boyut:integer);
```

```
var
  i,j,ks:integer;
  d,da:string;
```

```
procedure bul(a:string;n,i:integer;var b:string);
```

```
var
  bas,son:integer;
```

```
procedure basbul(var i:integer);
```

```
var
  kar:string;
begin
  repeat
    kar:=copy(a,i,1);
    i:=i+1;
  until (kar<>' ') or (i>length(a));
  i:=i-1;
  if (kar=' ') or (kar='') then rc:=1;
end;
```

```
procedure sonbul(var i:integer);
```

```
var
  kar:string;
begin
  kar:=copy(a,i,1);
  while (kar<>' ') and (i<length(a)) do
    begin
      i:=i+1;
      kar:=copy(a,i,1);
    end;
  if kar=' ' then i:=i-1;
end;
```

```
begin
  rc:=0;
  repeat
    basbul(i);
    bas:=i;
    son:=i;
    sonbul(son);
```

```

i:=son+1;
n:=n-1;
until (n=0) or (rc<>0);
b:=copy(a,bas,son-bas+1);
end;

```

```

procedure kesabul(d:string;var k:integer);

```

```

var
i:integer;
da:string;
begin
i:=0;rc:=0;
while rc=0 do
begin
i:=i+1;
bul(d,i,1,da);
end;
k:=i-1;
end;

```

```

begin
assign(okuma,'c:\bilgi\A'); {A MATRİSİNİ OKU}
reset(okuma);
readln(okuma,d);
kesabul(d,boyut);
rc:=0;
for i:=1 to boyut do
begin
kesabul(d,ks);
if ks<>boyut then cik('A matrisi kare olmalı !');
for j:=1 to boyut do
begin
bul(d,j,1,da);
if rc<>0 then cik('A matrisi kare olmalı !');
val(da,a[i,j],rc);
end;
readln(okuma,d);
end;
close(okuma);
assign(okuma,'c:\bilgi\B'); {B VEKTÖRÜNÜ OKU}
reset(okuma);
i:=0;

```

```

while not eof(okuma) do
begin
  readln(okuma,d);
  kesabul(d,zz);
  case zz of
    0: ;
    1: begin
      i:=i+1;
      val(d,b[i],rc);
      end;
    else cik('B bir vektör matris olmalı !');
      end;
  end;
  if i > boyut then cik('B vektörü satır sayısı, A matrisinin satır yada sütun sayısına eşit
olmalı !');
  close(okuma);
  assign(okuma,'c:\bilgi\y0'); {Y0 BASLANGIÇ ŞARTLARI VEKTÖRÜNÜ OKU}
  reset(okuma);
  i:=0;
  while not eof(okuma) do
  begin
    readln(okuma,d);
    kesabul(d,zz);
    case zz of
      0: ;
      1: begin
        i:=i+1;
        val(d,y[i,1],rc);
        end;
      else cik('Y0 bir vektör matris olmalı !');
        end;
      end;
      if i > boyut then cik('Y0 vektörü satır sayısı, A matrisinin satır yada sütun sayısına eşit
olmalı !');
      close(okuma);
      end;

procedure okutu;
var
  d:string;

procedure bul(a:string;n,i:integer;var b:string);
var
  bas,son:integer;

```

```

procedure basbul(var i:integer);
var
  kar:string;
begin
  repeat
    kar:=copy(a,i,1);
    i:=i+1;
  until (kar<>' ') or (i>length(a));
  i:=i-1;
  if (kar=' ') or (kar='') then rc:=1;
end;

```

```

procedure sonbul(var i:integer);
var
  kar:string;
begin
  kar:=copy(a,i,1);
  while (kar<>' ') and (i<length(a)) do
  begin
    i:=i+1;
    kar:=copy(a,i,1);
  end;
  if kar=' ' then i:=i-1;
end;

```

```

begin
  rc:=0;
  repeat
    basbul(i);
    bas:=i;
    son:=i;
    sonbul(son);
    i:=son+1;
    n:=n-1;
  until (n=0) or (rc<>0);
  b:=copy(a,bas,son-bas+1);
end;

```

```

procedure kesabul(d:string;var k:integer);
var
  i:integer;
  da:string;
begin

```

```

i:=0;rc:=0;
while rc=0 do
begin
i:=i+1;
bul(d,i,1,da);
end;
k:=i-1;
end;

```

```

begin
if ilkde=1 then          {İLK DEFA GELİNMIYORSA}
begin
t[0]:=time;             {ÖNCEKİ DATAYA BAĞLA}
u[0]:=uu;               {ÖNCEKİ DATAYA BAĞLA}
n:=0;                   {SAYACI AYARLA}
end;
if ilkde<>1 then n:=1;   {İLK DEFA GELİNİYORSA}
i:=0;                   {T MATRİSİNİ OKU}
while (not eof(okumat)) and (i<ntt) do
begin
readln(okumat,d);
kesabul(d,zz);
case zz of
0: ;
1: begin
i:=i+1;
val(d,t[i],rc);
end;
else cik('T bir vektör matris olmalı !');
end;
end;
nt:=i;
i:=0;                   {U MATRİSİNİ OKU}
while (not eof(okumau)) and (i<ntt) do
begin
readln(okumau,d);
kesabul(d,zz);
case zz of
0: ;
1: begin
i:=i+1;
val(d,u[i],rc);
end;
else cik('T bir vektör matris olmalı !');
end;
end;

```

```

end;
if nt <> i then cik('T ile U vektörleri, aynı boyutta olmalılar !');
end;

```

```

procedure yazy(y:extended;i,boyut:integer);
begin
if (round(adim/atlama)*atlama-adim=0) or (adim=1) then
case i <> boyut of
true:write(yazmay,' ',y);
false:writeln(yazmay,' ',y);
end;
end;
end;

```

```

procedure yazt;
begin
if (round(adim/atlama)*atlama-adim=0) or (adim=1) then writeln(yazmat,time);
end;

```

```

procedure yazi;
var
i:integer;
begin
for i:=1 to boyut do
begin
if (round(adim/atlama)*atlama-adim=0) or (adim=1) then writeln(yazmai,fb[i]);
end;
end;
end;

```

```

procedure karar;
begin
h:=0.00005;
derece:=5;
atlama:=100;
ntt:=100;
end;

```

```

procedure turev(boyut:integer;k:longint);
var

```

```

i,j:integer;
begin
uu:=(u[n+1]-u[n])/(t[n+1]-t[n])*(time-t[n])+u[n]; {İkinci taraf bulunumunda,}
for i:=1 to boyut do {lineer yaklaşım}
begin
f[i,k]:=0;
fb[i]:=0;
for j:=1 to boyut do
begin
f[i,k]:=f[i,k]+a[i,j]*y[j,k];
end;
fb[i]:=f[i,k]; {uzaydaki referans eksenine göre ivme}
f[i,k]:=f[i,k]+b[i]*uu; {tabana göre ivme}
end;
end;
end;

```

```

procedure tbul;
begin
if ((nt-n)<=1) and (abs(t[nt]-time)<h/2) then okutu;
if nt=0 then cikis;
if time+h>t[n+1] then n:=n+1;
time:=time+h;
adim:=adim+1;
end;

```

```

procedure deayarla;
begin
ilkde:=1;
adim:=1;
yarisi:=round(boyut/2);
for i:=1 to boyut do
begin
ymax[i]:=0.0;
end;
time:=t[1];
end;

```

```

procedure deayar;
begin
ilkde:=0;
end;

```

```

procedure basyaz;
var
  i:integer;
begin
  yazt;
  for i:=1 to boyut do
    begin
      yazy(y[i,1],i,boyut);
    end;
  end;
end;

```

```

procedure baslat;
var
  i,j,k:integer;
begin
  turev(boyut,1);
  yazi;
  for i:=1 to boyut do
    begin
      case (i=1) or (i=yarisi+1) of
        {M}
        true: if abs(y[i,1])>ymax[i] then
          {A}
          begin
            {X}
            ymax[i]:=abs(y[i,1]);tmax[i]:=time;
          end;
        false: if abs(y[i,1]-y[i-1,1])>ymax[i] then
          begin
            ymax[i]:=abs(y[i,1]-y[i-1,1]);tmax[i]:=time; {B}
          end;
          {U}
          {L}
      end;
    end;
  end;
  for j:=1 to derece do
    begin
      tbul;
      yazt;
      for i:=1 to boyut do
        begin
          y[i,j+1]:=y[i,j];
          for k:=1 to j do
            begin
              y[i,j+1]:=y[i,j+1]+h*kb[j,k]*f[i,k];
            end;
          end;
        end;
      turev(boyut,j+1);
      for i:=1 to boyut do
        begin

```



```

y[i,j+1]:=y[i,j];
for k:=1 to j+1 do
  begin
    y[i,j+1]:=y[i,j+1]+h*km[j+1,k]*f[i,k];
  end;
yazy(y[i,j+1],i,boyut);
case (i=1) or (i=yarisi+1) of           {M}
  true: if abs(y[i,j+1])>ymax[i] then    {A}
    begin                                {X}
      ymax[i]:=abs(y[i,j+1]);tmax[i]:=time;
    end;
  false: if abs(y[i,j+1]-y[i-1,j+1])>ymax[i] then
    begin
      ymax[i]:=abs(y[i,j+1]-y[i-1,j+1]);tmax[i]:=time; {B}
    end;                                  {U}
end;                                       {L}
end;
end;
turev(boyut,j+1);
yazi;
end;
end;

```

```

procedure kaydir;
  var
    i,j:integer;
  begin
    for i:=1 to boyut do
      begin
        for j:=2 to derece+1 do
          begin
            f[i,j-1]:=f[i,j];
            y[i,j-1]:=y[i,j];
          end;
        end;
      end;
    end;
end;

```

```

procedure hesapla;
  var
    i,k:integer;
  begin
    tbul;
    yazt;
    for i:=1 to boyut do
      begin
        y[i,derece+1]:=y[i,derece];
        for k:=1 to derece do

```

```

begin
  y[i,derece+1]:=y[i,derece+1]+h*kb[derece,k]*f[i,k];
end;
end;
turev(boyut,derece+1);
for i:=1 to boyut do
begin
  y[i,derece+1]:=y[i,derece];
  for k:=1 to derece+1 do
  begin
    y[i,derece+1]:=y[i,derece+1]+h*km[derece+1,k]*f[i,k];
  end;
  yazy(y[i,derece+1],i,boyut);
  case (i=1) or (i=yarisi+1) of
    true: if abs(y[i,derece+1])>ymax[i] then
      begin
        ymax[i]:=abs(y[i,derece+1]);tmax[i]:=time;
      end;
    false: if abs(y[i,derece+1]-y[i-1,derece+1])>ymax[i] then
      begin
        ymax[i]:=abs(y[i,derece+1]-y[i-1,derece+1]);tmax[i]:=time;
      end;
  end;
end;
end;
turev(boyut,derece+1);
yazi;
end;

```

```

begin
  clrscr;
  pascl(pas,19);
  katsayib(pas,kb,18);
  katsayim(pas,km,19);
  dosyaac;
  okumaby0(boyut);
  karar;
  deayar;
  okutu;
  deayarla;
  basyaz;
  baslat;
  repeat
    kaydir;
    hesapla;
  until 6<>6;
end.

```

PROGRAM - 2



**- PROGRAMIN GİRDİLERİ:**

- 1) “T” isimli dosya. Sütun vektördür. Time-history’de kullanılacak zaman eksenini içerir.
- 2) “U” isimli dosya. Sütun vektördür. “T” isimli dosyadaki zaman adımlarına karşılık gelen ivme değerlerini içerir.

**- PROGRAMIN ÇIKTILARI:**

- 1) “SPCT” isimli dosya. Sütun vektördür. Hesap yapılan periyot değerlerini içerir.
- 2) “SPCY” isimli dosya. Satır sayısı “SPCT” dosyası ile aynıdır. Sütun sayısı üç tür. Sırasıyla, karşılık gelen periyot değerlerine ait deplasman, hız ve normalize edilmiş ivmenin maksimum (spektral) değerlerini içerir.



```

program spektrum(okuma,yazmay,yazmat);
uses crt,graph;
type
  dizi19=array [1..19,1..19] of extended;
  dizi18=array [1..18,1..18] of extended;
var
  okuma,yazmay,yazmat:text;
  a:array [1..10,1..10] of extended;
  b:array [1..10] of extended;
  y:array [1..10,1..19] of extended;
  ymax:array [1..10] of extended;
  tmax:array [1..10] of extended;
  f:array [1..10,1..19] of extended;
  u,t:array [1..2200] of extended;
  fmax,uu:extended;
  rc,boyut,derece,zz,yarisi,cikma:integer;
  n,nt,i,j:longint;
  time,ksi,w,h,periyod:extended;
  kb:dizi18;
  pas,km:dizi19;

procedure pascl(var pas:dizi19;n:integer);
var
  i,j,s:integer;

begin
  pas[1,1]:=1;pas[2,1]:=1;pas[2,2]:=-1;
  for i:=3 to n do
    begin
      s:=1;
      for j:=2 to i-1 do
        begin
          s:=-s;
          pas[i,j]:=(abs(pas[i-1,j-1])+abs(pas[i-1,j]))*s;
        end;
      pas[i,1]:=1;pas[i,i]:=-s;
    end;
end;

procedure katsayib(b:dizi19;var kb:dizi18;n:integer);
var
  i,j,k:integer;

```

```

adb:array [0..18] of extended;
begin
for i:=1 to 18 do
begin
for j:=1 to i do
begin
kb[i,j]:=0;
end;
end;
adb[1]:=1.0;
adb[2]:=1.0/2.0;
adb[3]:=5.0/12.0;
adb[4]:=3.0/8.0;
adb[5]:=251.0/720.0;
adb[6]:=95.0/288.0;
adb[7]:=19087.0/60480.0;
adb[8]:=5257.0/17280.0;
adb[9]:=1070017.0/3628800.0;
adb[10]:=25713.0/89600.0;
adb[11]:=26842253.0/95800320.0;
adb[12]:=4777223.0/17418240.0;
adb[13]:=703604254357.0/2615348736000.0;
adb[14]:=106364763817.0/402361344000.0;
adb[15]:=1166309819657.0/4483454976000.0;
adb[16]:=25221445.0/98402304.0;
adb[17]:=8092989203533249.0/32011868528640000.0;
adb[18]:=85455477715379.0/34237292544000.0;
for i:=1 to n do
begin
for j:=1 to i do
begin
for k:=1 to j do
begin
kb[i,i-k+1]:=kb[i,i-k+1]+adb[j]*b[j,k];
end;
end;
end;
end;
end;

```

```

procedure katsayim(b:dizi19;var km:dizi19;n:integer);
var
i,j,k:integer;
adm:array [0..19] of extended;
begin

```

```

for i:=1 to 19 do
begin
for j:=1 to i do
begin
km[i,j]:=0;
end;
end;
adm[1]:=-1.0;
adm[2]:=1.0/2.0;
adm[3]:=1.0/12.0;
adm[4]:=1.0/24.0;
adm[5]:=19.0/720.0;
adm[6]:=3.0/160.0;
adm[7]:=863.0/60480.0;
adm[8]:=275.0/24192.0;
adm[9]:=33953.0/3628800.0;
adm[10]:=8183.0/1036800.0;
adm[11]:=3250433.0/479001600.0;
adm[12]:=4671.0/788480.0;
adm[13]:=13695779093.0/2615348736000.0;
adm[14]:=2224234463.0/475517952000.0;
adm[15]:=132282840127.0/31384184832000.0;
adm[16]:=2639651053.0/689762304000.0;
adm[17]:=111956703448001.0/32011868528640000.0;
adm[18]:=50188465.0/15613165568.0;
adm[19]:=2334028946344463.0/786014494949376000.0;
for i:=1 to n do
begin
for j:=1 to i do
begin
for k:=1 to j do
begin
km[i,i-k+1]:=km[i,i-k+1]-adm[j]*b[j,k];
end;
end;
end;
end;
end;

```

```

procedure dosyaac;
begin
assign(yazmay,'c:\bilgi\spcy');
assign(yazmat,'c:\bilgi\spct');
rewrite(yazmay);
rewrite(yazmat);

```

```
end;
```

```
procedure cik(d:string);
begin
  writeln(d);
  halt;
end;
```

```
procedure oku(var boyut:integer;var n:longint);
var
  i,j,ks:integer;
  d,da:string;
```

```
procedure bul(a:string;n,i:integer;var b:string);
var
  bas,son:integer;
```

```
procedure basbul(var i:integer);
var
  kar:string;
begin
  repeat
    kar:=copy(a,i,1);
    i:=i+1;
  until (kar<>' ') or (i>length(a));
  i:=i-1;
  if (kar=' ') or (kar='') then rc:=1;
end;
```

```
procedure sonbul(var i:integer);
var
  kar:string;
begin
  kar:=copy(a,i,1);
  while (kar<>' ') and (i<length(a)) do
  begin
    i:=i+1;
    kar:=copy(a,i,1);
  end;
  if kar=' ' then i:=i-1;
end;
```



```

begin
  rc:=0;
  repeat
    basbul(i);
    bas:=i;
    son:=i;
    sonbul(son);
    i:=son+1;
    n:=n-1;
  until (n=0) or (rc<>0);
  b:=copy(a,bas,son-bas+1);
end;

```

```

procedure kesabul(d:string;var k:integer);
var
  i:integer;
  da:string;
begin
  i:=0;rc:=0;
  while rc=0 do
    begin
      i:=i+1;
      bul(d,i,1,da);
    end;
    k:=i-1;
  end;

```

```

begin
  assign(okuma,'c:\bilgi\t'); {T ZAMAN VEKTÖRÜNÜ OKU}
  reset(okuma);
  i:=0;
  while not eof(okuma) do
    begin
      readln(okuma,d);
      kesabul(d,zz);
      case zz of
        0: ;
        1: begin
            i:=i+1;
            val(d,t[i],rc);

```

```

    end;
    else cik('T bir vektör matris olmalı !');
    end;
    end;
    nt:=i;
    close(okuma);
    assign(okuma,'c:\bilgi\u'); {U ZAMAN VEKTÖRÜNÜ OKU}
    reset(okuma);
    i:=0;
    while not eof(okuma) do
    begin
        readln(okuma,d);
        kesabul(d,zz);
        case zz of
        0: ;
        1: begin
            i:=i+1;
            val(d,u[i],rc);
            end;
        else cik('U bir vektör matris olmalı !');
        end;
        end;
    if nt<>i then cik('T ile U vektörleri, aynı boyutta olmalılar !');
    close(okuma);
    end;

```

```

procedure ayap;
begin
    w:=2*3.141592654/periyod;
    a[2,1]:=-w*w;a[2,2]:=-2*ksi*w;
    end;

```

```

procedure karar;
begin
    if periyod<0.088858 then h:=1e-4;
    if (periyod>0.08858) and (periyod<0.62832) then h:=1e-3;
    if periyod>0.62832 then h:=1e-2;
    if h>t[2]-t[1] then h:=t[2]-t[1];
    derece:=5;
    end;

```

```

procedure turev(boyut:integer;k:longint);

```

```

var
  i,j:integer;
begin
  uu:=(u[n+1]-u[n])/(t[n+1]-t[n])*(time-t[n])+u[n]; {İkinci taraf bulunumunda,}
  for i:=1 to boyut do {lineer yaklaşım}
  begin
    f[i,k]:=0;
    for j:=1 to boyut do
    begin
      f[i,k]:=f[i,k]+a[i,j]*y[j,k];
    end;
    f[i,k]:=f[i,k]+b[i]*uu;
  end;
end;

```

```

procedure tbul;
var b:extended;
begin
  case ((nt-n)<=1) and (abs(t[nt]-time)<h/2) of
    true: cikma:=1;
    false: begin
      if time+h>t[n+1] then n:=n+1;
      time:=time+h;
    end;
  end;
end;

```

```

procedure deayar;
begin
  boyut:=2;
  a[1,1]:=0;a[1,2]:=1;
  b[1]:=0;
  b[2]:=-1;
  ksi:=0.05;
end;

```

```

procedure deayarla;
begin
  cikma:=0;
  n:=1;
  time:=t[1];

```

```

yarisi:=1;
ymax[1]:=0.0;
ymax[2]:=0.0;
y[1,1]:=0;
y[2,1]:=0;
fmax:=0;
end;

```

```

procedure baslat;

```

```

var

```

```

i,j,k:integer;

```

```

begin

```

```

turev(boyut,1);

```

```

if abs(f[2,1]+uu)>fmax then fmax:=abs(f[2,1]+uu); {MAX BUL}

```

```

for i:=1 to boyut do

```

```

begin

```

```

if abs(y[i,1])>ymax[i] then {MAX}

```

```

begin

```

```

ymax[i]:=abs(y[i,1]);

```

```

end; {BUL}

```

```

end;

```

```

for j:=1 to derece do

```

```

begin

```

```

tbul;

```

```

for i:=1 to boyut do

```

```

begin

```

```

y[i,j+1]:=y[i,j];

```

```

for k:=1 to j do

```

```

begin

```

```

y[i,j+1]:=y[i,j+1]+h*kb[j,k]*f[i,k];

```

```

end;

```

```

end;

```

```

turev(boyut,j+1);

```

```

for i:=1 to boyut do

```

```

begin

```

```

y[i,j+1]:=y[i,j];

```

```

for k:=1 to j+1 do

```

```

begin

```

```

y[i,j+1]:=y[i,j+1]+h*km[j+1,k]*f[i,k];

```

```

end;

```

```

if abs(y[i,j+1])>ymax[i] then {MAX}

```

```

begin

```

```

ymax[i]:=abs(y[i,j+1]);

```

```

end;

```

```

{BUL}

```

```

end;

```

```

turev(boyut,j+1);
if abs(f[2,j+1]+uu)>fmax then fmax:=abs(f[2,j+1]+uu);  {MAX BUL}
end;
end;

```

```

procedure kaydir;
var
  i,j:integer;
begin
  for i:=1 to boyut do
    begin
      for j:=2 to derece+1 do
        begin
          f[i,j-1]:=f[i,j];
          y[i,j-1]:=y[i,j];
        end;
      end;
    end;
end;

```

```

procedure hesapla;
var
  i,k:integer;
begin
  tbul;
  case cikma <> 1 of
    true: begin
      for i:=1 to boyut do
        begin
          y[i,derece+1]:=y[i,derece];
          for k:=1 to derece do
            begin
              y[i,derece+1]:=y[i,derece+1]+h*kb[derece,k]*f[i,k];
            end;
          end;
          turev(boyut,derece+1);
          for i:=1 to boyut do
            begin
              y[i,derece+1]:=y[i,derece];
              for k:=1 to derece+1 do
                begin
                  y[i,derece+1]:=y[i,derece+1]+h*km[derece+1,k]*f[i,k];
                end;
              if abs(y[i,derece+1])>ymax[i] then  {MAX}
                begin

```

```

        ymax[i]:=abs(y[i,derece+1]);
        end;
        end;
        turev(boyut,derece+1);
{MAX BUL}   if abs(f[2,derece+1]+uu)>fmax then fmax:=abs(f[2,derece+1]+uu);
        end; {CASE TRUE}
        false;
        end; {CASE}
        end;

begin
  clrscr;
  pascl(pas,19);
  katsayib(pas,kb,18);
  katsayim(pas,km,19);
  oku(boyut,n);
  deayar;
  dosyaac;
  peryod:=0.00;
  while peryod<=6.5 do
  begin
    peryod:=peryod+0.05;
    ayap;
    karar;
    deayarla;
    writeln('peryod=',peryod,' h=',h);
    baslat;
    while cikma<>1 do
    begin
      kaydir;
      hesapla;
    end;
    writeln(yazmat,peryod);
    write(yazmay,ymax[1]);
    write(yazmay,ymax[2]);
    writeln(yazmay,fmax/9.81);
  end;
  close(yazmay);
  close(yazmat);
end.

```

**KAYNAKLAR**

- [1] Polat, Z., “İleri yapı dinamiđi ders notları”, Y.T.Ü., İstanbul, 1993.
- [2] Polat, Z., “Influence of earthquake direction on response of structures”, Mangalia, Romania, 1995
- [3] Erdik, M., Yüzügüllü, Ö., “Deprem mühendisliđi açısından yapı dinamiđine giriş”, ODTÜ, Ankara, 1980.
- [4] Clough, R., Penzien, J., “Dynamics of structures”, University of California, Berkeley, 1975
- [5] Scheid, F., “2000 solved problems in numerical analysis”, Boston Universty, 1990
- [6] Rischard, J., “Numerical methods, software and, analysis”, McGraw Hill Book Press, 1983
- [7] Çağal, B., “Sayısal analiz”, Y.T.Ü., İstanbul, 1989

## ÖZGEÇMİŞ

Doğum yeri : İstanbul

Doğum tarihi : 13.10.1970

Öğrenim hayatı : 1977 yılında Erenköy ilk okuluna başladım. Orta öğrenimimi Göztepe Orta Okulu'nda, lise öğrenimimi ise Tahran Lisesi (Göztepe)'de tamamladım. 1988 yılında Y.T.Ü. İnşaat Mühendisliği Bölümüne girdim. 1992 yılında aynı bölümden mezun oldum. 1993 yılında ise hâlen devam etmekte olduğum, Y.T.Ü. Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı Yapı Gurubunda Yüksek Lisans programına başladım.