



TÜRKİYE BİLİMSEL VE  
TEKNİK ARAŞTIRMA KURUMU

THE SCIENTIFIC AND TECHNICAL  
RESEARCH COUNCIL OF TURKEY

2001-17



**PREFABRİK ÇERÇEVE ve SÜNEK DUVARLARDAN**

**OLUŞAN YAPI SİSTEMLERİNİN DEPREM DAVRANIŞININ**

**ANALİTİK İNCELENMESİ**

Proje No: 195 I 001

**PROJE NO: İNTAG 542**

İnşaat Teknolojileri Araştırma Grubu

Construction Technologies Research Grant Committee



TÜRKİYE BİLİMSEL VE  
TEKNİK ARAŞTIRMA KURUMU

THE SCIENTIFIC AND TECHNICAL  
RESEARCH COUNCIL OF TURKEY

PREFABRİK ÇERÇEVE ve SÜNEK DUVARLARDAN

PREFABRİK ÇERÇEVE ve SÜNEK DUVARLARDAN

OLUŞAN YAPI SİSTEMLERİNİN DEPREM DAVRANIŞININ

ANALİTİK İNCELENMESİ

Proje No: 195 I 001

PROJE NO: İNTAG 542

2001 - 17

1.10.94

30.4.98

(25)

1-56 EKA

Prof. S.M. Çizdemir  
Doç. Dr. Uğur Pala  
Prof. Dr. Güneş Özçelik  
Prof. Dr. Uğur Fırat

ODTÜ İnşaat Müh.  
81, Ank.

İnşaat Teknolojileri Araştırma Grubu

Construction Technologies Research Grant Committee

ANKARA

**PREFABRİK ÇERÇEVE ve SÜNEK DUVARLARDAN  
OLUŞAN YAPI SİSTEMLERİNİN DEPREM DAVRANIŞININ  
ANALİTİK İNCELENMESİ**

**PROJE NO: İNTAG 542**

**Prof. Ş.M.Üzümeri  
Doç.Dr.Uğur Polat  
Prof.Dr.Güney Özcebe  
Prof.Dr.Uğur Ersoy**

**Aralık 1999  
ANKARA**

## ÖNSÖZ

Prefabrik çerçeve ve süknk duvarlardan oluşan yapı sistemlerinin deprem davranışının analitik incelenmesi amacı ile yapılan bu araştırma ODTÜ İnşaat Mühendisliği Bölümü Yapı Mekanığı Laboratuvarında TÜBİTAK'ın desteğı ile gerçekleştirilmiştir.

Bu çalışma Dr.Uğur Polat yönetiminde ODTÜ İnşaat Mühendisliği Bölümü öğrencilerinden A.Hakan Argeşo, Özgür Yüncü, Özgür Kurç ve H.Bahadır Bahat'ın Yüksek Lisans tez çalışmalarını çerçevesinde gerçekleştirilmiştir.

## ÖZ

Uygulamada uzay çerçeve yapısal taşıyıcı sistemlerin analizleri genellikle birleşimleri için yapılan *rijit birleşim* veya *mafsallı birleşim* kabulü ile gerçekleştirilmektedir. Bu proje kapsamında, prefabrik çerçeve sistemler için pek de gerçekçi ve geçerli olmadığı bilinen bu basitleştirmeyi gereksiz kılmak ve bu tür yapısal sistemlerin analizlerini daha gerçekçi olarak gerçekleştirebilmek için gerekli olan, bu tür sistemlerin *yarı rijit* birleşimlerini modellemek üzere bir *Elasto-Plastik Betonarme Arayüz Elemanı* geliştirilmiştir.

Boşluklu ve boşluksuz perde duvar ve döşemeler gibi iki boyutlu elemanlar içeren yapısal sistemlerin sonlu elemanlar yöntemi ile analizlerinde karşılaşılan modelleme, bilgisayar kapasitesi ve yüksek hacimdeki tepki çıktılarının tasarıma temel teşkil edecek tepkiler haline dönüştürülmesi aşamalarında karşılaşılan güçlükleri gidermek üzere bir *Düzlem Panel Makro (Süper) Elemanı* geliştirilmiştir.

Yapısal sistemlerin düzgün olmayan bir geometriye sahip ve/veya düzensiz yüklemeye maruz döşemelerinin analizlerini sonlu elemanlar yöntemi ile gerçekçi olarak gerçekleştirebilmek, sonuçları tasarımda direkt olarak kullanılabilir hale getirmek ve bu yatay elemanlardan çerçeve taşıyıcı sisteme aktarılan yükleri sağlıklı bir şekilde belirleyebilmek üzere bir Sonlu Elemanlar Tabanlı Plak İşlemcisi bilgisayar yazılımı geliştirilmiştir.

## ABSTRACT

In practice, the analysis of space frame structural systems are carried out under the assumption that their joints are either rigid or ideal hinges. However, this is far from reality in the case of structural systems composed of prefabricated elements whose connections are semi-rigid. A special *Elasto-Plastic R/C Interface Element* is formulated and implemented for the modeling of such semi-rigid connections of prefabricated structural systems.

Accurate analysis of structural systems containing planar structural elements such as the structural walls and the slabs with or without openings using finite elements normally requires computers with very high processing speed and capacity. A *Panel Macro (Super) Element* is developed for the analysis of structural systems containing such elements which minimizes the demand on the part of the computers as well as the analyst engineer.

There are some serious difficulties in the finite element modelling, analysis and interpretation and digestion of the high volume of analysis results of structural plates with irregular geometries under arbitrary loading. A general purpose, finite element based, plate processing environment with a graphical user interface was also developed to overcome these difficulties which makes it possible to calculate the loads being transferred to the structural system from floor slabs.

## İÇİNDEKİLER

ÖNSÖZ.....	i
ÖZ .....	ii
ABSTRACT.....	iii
İÇİNDEKİLER.....	iv
TABLULAR LİSTESİ .....	vii
ŞEKİLLER LİSTESİ .....	viii
SEMBOLLER LİSTESİ.....	x
PROJE İLE İLGİLİ AÇIKLAMALAR .....	xi
<b>1 BİRLEŞİMLERİ ELASTO-PLASTİK MAFSALLI BETONARME ÇERÇEVE TAŞIYICI SİSTEMLER.....</b>	<b>1</b>
1.1 Giriş.....	1
1.2 Yarı-Rijit Birleşimler .....	3
1.3 Elasto-Plastik Betonarme Arayüz Elemanı .....	6
1.4 Betonarme Arayüz Eleman Modelinin Kalibrasyonu.....	11
1.4.1 Eksenel Davranış Düzeltme Faktörü.....	12
1.4.2 Beton Kesit Etkin Kesme Alanı Oranı .....	13
1.4.3 Kesit Etkin Etriye Alanı Katsayısı .....	13
1.5 Betonarme Arayüz Eleman Verifikasyon Çalışması .....	17
1.5.1 Deneysel Sonuçlar.....	19
1.5.2 Elastik Çerçeve/Betonarme Arayüz Elemanları Analitik Modeli.....	19
1.5.3 Analitik Model Sonuçları .....	20

<b>2 DÜZLEM PANEL MAKRO ELEMANI.....</b>	<b>27</b>
2.1 Giriş.....	27
2.2 Panel Makro (Süper) Elemanı .....	29
2.2.1 Makro Eleman Kenar Deplasman Fonksiyonları.....	31
2.2.2 Kısıt Denklemlerinin Global Sistemde İfadesi .....	35
2.3 Sayısal Test Problemi.....	37
<b>3 XSPlate: SONLU ELEMANLAR TABANLI BİR PLAK İŞLEMCİSİ .....</b>	<b>42</b>
3.1 Giriş.....	42
3.2 XSPlate Plak Analiz Sisteminin Ana Hatları .....	44
3.3 DKT Plak Eğilme Elemanı.....	45
3.4 Sayısal Test Problemleri.....	46
3.4.1 Duvar Yüküne Maruz, Kenarları Boyunca Basit Mesnetli Dikdörtgen Plak	46
3.4.2 Geometrisi Düzensiz Boşluklu Döşeme.....	50
<b>KAYNAKLAR.....</b>	<b>54</b>
<b>EK A BİRLEŞİMLERİ ELASTO-PLASTİK MAFSALLI ÇERÇEVE SİSTEMLERİN ANALİZİ PROGRAMI – TANIMI VE LİSTESİ.....</b>	<b>57</b>
A.1 NODE Objesi.....	57
A.2 FRAME Objesi .....	60
A.3 ELEMENT Objesi.....	64
A.4 TRUSS Objesi.....	69
A.5 CONNECTION Objesi.....	70
A.6 STRUCTURE Objesi .....	73
A.7 SOLVER Objesi.....	74
A.8 PROBLEM Objesi .....	85
A.9 Veri Dosyası Yapısı .....	89



<b>EK B DENEYSEL ÇERÇEVE ANALİZLERİ VERİ VE ÇIKTI DOSYASI</b>	<b>93</b>
B.1 Veri Dosyası.....	93
B.2 Çıktı Dosyası.....	94



## ŞEKİLLER LİSTESİ

1.1 Tipik bir Moment-Eğrilik İlişkisi .....	2
1.2 Uçları elasto-plastik açısız yaylar içeren çerçeve elemanı.....	3
1.3 Tipik uç mafsallı rijitliği – ankastrelik katsayısı ( $k-\alpha$ ) ilişkisi .....	5
1.4 Elasto-Plastik Betonarme Arayüz Elemanı modeli temel yapısı ve bileşenleri	7
1.5 Betonun basınç davranışı .....	9
1.6 Betonun çekme davranışı .....	9
1.7 Donatı çeliği davranış modeli.....	10
1.8 Uçlarında rijit bölgeler ihtiva eden arayüz elemanı.....	10
1.9 Düz ve çapraz elemanlı arayüz elemanı eksenel davranışı .....	12
1.10 Arayüz elemanı kesme davranışı için kullanılan betonarme kiriş detayları....	14
1.11 Betonarme kirişin betonarme arayüz elemanları ile oluşturulan analitik modeli	15
1.12 Basit mesnetli kiriş için yük-eksenel boy değişimi (kiriş eksenine boyunca) ilişkisi	15
1.13 İki ucu ankastre kiriş için yük-eksenel kuvvet (kiriş eksenine boyunca) ilişkisi	16
1.14 İki ucu ankastre kiriş için yük-deplasman (kiriş orta noktası) ilişkisi .....	16
1.17 Basit mesnetli kiriş için yük-deplasman (kiriş orta noktası) ilişkisi .....	17
1.18 Betonarme arayüz elemanı test çalışmasında kullanılan betonarme çerçeve[7]	18
1.19 Betonarme deneysel test çerçevesi analitik modeli .....	20
1.20 Birinci kat kirişi eksenel uzaması .....	23
1.21 Birinci kat seviyesi yanal deplasmanı.....	23
1.22 İkinci kat seviyesi yanal deplasmanı.....	24
1.23 Sol tepe noktasındaki dönme.....	24
1.24 Birinci kat kirişi üst donatısındaki birim deformasyon (basınç) .....	25
1.25 Birinci kat kirişi alt donatısındaki birim deformasyon (çekme).....	25

1.26 Sol kolon tabanındaki eğrilik.....	26
1.27 Sol kolon tabanındaki aksenal birim deformasyon.....	26
2.1 Tipik bir Panel Makro Elemanı ve topolojik tanımlar.....	30
2.2 Tipik bir kenar boyunca panel membran modu serbestlikleri.....	31
2.3 Tipik bir kenar boyunca panel plak eğilme modu serbestlikleri.....	33
2.4 İki gözlü, kapalı kutu tipi test yapısı: plan görünüşü ve yüklemeler (Q ve T).....	38
2.5 Q yüklemesi altında 1-1 hattı boyunca oluşan yanal deplasmanlar.....	38
2.6 Q yüklemesi altında P3 tabanı boyunca oluşan aksenal kuvvet dağılımı.....	39
2.7 T yüklemesi altında 1-1 hattı boyunca oluşan yanal deplasmanlar.....	39
2.8 T yüklemesi altında P3 tabanı boyunca oluşan aksenal kuvvet dağılımı.....	40
3.1 DKT plak eğilme elemanı ilk ve son topolojisi.....	45
3.2 Kenarları boyunca basit mesnetli ve duvar yüküne maruz dikdörtgen plak.....	47
3.3 X-ekseni boyunca oluşan plak deplasmanları (Y=0.0).....	47
3.4 Y-ekseni boyunca oluşan plak deplasmanları (X=2.40).....	48
3.5 X-ekseni boyunca $M_{xx}$ değişimi (Y=0.0).....	48
3.6 X-ekseni boyunca $M_{yy}$ değişimi (Y=0.0).....	49
3.7 Y-ekseni boyunca $M_{xx}$ değişimi (X=2.40).....	49
3.8 Y-ekseni boyunca $M_{yy}$ değişimi (X=2.40).....	50
3.9 Geometrisi düzensiz boşluklu döşeme.....	51
3.10 X-ekseni boyunca deplasman (Y=3.0m).....	51
3.11 X-ekseni boyunca $M_{xx}$ değişimi (Y=3.0m).....	52
3.12 X-ekseni boyunca $M_{yy}$ değişimi (Y=3.0m).....	52

## SEMBOLLER LİSTESİ

L	Eleman boyu
E	Elastisite modülü
K	Rijitlik matrisi
$u, v, \theta$	x-deplasmanı, y-deplasmanı ve z-dönmesi serbestlikleri
$F_i$	$i$ serbestliği yönündeki kuvvet
$\alpha_1, \alpha_2$	Açısal yaylar 1 ve 2 için yay katsayıları
f	Yük vektörü
a	Deplasman vektörü
$\Delta f, \Delta a$	Yük ve deplasman değişimi
r	Residual force vector
$K_t$	Tanjant rijitlik matrisi
$r_1, r_2$	Rijit bölge boyutları
T	Rijit bölgeler için transformasyon matrisi
$\alpha_a$	Eksenel davranış düzeltme faktörü
$\alpha_c$	Beton kesit etkin kesme alanı oranı
$\alpha_s$	Kesit etkin etriye alanı katsayısı
$A'$	Modifiye edilmiş beton alanı
A	Toplam kesit alanı
$A_{eff}$	Etkin çapraz alanı
$A_c$	Beton çapraz alanı
$A_s$	Çelik çapraz alanı
$A_{st}$	Gövde donatısı alanı
h	Kesit yüksekliği
s	Gövde donatısı aralığı

## PROJE İLE İLGİLİ AÇIKLAMALAR

### Amaç

Bu çalışmanın amacı Türkiye’de imal edilen prefabrik çerçevelerin sünek duvarlar ile birlikte kullanıldıkları yapılarda, yapının deprem davranışını inceleyerek, nasıl bir sünek duvar/prefabrik çerçeve karışımının en iyi sonuçları verdiğini belirlemek ve böylece Türk Mühendislerine, prefabrik üreticilerine ve topluma hizmet etmektir.

Prefabrik elemanlardan yapılan çerçevelerin en *zayıf* noktaları birleşim (düğüm) noktalarıdır. Düğüm noktaları için, hemen her üretici, kendi ürettiği ve ekonomik bulduğu detayları kullanır. Bu sebeple prefabrik çerçevelerin davranışları, prefabrik eleman üreticisinin kullandığı düğüm noktalarının tasarım ve imalat detaylarına ve kalitesine dayanır. Düğüm noktalarının davranışları, adeta üreticiye özeldir ve üreticiden üreticiye değişir.

Prefabrike çerçevelerin elemanlarının düğüm noktalarının detaylarının ne şekilde oluşturulmaları gerektiği hakkında ülke ve hatta dünya çapında benimsenmiş kurallar yoktur. Bazı ülkeler, prefabrik elemanlardan yapılmış yapıların deprem bölgelerinde kullanımlarını yönetmeliklerine bile dahil etmemektedirler.

Türkiye, deprem kuşağında olduğu halde, ekonomik bakımdan, prefabrike yapıların kullanımını yönetmelik dışı bırakabilecek durumda değildir. Bu sebeplerle, Türkiye’de kullanılan prefabrike elemanlardan oluşturulan prefabrik yapıların, deprem karşısındaki davranışlarının incelenmesi gerektiği düşüncesindeyiz.

### Yöntem

Literatürde, daha evvel yapılmış, prefabrike yapılar, perde duvarları gibi konuları içeren bir çok çalışma bulunmaktadır. Fakat bu çalışmaların hiç birisi, konuya arzu edilen şekilde yaklaşmamış ve bilhassa hiç birisi, Türk prefabrik eleman üreticilerinin elemanlarını, araştırmanın konusu olarak almamıştı.

İnceleme yöntemi, Türkiye’de kullanılan prefabrike çerçeve elemanları ve bunların düğüm noktaları hakkında bilgi toplamak, bunların davranışlarının matematiksel modellerini çıkarmak ve bu matematiksel modellere, sünek duvarları da ekleyerek, prefabrik çerçeve ve sünek duvarlardan yapılmış yapı sistemlerinin depreme karşı davranışlarını incelemektir.

Bu çalışmanın önemli bir karakteri elde edilecek analitik çözüm tekniğinin endüstrideki deneyimle kalibre edilmesidir. Bu amaçla endüstri ile ilişkileri geliştirmek için Prefabrik Birliği’nden listesini aldığımız 23 firmaya bir yazı gönderilmişti. Yazı yollanan firmaların isim ve adresleri ve iletilen yazının bir kopyası 2. Gelişme Raporumuza (15 Kasım 1996) eklenmişti. Maalesef, prefabrik eleman üreten firmalardan umulan yanıtı alamadık.

Bu durum karşısında, Prefabrik Birliği üyeleri ile yapılan temaslardan, sektörün genelde o günün koşullarına göre yönlendirildiği anlaşılmaktadır. Bundan on yıl kadar önce konut ve öğrenci yurdu gibi binalara duyulan acil gereksinme dolayısı ile çok katlı prefabrike binalar gündeme gelmiş ve çok sayıda yapılmıştır. Ancak daha sonra konut sektörüne aktarılan

kaynakların azalması, yurt yapımının güncelliğini kaybetmesi nedeniyle prefabrik sektörü tamamen tek katlı yapılara dönüşmüştür. Çok katlı prefabrik yapılar şu anda gündemde olmadığından, sektör günlük işleri içinde bu konuya fazla ilgi duymamaktadır. Ancak bazı firmalar ve üniversite, bu tür bir araştırmanın bilgi birikimi açısından çok önemli olduğu kanısındadır. Ülkenin konut politikasında yapılacak bir değişiklikle çok katlı prefabrike yapılar yeniden güncellik kazanabilecektir.

Çalışma için büyük önem taşıyan ve prefabrik üreticileri ile yapılacak işbirliği yolu ile elde edilmesi umulan verilerin elde edilememesi, araştırmanın yönünün değiştirilmesini gerektirdi. Matematik davranış modellerinin geliştirilmesine devam edilirken, model kalibrasyonu, literatürde yer alan iyi dokümente edilmiş deneysel yayınlar ile yapılmaya çalışıldı.

Bu sebeple, başlangıçta, bilhassa Prefabrik üreticilerince yapılmış olan deneysel araştırmaların sonuçlarını inceleyip, bunların matematiksel modelinin belirlenmesi ve kalibrasyonunda kullanılması düşünüldüğünden, araştırmaya yardım için, bu alanlarda yeteneği ve tecrübesi olan Doktora öğrencisi Erdem Canbay araştırma görevlisi olarak seçilmiş ve burs ona verilmiştir.

Daha sonra, araştırmanın tamamen analitik bir çalışma sekline bürünmesi dolayısı ile, Projenin bütün çalışmaları Doç. Dr. Uğur Polat tarafından üstlenmiş olup, araştırma görevlisi olarak da, Erdem Canbay'ın yerine 1.5.1997 den itibaren M.S. öğrencisi Hakan Bahadır Bahat atanmıştır.

Doç.Dr. Uğur Polat tarafından yapılan çalışmalar üç bölüm ve iki ekten oluşmakta olup bu bölümler ve kısa özetleri aşağıda verilmektedir.

### **Bölüm 1 – Birleşimleri Elasto-Plastik Mafsallı Betonarme Çerçeve Taşıyıcı Sistemler.**

Uygulamada uzay çerçeve yapısal taşıyıcı sistemlerin analizleri genellikle birleşimleri için yapılan *rijit birleşim* veya *mafsallı birleşim* kabulü ile gerçekleştirilmektedir. Bu proje kapsamında, prefabrik çerçeve sistemler için pek de gerçekçi ve geçerli olmadığı bilinen bu basitleştirmeyi gereksiz kılmak ve bu tür yapısal sistemlerin analizlerini daha gerçekçi olarak gerçekleştirebilmek için gerekli olan, bu tür sistemlerin *yarı rijit* birleşimlerini modellemek üzere bir *Elasto-Plastik Betonarme Arayüz Elemanı* geliştirilmiştir.

### **Bölüm 2 – Düzlem Panel Makro Elemanı**

Boşluklu ve boşluksuz perde duvar ve döşemeler gibi iki boyutlu elemanlar içeren yapısal sistemlerin sonlu elemanlar yöntemi ile analizlerinde karşılaşılan modelleme, bilgisayar kapasitesi ve yüksek hacimdeki tepki çıktılarının tasarıma temel teşkil edecek tepkiler haline dönüştürülmesi aşamalarında karşılaşılan güçlükleri gidermek üzere bir *Düzlem Panel Makro (Süper) Elemanı* geliştirilmiştir.

### **Bölüm 3 – XSPlate: Sonlu Elemanlar Tabanlı bir Plak İşlemcisi**

Yapısal sistemlerin düzgün olmayan bir geometriye sahip ve/veya düzensiz yüklemeye maruz döşemelerinin analizlerini sonlu elemanlar yöntemi ile gerçekçi olarak gerçekleştirebilmek,

sonuçları tasarımda direkt olarak kullanılabilir hale getirmek ve bu yatay elemanlardan çerçeve taşıyıcı sisteme aktarılan yükleri sağlıklı bir şekilde belirleyebilmek üzere bir Sonlu Elemanlar Tabanlı Plak İşlemcisi bilgisayar yazılımı geliştirilmiştir.

#### **Ek A – Birleşimleri Elasto-Plastik Mafsallı Çerçeve Sistemlerin Analizi Programı – Tanımı ve Listesi**

#### **Ek B – Deneysel Çerçeve Sistemlerin Analizleri Veri ve Çıktı Dosyaları**

#### **Bütçe**

Proje bütçesinin üç ana kalemi vardı. Bunlardan ilki, analitik çalışmalarda kullanılmak üzere güçlü bir bilgisayarın alınması; ikincisi bir analiz programı alınması ve üçüncüsü ise bir öğrenci asistana verilmek üzere bir burs fonu idi. Ayrıca sarf malzemeleri için de küçük bir kalembulunmaktaydı. Projenin gelişim sürecinde sarf malzemesi için ön görülen miktar TÜBİTAK onayı ile bilgisayarın belleğini genişletmeye harcanmıştır. Böylece, genelde harcamalar tümü ile bütçe kalemlerine uygun olarak yapılmıştır.



## BÖLÜM 1

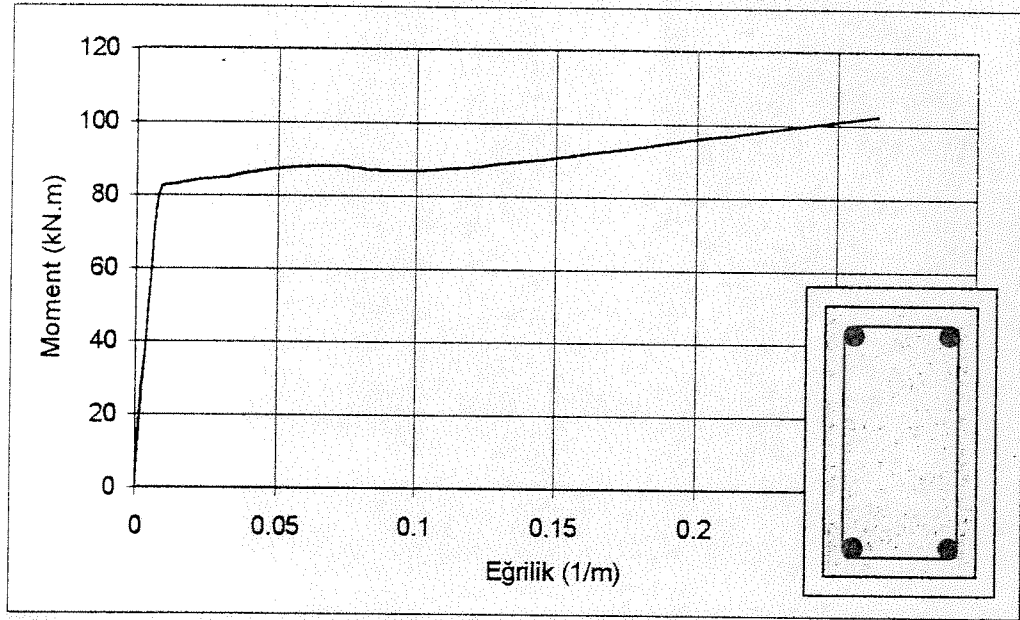
# BİRLEŞİMLERİ ELASTO-PLASTİK MAFSALLI BETONARME ÇERÇEVE TAŞIYICI SİSTEMLER

### 1.1 Giriş

Yapısal sistemlerin analizlerini analitik yöntemlerle gerçekleştirmek üzere oluşturulacak matematiksel modellerin oluşturulması aşamasında değişik düzeylerde de olsa birtakım basitleştirme ve idealleştirmelerin uygulanması kaçınılmazdır. Bu yaklaşımların bazıları, doğrusal elastik davranış varsayımında olduğu gibi, problemin çözümünü olası kılmak üzere çok temel düzeyde olmakla birlikte birçoğu da hesaplama kolaylığına yöneliktir. İkinci grup varsayımlara uzay çerçeve yapısal taşıyıcı sistemler özelinde önemli bir örnek olarak bu tür sistemlerin birleşimleri için yapılan *rijit birleşim* veya *mafsallı birleşim* kabulü gösterilebilir. Ancak gerçekte bu tür yapısal sistem birleşimlerinin her iki ideal durumun da dışında davrandığı bilinmektedir. Gerçekte bu iki ideal durum birleşimlerin davranış spektrumundaki iki uç noktayı tanımlamaktadır. Çelik yapılar için yüzyılın başlarında anlaşılan bu davranışı anlamaya ve *yarı rijit birleşim* kavramını analizlerde kullanmaya yönelik ciddi araştırma faaliyetleri 1960 lı yıllarda başlamıştır.

Betonarme çerçeve sistemler için kullanılan bu geleneksel yaklaşım yapıların analiz ve tasarım aşamalarında büyük kolaylıklar sağlamakla birlikte birçok durumda hesaplanan yapısal tepki pek gerçekçi olamamaktadır. Bu konudaki çok

çarpıcı bir durum prefabrik yapısal sistemlerde karşımıza çıkmaktadır. Yapılan deneysel çalışmalarda prefabrik elemanların birleşimlerinin relatif dönmelere karşı rijit olmasına yönelik her türlü gayrete karşın pratikte bir miktar moment aktarırken bir miktar da relatif dönme yaptığı ve sonuçta bu dönmelerin yapısal deplasmanlara ciddi düzeyde katkıda bulunduğu görülmektedir. İki prefabrik elemanın birleştirilmesi ile oluşan sistemler eğilme altında test edildiğinde oluşan dönmeler prefabrik elemanların eğilmesi yanında birleşimde oluşan relatif dönmeleri de içermektedir. Birleşimde oluşan rijit cisim dönmeleri ölçülen toplam dönmelerden çıkartılarak uygulanan moment birleşimdeki karakteristik dönmeye (eğrilik) karşı çizilecek olursa Şekil 1.1 de görülen tipik bir Moment-Eğrilik ( $M/\phi$ ) ilişkisi elde edilir.



Şekil 1.1 Tipik bir Moment-Eğrilik ilişkisi

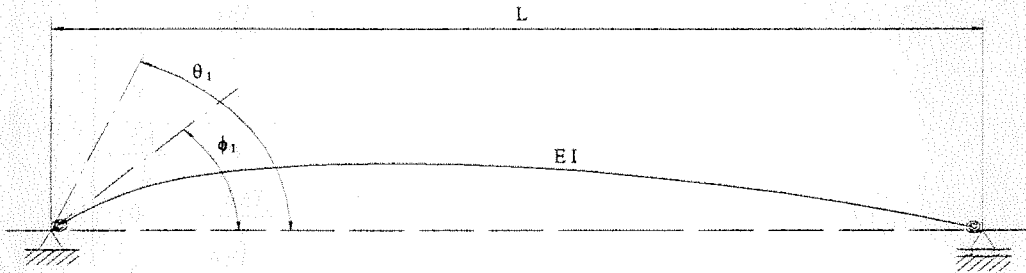
Görüldüğü gibi davranış doğrusal olmayıp belli bir düzeye kadar gerçekleşen rijit davranışı artan dönme zorlamaları altında çok daha yumuşak bir davranış izlemektedir.

Yarı rijit esnek birleşim davranışı yapısal deplasmanları olduğu gibi hiperstatik sistemlerde yapısal elemanlarda oluşan zorlamaları da etkilemektedir. Bu

araştırma projesinin ilk aşamasında birleşimleri Şekil 1.1 de görüldüğü gibi Elasto-Plastik davranış gösteren prefabrik yapısal sistemlerin analizlerini daha gerçekçi olarak gerçekleştirmek üzere geliştirilecek bir uzay çerçeve yapısal analiz yazılımının oluşturulması ve özellikle de böyle bir sistemin temel yapı taşı olan birleşimlerdeki elasto-plastik davranışı modelleyebilecek bir arayüz elemanı üzerinde yoğunlaşmıştır.

## 1.2 Yarı-Rijit Birleşimler

Birleşimleri elasto-plastik mafsallı çerçeve sistemleri oluşturan elemanlar en basit şekli ile Şekil 1.2 de görüldüğü gibi uçlarına rijitlikleri  $k_1$  ve  $k_2$  olan elasto-plastik açısallı yaylar bağlanmış süper (makro) elemanlar olarak düşünülebilir.



Şekil 1.2 Uçları elsto-plastik açısallı yaylar içeren çerçeve elemanı

Böyle bir elemanın rijitlik matrisi ise standart elemanın rijitlik matrisi modifiye edilerek rahatlıkla oluşturulabilir. Örneğin böyle bir düzlem çerçeve makro elemanın rijitlik matrisi (1.1) de verilen standart eleman rijitlik matrisi

$$\begin{bmatrix} \frac{EA_x}{L} & 0 & 0 & -\frac{EA_x}{L} & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & \frac{6EI_z}{L^2} \\ 0 & \frac{6EI_z}{L^2} & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & \frac{2EI_z}{L} \\ \hline -\frac{EA_x}{L} & 0 & 0 & \frac{EA_x}{L} & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & -\frac{6EI_z}{L^2} \\ 0 & \frac{6EI_z}{L^2} & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & \frac{4EI_z}{L} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} \quad (1.1)$$

uçlarında bulunan yayların katkısı ile modifiye edilerek (1.2) de verildiği şekilde elde edilir.

$$\begin{bmatrix} \frac{EA_x}{L} & 0 & 0 & 0 & -\frac{EA_x}{L} & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & \frac{6EI_z}{L^2} & 0 & 0 & -\frac{12EI_z}{L^3} & \frac{6EI_z}{L^2} & 0 \\ 0 & \frac{6EI_z}{L^2} & \frac{4EI_z}{L} + k_1 & -k_1 & 0 & -\frac{6EI_z}{L^2} & \frac{2EI_z}{L} & 0 \\ 0 & 0 & -k_1 & k_1 & 0 & 0 & 0 & 0 \\ \hline -\frac{EA_x}{L} & 0 & 0 & 0 & \frac{EA_x}{L} & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & -\frac{6EI_z}{L^2} & 0 & 0 & \frac{12EI_z}{L^3} & -\frac{6EI_z}{L^2} & 0 \\ 0 & \frac{6EI_z}{L^2} & \frac{2EI_z}{L} & 0 & 0 & -\frac{6EI_z}{L^2} & \frac{4EI_z}{L} + k_2 & -k_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \\ \phi_1 \\ u_2 \\ v_2 \\ \theta_2 \\ \phi_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ 0 \\ F_4 \\ F_5 \\ F_6 \\ 0 \end{Bmatrix} \quad (1.2)$$

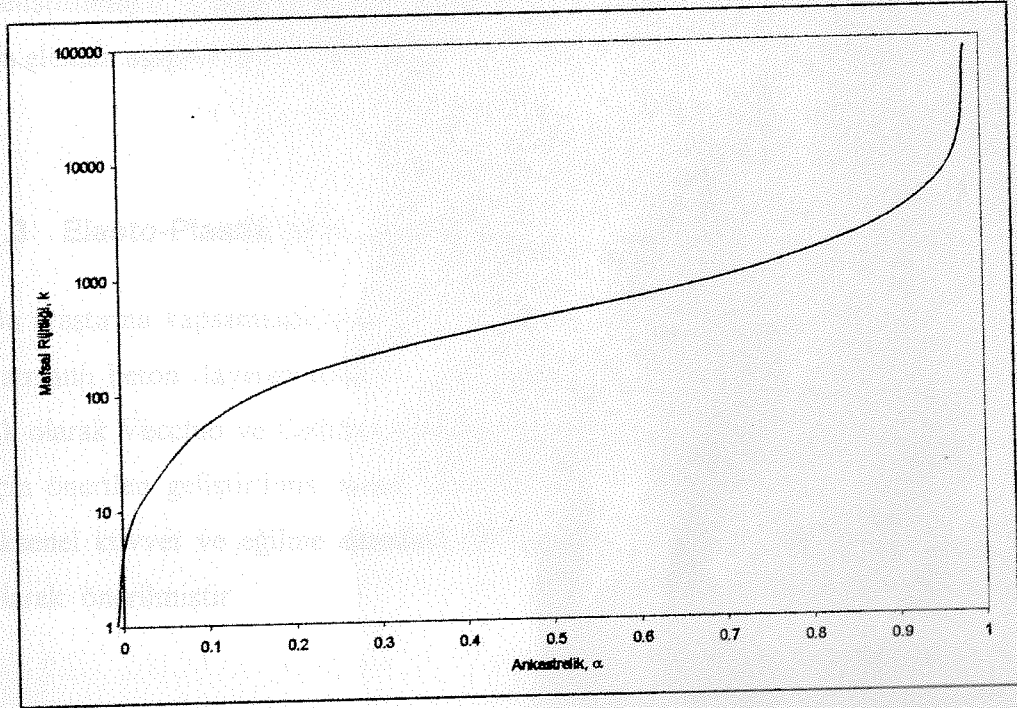
Bu düzlem çerçeve makro eleman rijitlik matrisi eleman tali serbestlikleri  $\phi_1$  ve  $\phi_2$  elimine edilerek Monforton ve Wu [1] tarafından

$$\begin{bmatrix} \frac{EA_x}{L} & 0 & 0 & -\frac{EA_x}{L} & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} (\alpha_1 + \alpha_2 + \alpha_1 \alpha_2) & \frac{6EI_z}{L^2} (2\alpha_1 + \alpha_2 \alpha_2) & 0 & -\frac{12EI_z}{L^3} (\alpha_1 + \alpha_2 + \alpha_1 \alpha_2) & \frac{6EI_z}{L^2} (2\alpha_1 + \alpha_2 \alpha_2) \\ 0 & \frac{6EI_z}{L^2} (2\alpha_1 + \alpha_2 \alpha_2) & \frac{4EI_z}{L} (3\alpha_1) & 0 & -\frac{6EI_z}{L^2} (2\alpha_1 + \alpha_2 \alpha_2) & \frac{2EI_z}{L} (3\alpha_1) \\ \hline -\frac{EA_x}{L} & 0 & 0 & \frac{EA_x}{L} & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} (\alpha_1 + \alpha_2 + \alpha_1 \alpha_2) & -\frac{6EI_z}{L^2} (2\alpha_1 + \alpha_2 \alpha_2) & 0 & \frac{12EI_z}{L^3} (\alpha_1 + \alpha_2 + \alpha_1 \alpha_2) & -\frac{6EI_z}{L^2} (2\alpha_1 + \alpha_2 \alpha_2) \\ 0 & \frac{6EI_z}{L^2} (2\alpha_2 + \alpha_1 \alpha_2) & \frac{2EI_z}{L} (3\alpha_2) & 0 & -\frac{6EI_z}{L^2} (2\alpha_2 + \alpha_1 \alpha_2) & \frac{4EI_z}{L} (3\alpha_2) \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} \quad (1.3)$$

olarak verilmektedir. Burada parametreler  $\alpha_1$  and  $\alpha_2$  ankastrelik faktörleri olarak tanımlanmakta ve açılmal yay rijitlikleri  $k_1$  ve  $k_2$  cinsinden

$$\alpha = \frac{\phi}{\theta} ; \alpha_1 = \frac{1}{1+3EI/k_1L} ; \alpha_2 = \frac{1}{1+3EI/k_2L} \quad (1.4)$$

olarak hesaplanmaktadır. Ankastrelik faktörü  $\alpha$  birleşimin bağlandığı elemana göre görece rijitliğini ifade etmekte olup elemanın yapı içindeki davranışını uç mafsallı mutlak rijitliğine oranla çok daha anlamlı olarak ifade etmektedir. Uç mafsallı rijitliği  $k$  ile ankastrelik katsayısı  $\alpha$  arasındaki tipik ilişki aşağıda Şekil 1.3 de görüldüğü gibidir.



Şekil 1.3 Tipik uç mafsallı rijitliği - ankastrelik katsayısı ( $k-\alpha$ ) ilişkisi

Şekilde de görüldüğü gibi tam rijitlikten ( $\alpha=1.0$ ) oldukça büyük sapmalar ankastrelik katsayısı  $\alpha$  da oldukça küçük değişim yaratmakta ve dolayısı ile yapının genel davranışını önemli ölçüde etkilememektedir. Bu yüzden elasto-plastik mafsallı için sadece yaklaşık bir hassasiyetteki  $M/\phi$  ilişkisi yeterli bir hassasiyet düzeyindeki yapısal analiz için yeterli olacaktır.

Yukarıda özetlenen yaklaşım genel olarak geçerli olmakla birlikte betonarme yapılar bazında direkt olarak kullanımı pek kolay değildir. Betonun izotropik bir malzeme olmaması nedeni ile yarı-rijit bir betonarme birleşimin davranışı doğrusal-elastik olmayan bir açısal yay ile modellenemeyecek kadar karmaşıktır. Literatürde betonarme elemanlar için önerilen sonlu eleman modelleri ise genellikle sürekli ortam elemanları kullanılarak oluşturulan modeller olup yüzlerce ve hatta binlerce birleşim içeren gerçek bir yapıda kullanılması pratik açıdan pek mümkün alamamaktadır. Ayrıca bu tür modellerin ürettiği sonuçlar günlük hayatta yapısal tasarımda hemen kullanılabilir formatta değildir. Bu nedenle bu araştırma kapsamında aşağıda özetlenen Betonarme Arayüz Elemanı geliştirilmiştir. Özgür Yüncü'nün [2] lisansüstü tez çalışması olarak geliştirilen bu eleman aşağıda kısaca özetlenmektedir

### 1.3 Elasto-Plastik Betonarme Arayüz Elemanı

Bu araştırma kapsamında geliştirilen Elasto-Plastik Betonarme Arayüz Elemanı katmanlı beton (layered concrete) modelinden hareket etmektedir. Bu yaklaşım ilk olarak Vecchio ve Collins [3] tarafından daha önce beton malzeme davranışı için önerilen geliştirilmiş basınç alanı teorisi kullanılarak betonarme kesitlerin aksel kuvvet ve eğilme altındaki davranışının simülasyonu amacıyla yönelik olarak önerilmiştir.

Arayüz elemanının temel fonksiyonu betonarme düzlem çerçeve elemanının davranışını karmaşık gerilme analizlerine gereksinim duymaksızın gerçekleştirmektir. Bu amaçla geliştirilen modelde bir betonarme kesitin mekanik davranışı sadece çubuk elemanlar kullanılarak üretilmeye çalışılmıştır. Çubuk elemanlar tek aksel gerilme ve deformasyon altında bulduklarından malzeme modeli olarak gerek çelik gerekse de beton için oldukça iyi bilinen tek aksel gerilme deformasyon ilişkileri kullanılabilir. Böylece beton malzeme için çok büyük güçlük ve belirsizlikler içeren çok aksel gerilme

altındaki malzeme davranışını gerilme ve deformasyon tensörleri cinsinden ifade edilen bünye denklemlerinin kullanılması zorunluluğu ortadan kalkacaktır.

Elasto-Plastik Betonarme Arayüz Elemanı modeli Şekil 1.4 te görülmektedir. Betonarme çerçeve elemanından alınan bir dilim şeklinde düşünülebilecek olan ve temel olarak iki rijit yüzey arasında bir grup çubuk elemandan oluşan arayüz elemanının formülasyonu aşamasında başlıca şu varsayımlar yapılmaktadır:

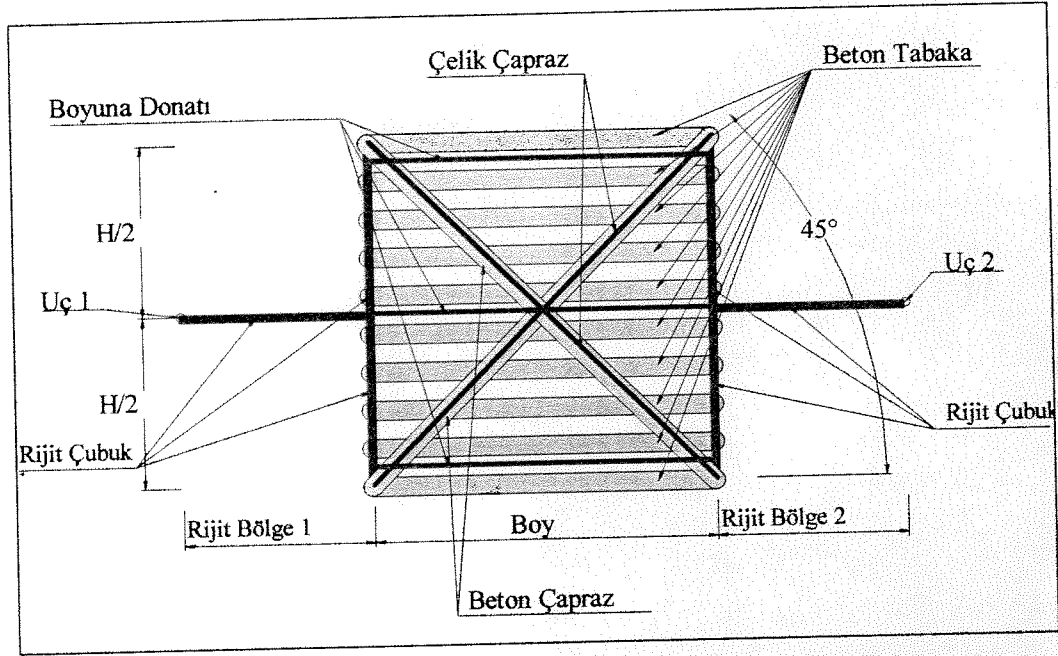


Figure 1.4 Elasto-Plastik Betonarme Arayüz Elemanı modeli temel yapısı ve bileşenleri

- Deformasyon öncesi düzlem olan kesitler deformasyon sonrası da düzlem olarak kalırlar.
- Deplasmanlar ve deformasyonlar küçük olup kesme deformasyonları ihmal edilebilir düzeydedir.
- Beton malzeme özellikleri tüm yapı içinde sabittir.
- Beton ile donatı arasında mükemmel ve sınırsız bir kaynaşma vardır.

- Beton tabakaları birbirinden bağımsız olup aralarında herhangi bir etkileşim söz konusu değildir.

Elasto-Plastik Betonarme Arayüz Elemanı modeli başlıca şu bileşenlerden oluşmaktadır:

- Gövde betonunu modellemek üzere elemanın her iki yüzündeki rijit düzlemler arasında düzlemlere dik olarak uzanan beton çubuk elemanları,
- Boyuna donatıyı modellemek üzere elemanın her iki yüzündeki rijit düzlemlere dik olarak uzanan çelik çubuk elemanları,
- Beton kesme davranışını modellemek üzere rijit düzlemler arasında ve arayüz elemanı derinliğinde X şeklinde yerleşmiş beton çubuk elemanları,
- Birleşim bölgesi içinde yer alan olası etriyeleri modellemek üzere rijit düzlemler arasında ve arayüz elemanı derinliğinde X şeklinde yerleşmiş çelik çubuk elemanları,
- Arayüz elemanının stabilitesini sağlamak, çerçeve elemanı uç bölgelerindeki rijit bölgeleri modellemek ve eğilme davranışının temel varsayımı olan Bernaulli varsayımını empoze etmek üzere rijit çubuk elemanlar.

Elasto-Plastik Betonarme Arayüz Elemanı modeli bu yapısı ile gerçekte bir betonarme kiriş sonlu elemanı yapısındadır. Bu modelde yatay çubuklar kesitin aksenal yük taşıma kapasitesini belirlerken diyagonal elemanlar kesitteki kesme kuvveti transferini ve kesme kuvveti taşıma kapasitesini sağlamaktadır. Herhangi bir zorunluluk olmamasına karşın bu çalışmada diyagonaller  $45^\circ$  eğimli olarak seçilmiştir. Bu değer betonarme kirişler için Ritter[4] ve Mörsch[5] tarafından önerilen kafes analogisi çalışmasından alınmıştır.



Betonun basınç altındaki davranışı için Şekil 1.5 de görülen Hognestad Modeli (Ersoy [6]) kullanılmıştır.

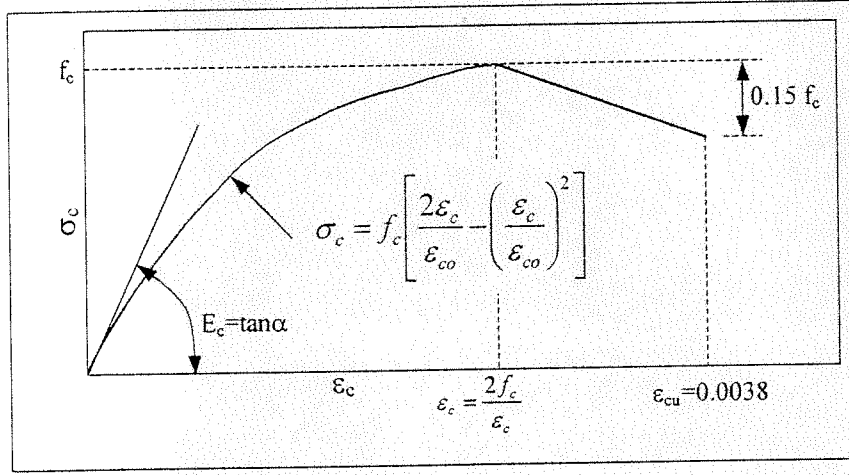


Figure 1.5 Betonun basınç davranışı.

Betonun çekme davranışı ise Şekil 1.6 da görüldüğü gibi alınmıştır.

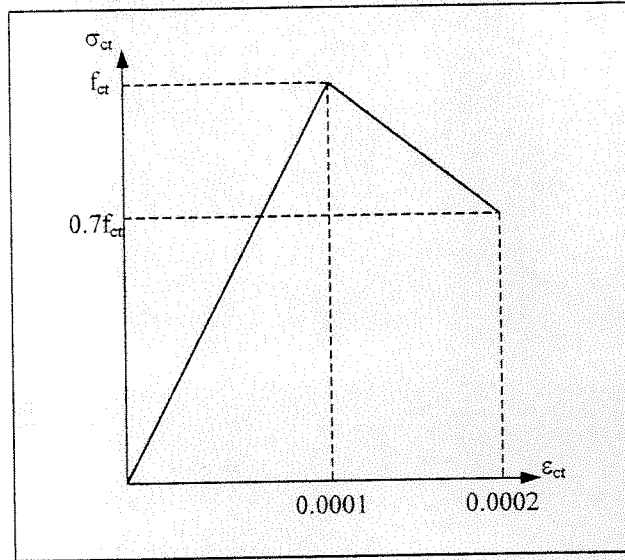


Figure 1.6 Betonun çekme davranışı

Donatı çubukları için ise üç doğrusal parçadan oluşan Şekil 1.7 daki model kullanılmış olup çelik davranışı basınç ve çekme altında benzer kabul edilmiştir.

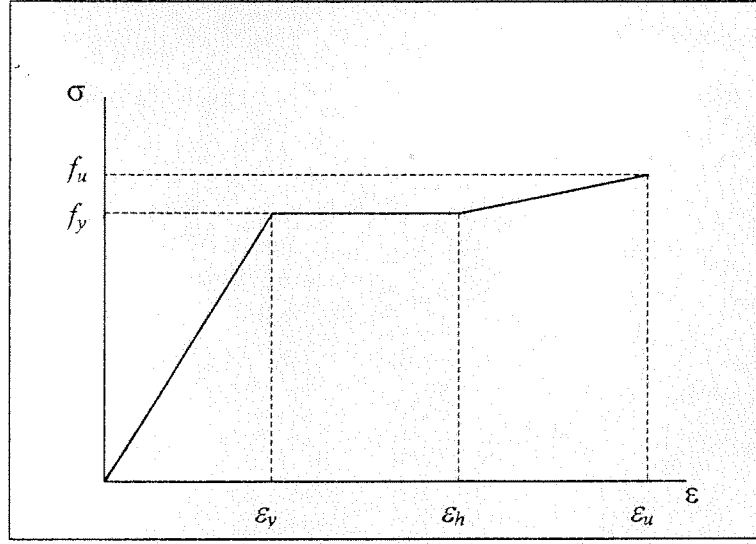


Figure 1.7 Donatı çeliği davranış modeli

Arayüz elemanları yerinde döküm klasik bir betonarme çerçeve sistemin inelastik sismik analizleri için eleman uç bölgelerinde oluşacak plastik mafsallaşmayı da modellemek üzere kullanılabilir. Bu durumda arayüz elemanları kolon ve kirişlerin uç bölgelerinde yer alacak olup her iki yüzünde de olası rijit bölgeler düşünüldüğünde Şekil 1.8 de görülen en genel şekli elde edilir.

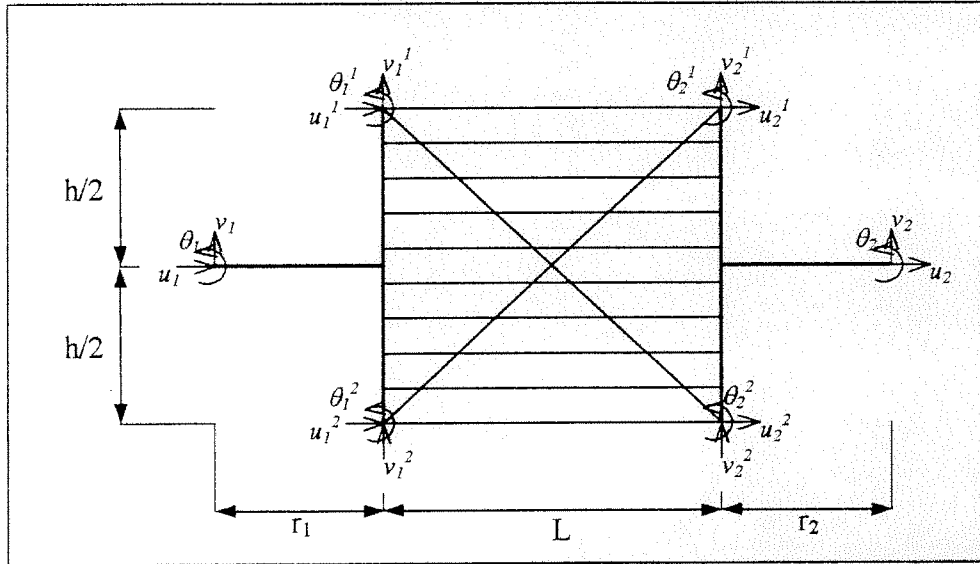


Figure 1.8 Uçlarında rijit bölgeler ihtiva eden arayüz elemanı

Bu durumda I ve J düğüm noktaları arasında tanımlı çubuk uç deplasmanları

$$\begin{Bmatrix} u_1^I \\ v_1^I \\ \theta_1^I \end{Bmatrix} = \mathbf{T}_1 \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -h_I \\ 0 & 1 & r_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \end{Bmatrix} ; \quad \begin{Bmatrix} u_2^J \\ v_2^J \\ \theta_2^J \end{Bmatrix} = \mathbf{T}_2 \begin{Bmatrix} u_2 \\ v_2 \\ \theta_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -h_J \\ 0 & 1 & -r_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_2 \\ v_2 \\ \theta_2 \end{Bmatrix} \quad (1.5)$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}$$

olarak ifade edilebilir. Arayüz elemanı rijitlik matrisi ise eleman rijitlik matrisleri önce her çubuk için (1.5) de tanımlanan dönüşüm matrisleri,  $\mathbf{T}$ , kullanılarak arayüz elemanı serbestlikleri uzayına dönüştürülüp daha sonra da süperpoze edilerek elde edilir.

#### 1.4 Betonarme Arayüz Eleman Modelinin Kalibrasyonu

Betonarme arayüz elemanı matematiksel modeli yukarıda özetlenen ve son derecede basite indirgenmiş yapısından dolayı her ne kadar amaçlanan davranışı büyük ölçüde benzeştirebilecek kapasitede olsa da tanımından kaynaklanan başlıca problemleri gidermek üzere kalibre edilmesi gereği bulunmaktadır. Modelde kalibrasyon gerektiren başlıca üç unsur bulunmaktadır. Bunlar

- Betonun kesme davranışını modellemek üzere kullanılan beton çapraz elemanlar ile aksel davranışını modelleyecek yatay beton çubukları arasındaki etkileşimden kaynaklanan *eksel davranış düzeltme faktörü*,  $\alpha_a$
- Beton çapraz elemanların kesit alanının betonarme yapısal eleman kesit alanı cinsinden ifade edecek *beton kesit etkin kesme alanı oranı*,  $\alpha_c$  ve
- Yapısal elemanın gövde donatısı (etriyeler) ile arayüz elemanı içine X şeklinde yerleştirilen çapraz çelik çubukların kesit alanı arasındaki ilişkiyi belirleyen *kesit etkin etriye alanı katsayısı*,  $\alpha_s$

Modelin kalibrasyonu aşamasında bu katsayıların belirlenmesi gerekmektedir ve bu işlem mevcut deneysel test sonuçları kullanılarak ve sistematik bir deneme yanılma yöntemi ile gerçekleştirilecektir.

#### 1.4.1 Eksenel Davranış Düzeltme Faktörü

Katmanlı beton modelinde betonarme yapısal eleman kasiti genişliği kesit genişliğine eşit birçok katmana bölünür. Bu modelde doğal olarak kesitin kesme dayanımı yok sayılacaktır. Bu eksikliğı gidermek üzere kullanılan X şeklindeki beton çapraz elemanlar ise eksenel zorlamalar altında da etkin olacağından kesitin eksenel dayanımını kaçınılmaz olarak artıracaktır.

Bu durumu gidermek üzere yatay beton katmanlarının kalınlıkları azaltılacaktır. Bu azaltma toplam kesit alanını çapraz elemanların enkesit alanı  $A_c$  nin belli bir oranı  $\alpha_a A_c$  kadar olacaktır. Bu oran X şeklindeki çaprazlar için teorik olarak  $2\cos^2\theta$  olup  $\theta=45^\circ$  için 1.0 dir.

Bu değeri kontrol etmek üzere değişik kesitli iki ayrı donatısız beton arayüz elemanı önce çapraz elemanlar olmadan,  $A_c = 0$ , daha sonra da değişik kesit alanlı çapraz çubuklar kullanılarak aynı davranışı elde etmek üzere test edilmişlerdir. Sonuçlar Şekil 1.9 da görülmektedir.

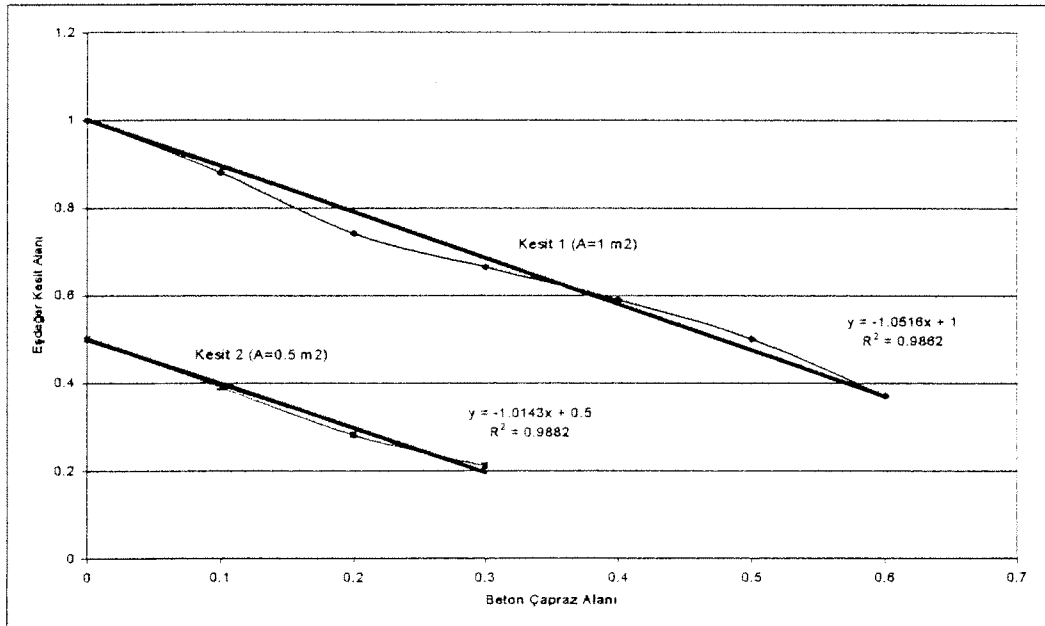


Figure 1.9 Düz ve çapraz elemanlı arayüz elemanı eksenel davranışı

Görüldüğü gibi arayüz elemanı eşdeğer kesit alanı çapraz eleman kesit alanı ile doğrusal olarak değişmekte ve bu değişim beklendiği gibi  $\alpha_a=1.0$  değerine karşı gelmektedir.

#### 1.4.2 Beton Kesit Etkin Kesme Alanı Oranı

Beton çapraz elemanların enkesit alanı yapısal eleman enkesit alanına bağlı olarak

$$A_c = \alpha_c A \quad (1.6)$$

şeklinde ifade edilebilir. Burada  $A_c$  beton çapraz eleman kesit alanı,  $A$  ise modellenen yapısal elemanın brüt enkesit alanıdır.

Kesit alanları eşit ancak beton çapraz enkesit alanı değişen arayüz elemanları ile gerçekleştirilen bir dizi analitik testler ve bu test sonuçlarının deneysel veriler ile kalibrasyonu sonucunda  $\alpha_c$  değerinin 0.10-0.20 aralığında olduğu saptanmıştır. Bu incelemeler sırasında bu değer beton gövde donatısını temsil eden çelik çapraz elemanların kesit alanına son derece hassas olduğu da görülmüştür. Bu nedenle bu parametre kesit etkin etriye alanı ile birlikte daha ayrıntılı olarak incelenecektir.

#### 1.4.3 Kesit Etkin Etriye Alanı Katsayısı

Modelde kullanılan çelik çapraz çubuklar betonarme elemanın arayüz elemanı ile modellenen bölgesi içindeki etriyelerin eleman davranışına katkısını temsil etmektedir. Bu elemanlar aynı zamanda arayüz elemanının aşırı deformasyon altında, gerçek hayatta da olduğu gibi, iç stabilitesini sağlamaktadır.

Toplam kesit alanı  $A_{st}$  olan ve  $h$  derinliğindeki betonarme eleman gövdesine  $s$  aralıklarla yerleştirilmiş etriyelere eşdeğer çelik çapraz çubuk alanı  $A_s$

$$A_s = \alpha_s \frac{A_{st} h}{s} \quad (1.7)$$

olarak tanımlanabilir.

Betonarme kesitin kesme davranışını modelleyecek beton kesit etkin kesme alanı oranı  $\alpha_c$  ve kesit etkin etriye alanı katsayısı  $\alpha_s$  parametrelerini birlikte saptamak üzere bir parametrik çalışma gerçekleştirilmiştir. Bu çalışmada Vecchio ve Emara [7] tarafından gerçekleştirilen deneysel çalışmadan yararlanılmıştır. Söz konusu deneysel çalışmada kullanılan betonarme kiriş detayı aşağıda Şekil 1.10 da gösterildiği gibidir.

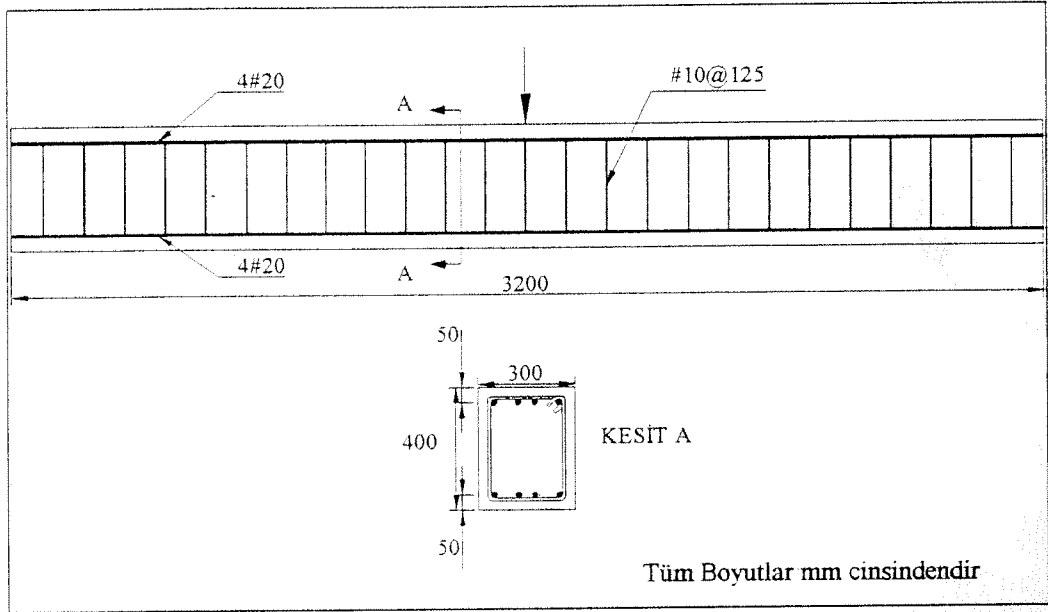


Figure 1.10 Arayüz elemanı kesme davranışı için kullanılan betonarme kiriş detayları.

Sistematik bir deneme yanılma yöntemi ile gerçekleştirilen analitik çalışmada betonarme kiriş Şekil 1.11 de görüldüğü gibi 8 adet arayüz elemanı kullanılarak modellenmiş ve deneysel çalışmada olduğu gibi iki ucu basit mesnetli ve ankastre olarak alınarak kiriş orta noktasından noktasal bir yük ile yüklenmiştir.

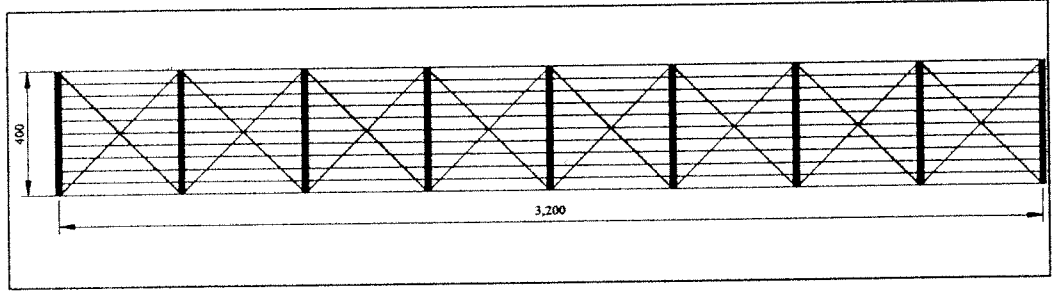


Figure 1.11 Betonarme kirişin betonarme arayüz elemanları ile oluşturulan analitik modeli.

Kesme davranış parametreleri  $\alpha_c$  ve  $\alpha_s$  için yapılan bu çalışmada parametre değerleri sistematik olarak değiştirilmiş ve elde edilen sonuçlar deneysel veriler ile karşılaştırılarak yeniden ayarlanmıştır. Deneysel veriler ile analitik çözümler arasındaki en iyi uyum  $\alpha_c=0.15$  ve  $\alpha_s=0.20$  değerleri için elde edilmiş olup sonuçlar Şekil 1:12 - 1.15 de gösterilmiştir.

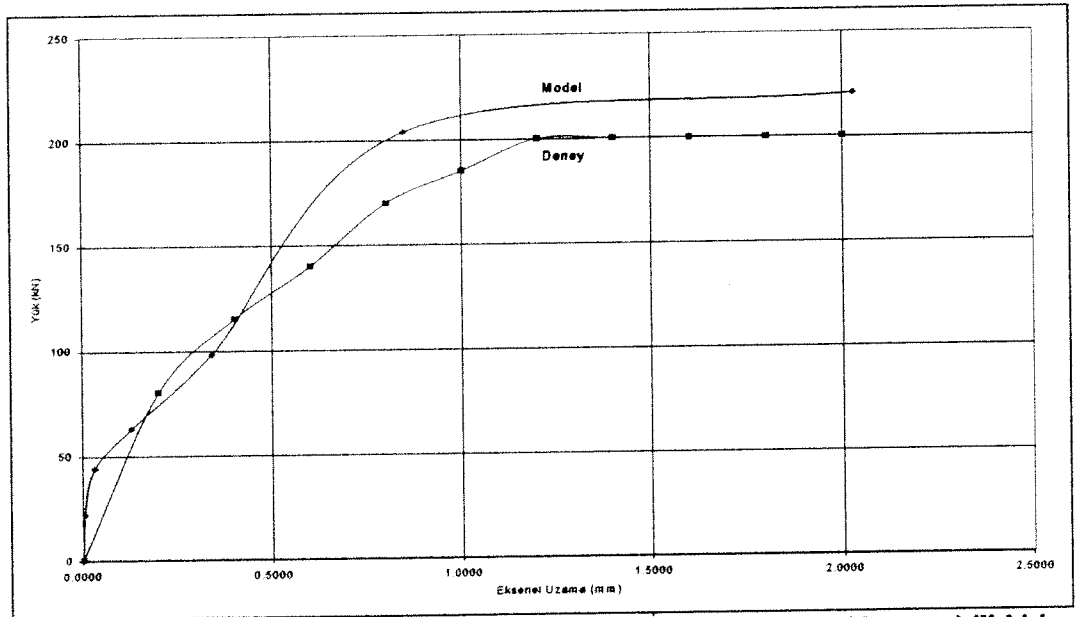


Figure 1.12 Basit mesnetli kiriş için yük-eksenel boy değişimi(kiriş ekseni boyunca) ilişkisi.

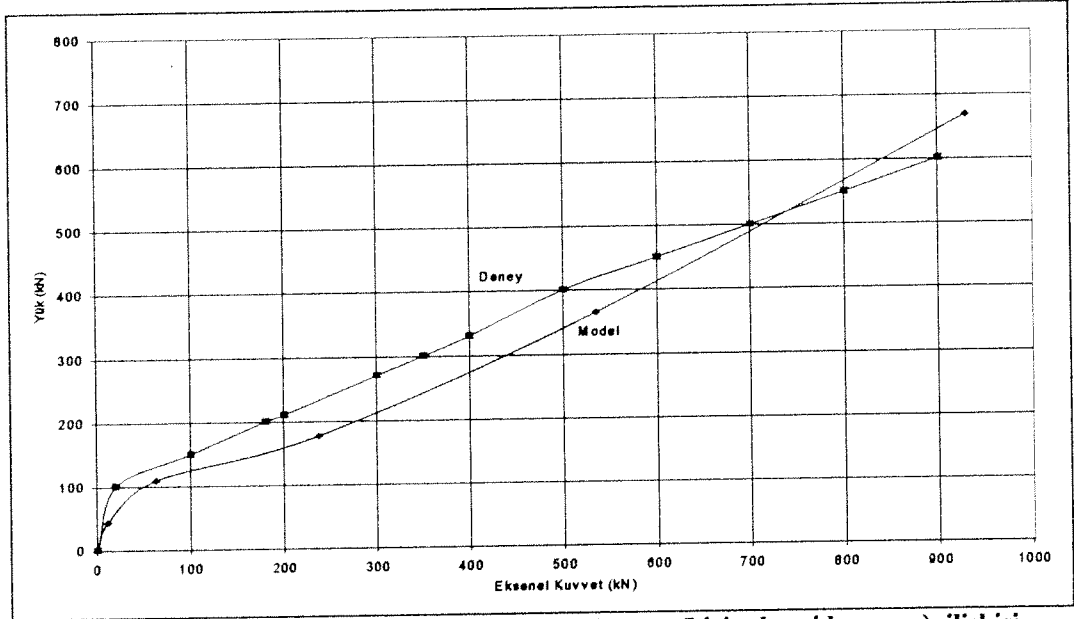


Figure 1.13 İki ucu ankastre kiriş için yük-eksenel kuvvet (kiriş eksenine boyunca) ilişkisi.

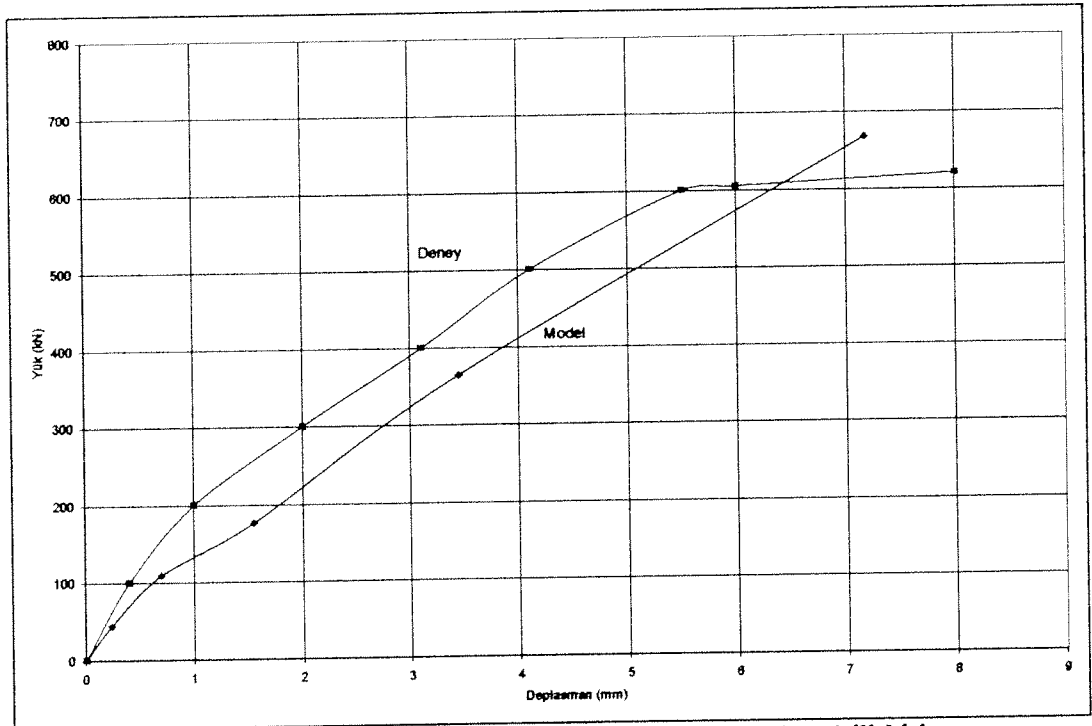


Figure 1.14 İki ucu ankastre kiriş için yük-deplasman(kiriş orta noktası) ilişkisi.



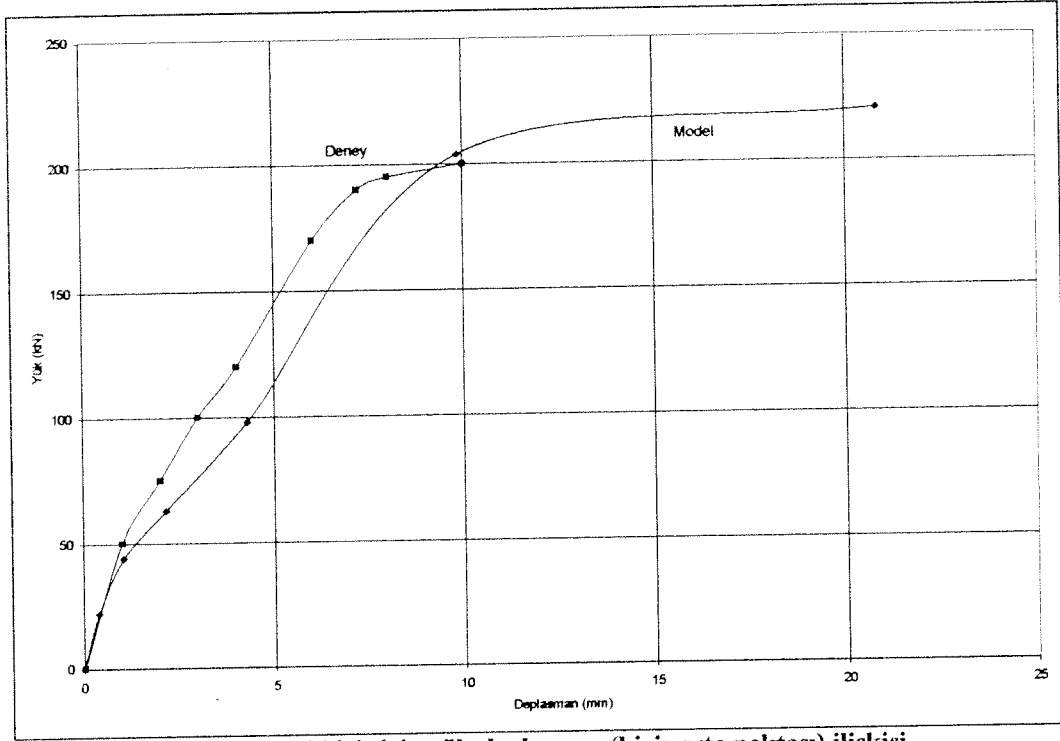


Figure 1.17 Basit mesnetli kiriş için yük-deplasman (kiriş orta noktası) ilişkisi

Kalibrasyon sonrası belirlenen model parametreleri Tablo 1.1 de verildiği gibidir.

Tablo 1.1. Elasto-Plastik Arayüz Elemanı Modeli Parametreleri

Eksenel Davranış Düzeltme Faktörü ( $\alpha_a$ )	1.00
Beton Kesit Etkin Kesme Alanı Oranı ( $\alpha_c$ )	0.15
Kesit Etkin Etriye Alanı Katsayısı ( $\alpha_s$ )	0.20

### 1.5 Betonarme Arayüz Elemanı Verifikasyon Çalışması

Modelin geçerliliğini göstermek üzere Vecchio ve Emara[7] tarafından deneysel olarak çalışılıp sonuçları rapor edilen tek açıklıklı ve iki katlı bir düzlem çerçeve örnek çalışma olarak seçilmiştir. Çerçevenin açıklığı 3500mm ve kat yükseklikleri 2000mm olup toplam yüksekliği 4600mm dir. Tüm kolon ve kirişler 300mm genişlikte ve 400mm derinliktedir. Çerçeve detayları Şekil 1.18 de görülmektedir.

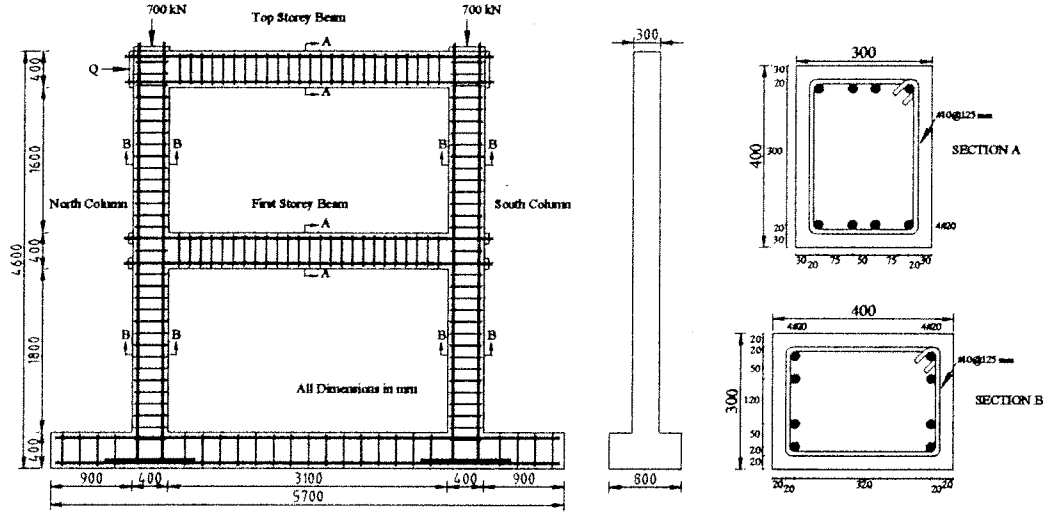


Figure 1.18 Betonarme arayüz elemanı test çalışmasında kullanılan betonarme çerçeve[7]

Tüm kesitler benzer şekilde donatılandırılmış olduğundan betonarme arayüz elemanları kullanılarak analitik modelinin oluşturulmasında büyük kolaylık sağlamaktadır. Kesitlerde dörder adet #20 nervürlü demir alt ve üst donatı olarak kullanılmış olup gövde donatısı olarak 125mm aralıklı #10 kapalı etriye kullanılmıştır. Kolonlar için beton kabul 20mm olup kirişler için bu değer 30mm dir.

Deneysel araştırma kapsamında yapılan incelemelerde boyuna donatı için malzeme özellikleri akma sınırı 418 MPa kopma gerilmesi ise 596 MPa olarak saptanmıştır. Bu değerler gövde donatısı için biraz daha yüksek olmakla birlikte betonarme arayüz elemanları ile gerçekleştirilen analitik modelde bu değerler tüm donatılar için aynı alınmıştır. Donatı elastik modülü 192,500 MPa ve pekleşme modülü ise 3100 MPa olarak verilmektedir.

Deneyler sırasında çerçeve kolonlara uygulanan 700kN sabit yükler altında üst kat seviyesinde monotonik olarak artan bir yatay yük altında test edilmiştir.. Deneyler ile ilgili diğer ayrıntılar Vecchio ve Emará[7] tarafından verilmektedir.

### 1.5.1 Deneysel Sonuçlar

Yükleme sırasında çerçevede ilk çatlaklar 52.5 kN yük seviyesinde oluşmuştur. İlk eğilme çatlakları birinci kat kirişinde sol alt yüzde ve sağ üst yüzeyde oluşmuştur. Yük-deplasman eğrisindeki ilk görünür rijitlik kaybı yine bu yük seviyesinde gözlenmiştir. Kolonların tabanındaki eğilme çatlakları ilk olarak 145 kN yük seviyesinde oluşmuştur. Yine bu yük seviyesinde birinci kat kirişi gövdesinde kesme çatlakları görülmeye başlanmıştır.

Yük seviyesi 264 kN düzeyine ulaştığında birinci kat kirişi sol alt yüzündeki boyuna donatıda akma başlamış yük 287 kN düzeyine ulaştığında ise kirişin sağ üst donatısakmaya başlamıştır. Kolonların tabanındaki boyuna donatı akması ile birinci kat kirişinin her iki ucundaki alt ve üst boyuna donatının akması ve betonun ezilerek mafsallaşmanın oluşması yük seviyesi 323 kN düzeyine yaklaşırken olmuştur. Çerçevenin taşıyabildiği en son yatay yük 332 kN olmuş ve bu yük seviyesinde %2.0 lik bir yanal deplasman oluşmuştur. Göçme mekanizması kiriş uçları ile kolon tabanlarında düktil mafsallaşma şeklinde gerçekleşmiş olup mafsal oluşumu maksimum yük seviyesine yaklaşıldıkça hızlanarak devam etmiştir.

### 1.5.2 Elastik Çerçeve/Betonarme Arayüz Elemanları Analitik Modeli

Deneysel çalışmada kullanılan betonarme çerçeveyi analitik olarak incelemek üzere elastik çerçeve ve betonarme arayüz elemanlarını içeren ve birleşimleri elasto-plastik mafsallı çerçeve sistemlerin yük altındaki doğrusal olmayan analizini gerçekleştirecek obje tabanlı bir yazılım Object Pascal programlama dili kullanılarak oluşturulmuştur. Bu yazılım için Delphi programlama ortamında bir de arayüz geliştirilmiştir. Eleman kütüphanesinde *Elastik Çerçeve ve Betonarme Arayüz* elemanları bulunan bu yazılımın ayrıntıları ve listesi EK A da verilmektedir. Yukarıda tanımlanan deneysel çerçeve doğrusal elastik çerçeve elemanları ve bu çalışmada geliştirilen betonarme arayüz elemanları kullanılarak analitik olarak modellenmiştir. Elasto-plastik betonarme arayüz (yarı-rijit

birleşim) elemanları Şekil 1.19 da görüldüğü gibi plastikleşmenin beklendiği elastik çerçeve elemanlarının uçlarına yerleştirilmiştir.

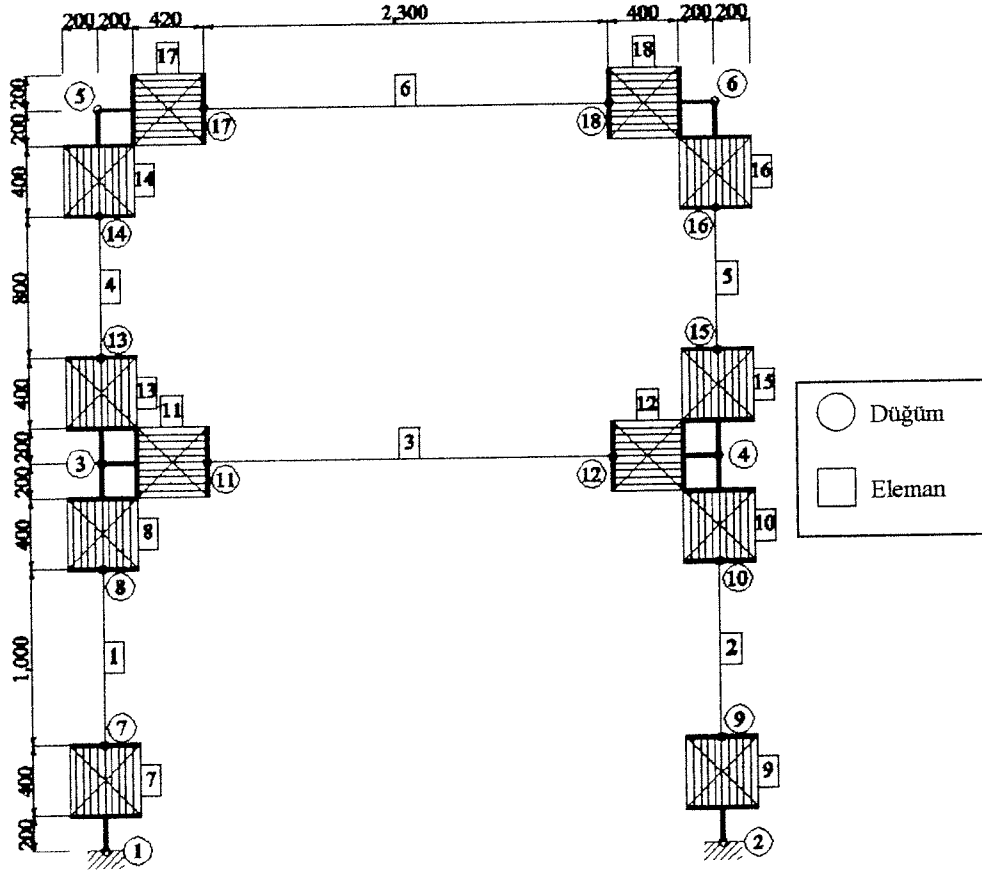


Figure 1.19 Betonarme deneysel test çerçevesi analitik modeli

### 1.5.3 Analitik Model Sonuçları

Oluşturulan matematiksel modelin beş adımda uygulanan yük altındaki doğrusal olmayan analiz sonuçları Tablo 1.2 de özetlenmektedir. Bu proje kapsamında geliştirilen yazılım kullanılarak gerçekleştirilen sayısal çözümlmeye ait veri dosyası ile yazılım tarafından üretilen detaylı çıktılar Ek B de verilmiştir. Analizler sırasındaki yükleme adımları çözüm algoritması tarafından otomatik olarak belirlenmiştir.

**Tablo 1.2 Betonarme Test Çerçevesi Analitik Modeli Analiz Sonuçları**

Yük Adımı	Yük (kN)	El. No	Beton Katmanlar (t: çekme hasarı, c: basınç hasarı)										Donatı (y: akma, h: pekleşme)		Çaprazlar					
															Beton		Çelik			
			1	2	3	4	5	6	7	8	9	10	Top	Bottom	1	2	1	2		
1	67.4	7	t	t																
		9	t																	
		11							t	t	t	t								
		12	t	t	t	t														
		17											t							
2	140.2	7	t	t	t	t	t	T								t				
		9	t	t	t	t	t									t				
		11					t	T	t	t	t	t					t			
		12	t	t	t	t	t	T									t			
		14	t	t	t	t											t			
		16	t	t																
		17						t	t	t	t	t						t		
		18	t	t	t	t	t										t			
3	287.1	7	t	t	t	t	t	t								t				
		8																t		
		9	t	t	t	t	t	t									t			
		10															t			
		11				t	t	t	t	t	t	t			y			t		
		12	t	t	t	t	t	t	t					y			t			
		13	t	t	t	t											t			
		14	t	t	t	t	t	t									t			
		15																t		
		17					t	t	t	t	t	t						t		
18	t	t	t	t	t	t									t					
4	326.7	7	t	t	t	t	t	t					y			t		t		
		8															t			
		9	t	t	t	t	t	t	t					y			t		t	
		10															t			
		11			t	t	t	t	t	t	t	t			y			t		t
		12	t	t	t	t	t	t	t	t				y			t		t	
		13	t	t	t	t	t										t			
		14	t	t	t	t	t	t									t			
15																t				

Yük Adımı	Yük (kN)	El. No	Beton Katmanlar (t: çekme hasarı, c: basınç hasarı)										Donatı (y: akma, h: pekleşme)		Çaprazlar						
													Top	Bottom	Beton		Çelik				
			1	2	3	4	5	6	7	8	9	10			1	2	1	2			
		16	t	t	t	t	t											t			
		17			t	t	t	t	t	t	t	t				y		t		t	
		18	t	t	t	t	t	t	t	t				y		t		t			
5	348.1	7	t	t	t	t	t	t	t	t			h		t		y				
		8													t						
		9	t	t	t	t	t	t	t		c	h	y	t		y					
		10	t	t											t						
		11			t	t	t	t	t	T	t		h		t		h				
		12	t	t	t	t	t	t	t			h		t		h					
		13	t	t	t	t	t	t						t							
		14	t	t	t	t	t	t						t							
		15												t							
		16	t	t	t	t	t							t							
		17			t	t	t	t	t	T	t		h		t		h				
		18	t	t	t	t	t	t	t			h		t		h					
6	Göçme																				

Analitik model sonuçlarının deneysel sonuçlar ile oldukça yakın bir uyum içinde olduğu görülmektedir. Deneysel çalışmada olduğu gibi ilk eğilme çatlakları birinci kat kirişinin sol alt (eleman 11) ve sağ üst (eleman 12) yüzlerinde 67.4 kN yük seviyesinde oluşmaktadır. Yük seviyesi 140 kN düzeyine ulaştığında kolonların alt bölgelerinde eğilme çatlakları ile ilk kesme çatlakları (çaprazların göçmesi) oluşmaktadır. Yük seviyesi 287 kN düzeyine ulaştığında birinci kat kirişi sol alt ve sağ üst boyuna donatılarında ilk akma gözlenmektedir. Kolon tabanlarındaki akma yük seviyesi 327 kN düzeyine yaklaşırken gerçekleşmekte ve çerçeve en son 348 kN yük taşıyabilmektedir.

Deneysel sonuçlar ile analitik model kullanılarak elde edilen analiz sonuçları aşağıda Şekil 1.20-1.27 arasında grafiksel olarak özetlenmektedir. Analitik model ile Vecchio ve Emara[7] deneysel sonuçları arasındaki uyum grafiklerden de açıkça görülmektedir.

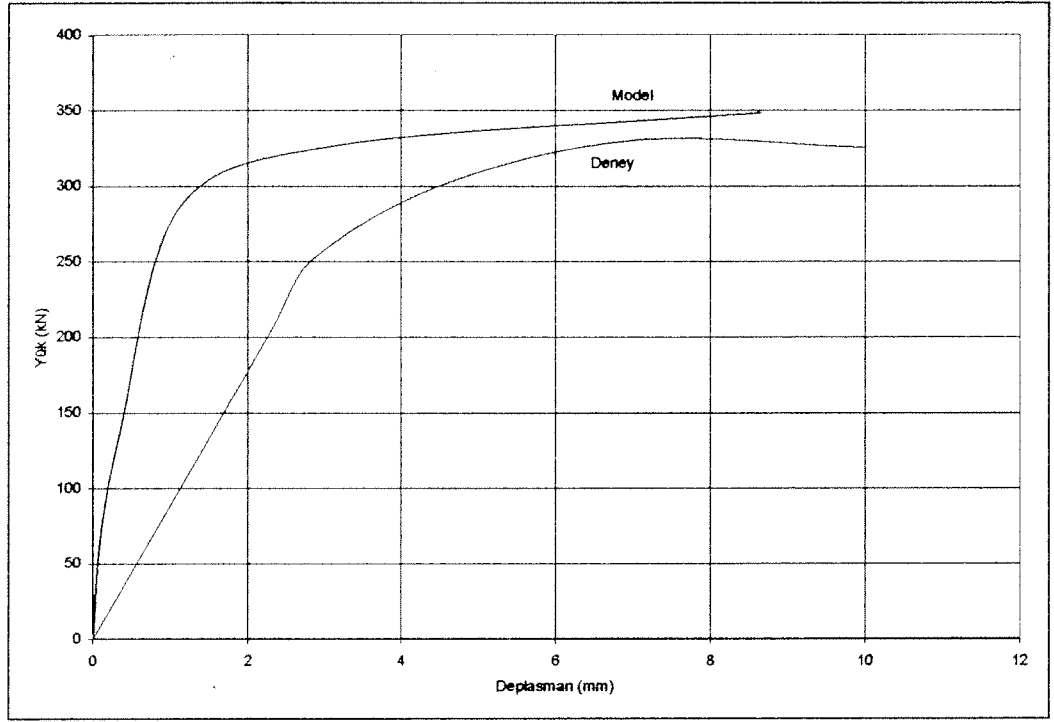


Figure 1.20 Birinci kat kirişi aksenal uzaması

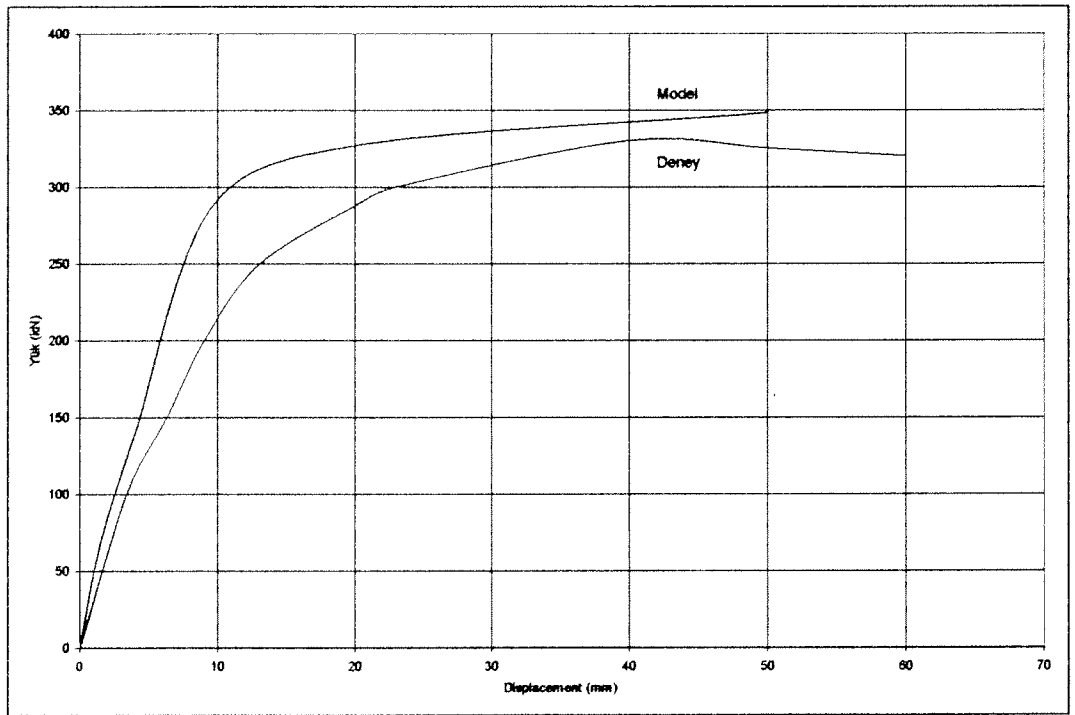


Figure 1.21 Birinci kat seviyesi yanal deplasmanı

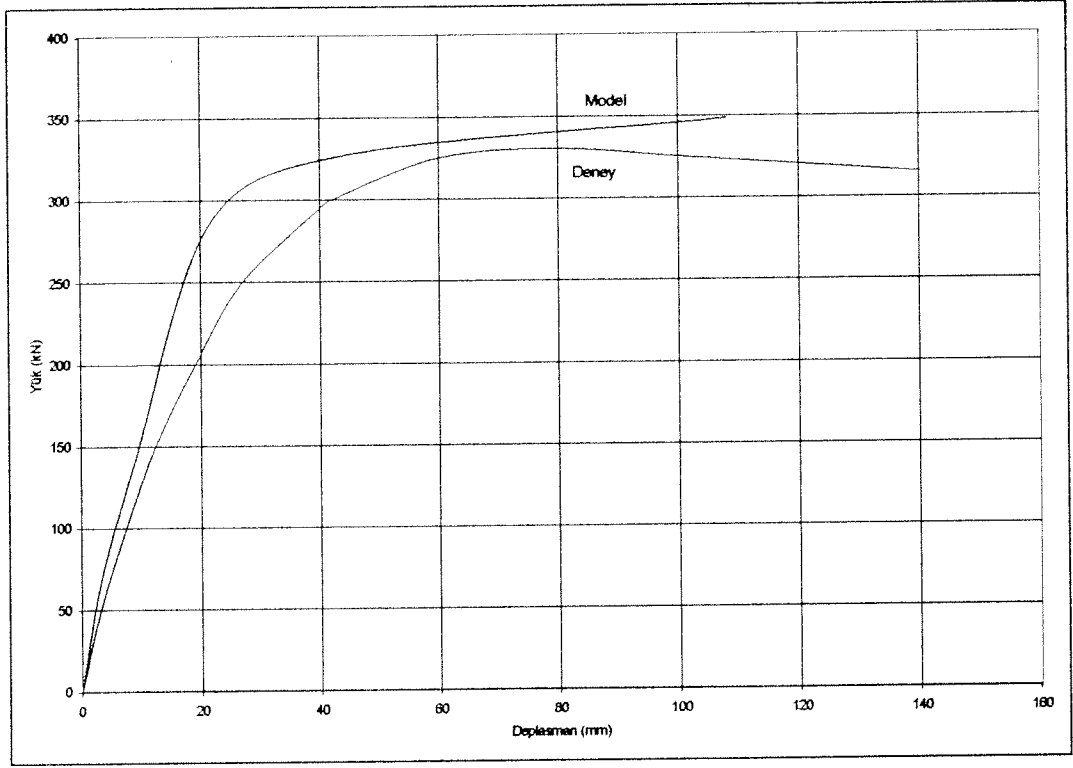


Figure 1.22 İkinci kat yanal deplasman

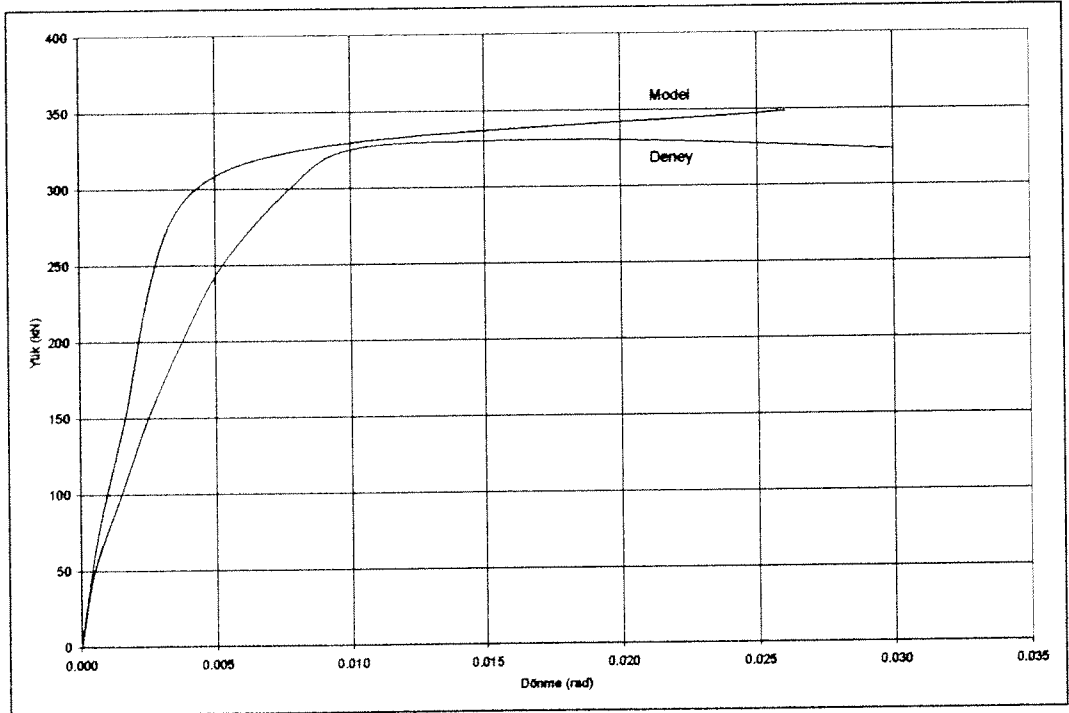


Figure 1.23 Sol tepe birleşim noktasındaki dönme



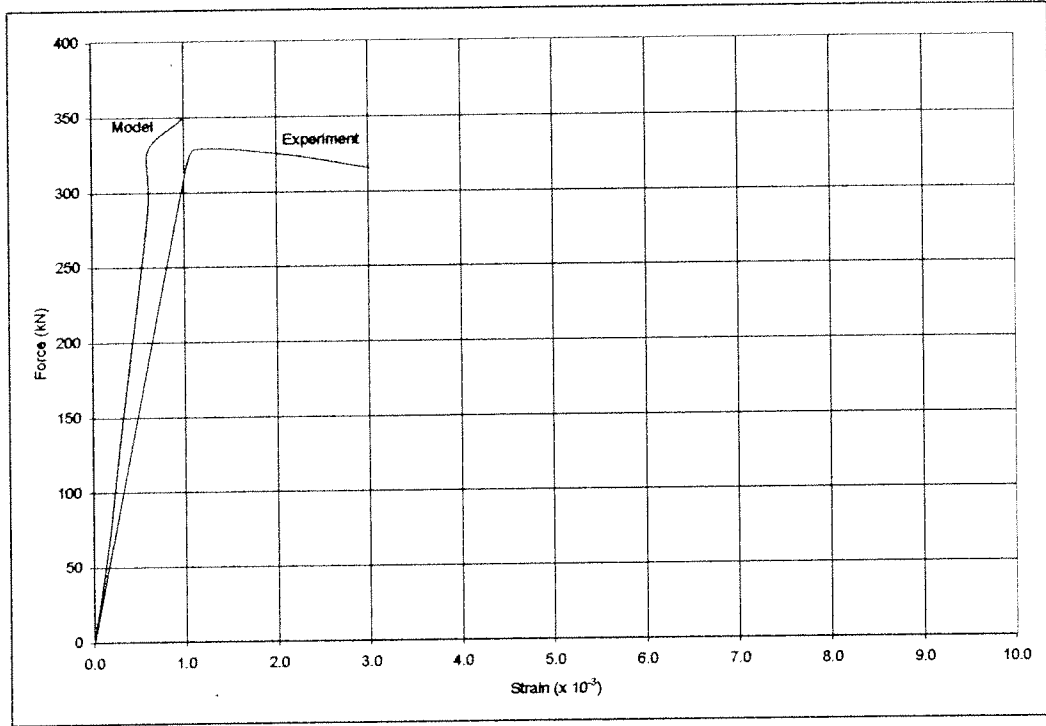


Figure 1.24 Birinci kat kirişi üst donatısındaki birim deformasyon (basınç)

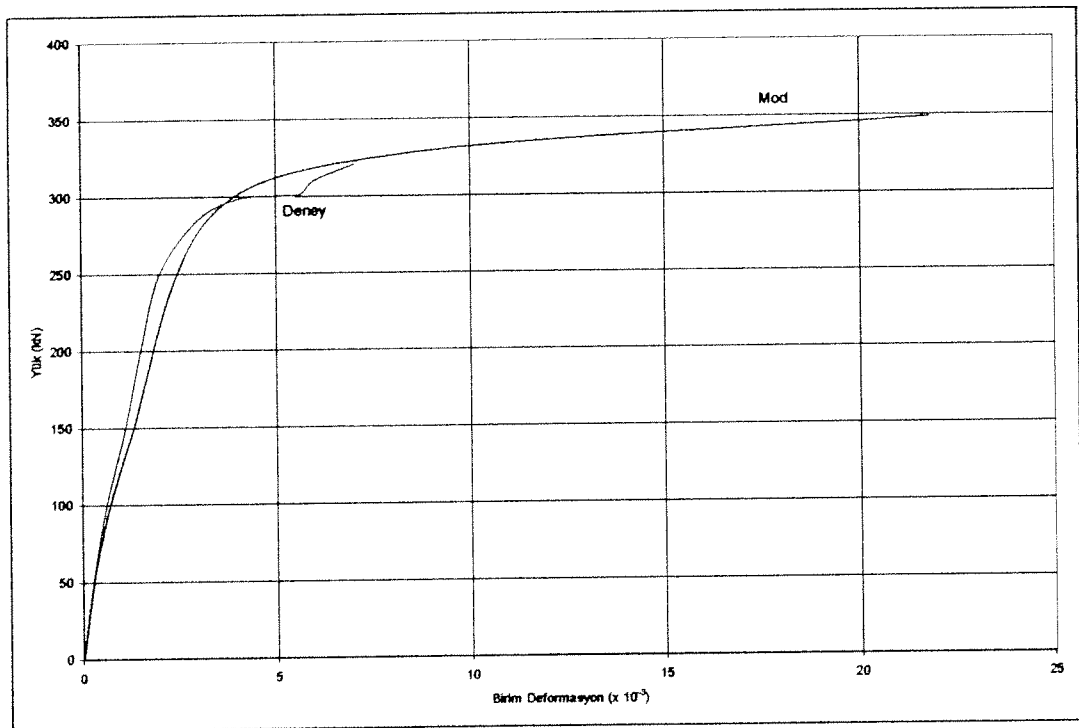


Figure 1.25 Birinci kat kirişi alt donatısındaki birim deformasyon (çekme)

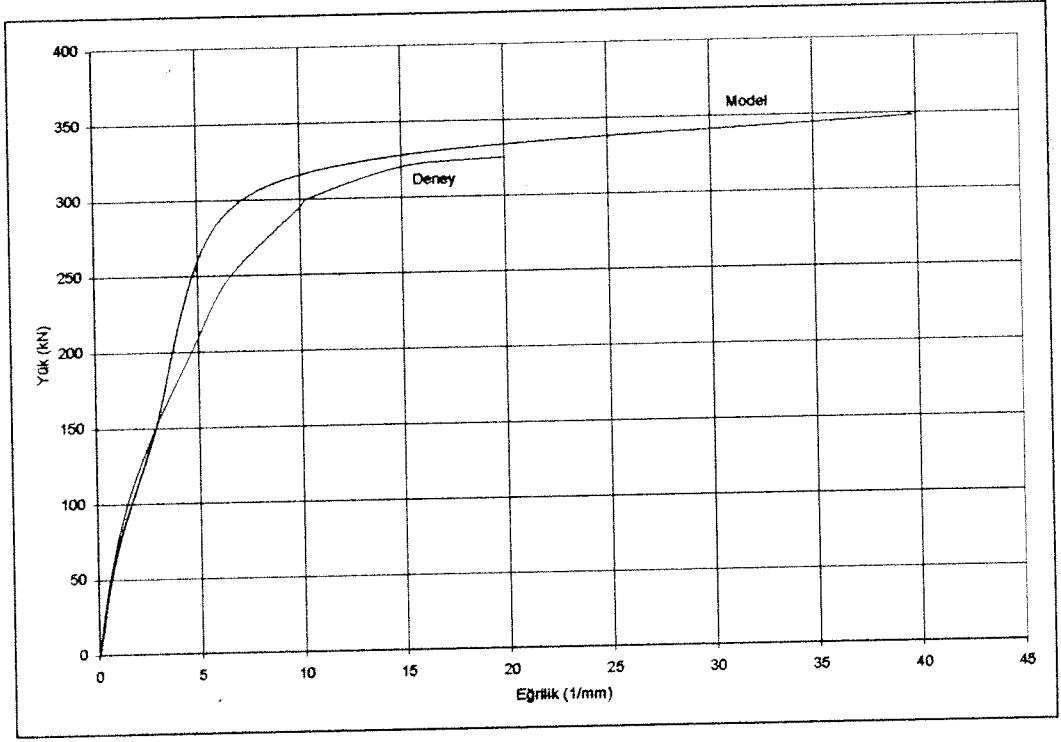


Figure 1.26 Sol kolon tabanındaki eğrilik

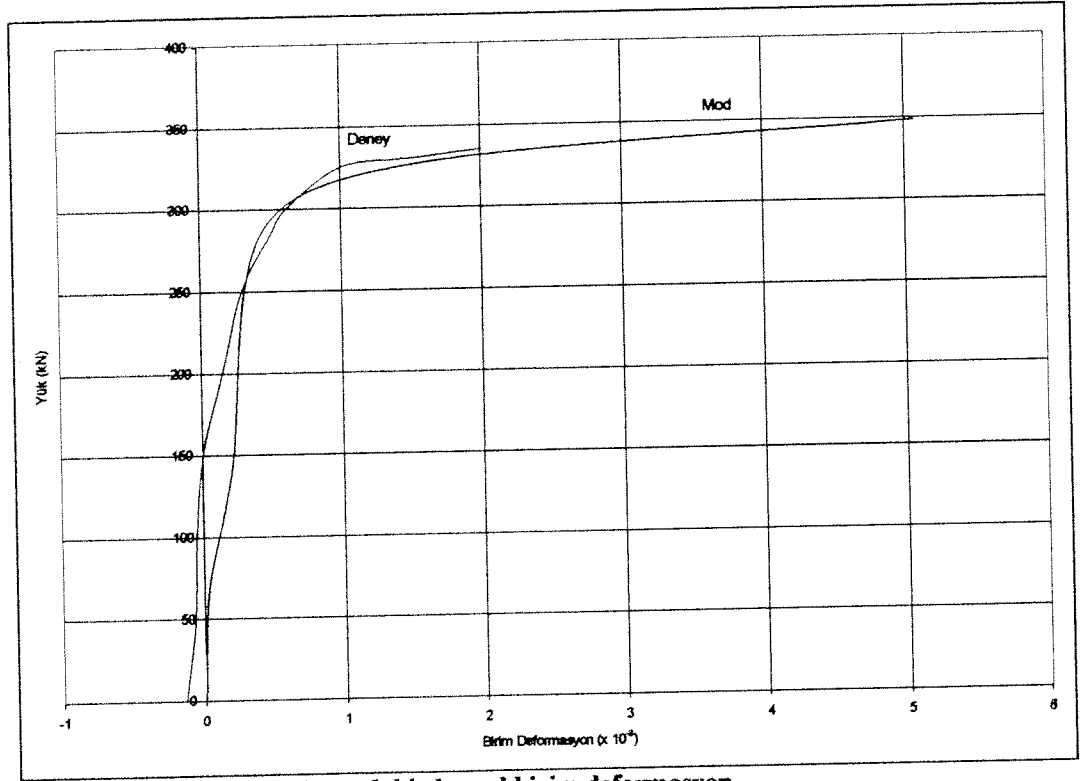


Figure 1.27 Sol kolon tabanındaki aksinel birim deformasyon

## BÖLÜM 2

### DÜZLEM PANEL MAKRO ELEMANI

#### 2.1 Giriş

Günümüzde boşluklu ve boşluksuz perde duvar ve döşemeler gibi iki boyutlu elemanlar içeren yapısal sistemlerin sonlu elemanlar yöntemiyle analizinde ciddi güçlüklerle karşılaşmakta ve sonuçta mühendisler genellikle daha basit ve zayıf modellere yönelmektedirler. Yapısal sistemlerin sonlu elemanlar yaklaşımı ile modellenmesi ve çözümlenmesi sürecinde karşımıza çıkan sorunlar başlıca iki grupta toplanabilir. Bunlardan birincisi büyüyen ve karmaşıklaşan yapısal sistemlerle hızla artan serbestlik derecelerinin günümüzün modern bilgisayarları için dahi bellek ve hız problemlerine neden olmasıdır. Bu sorun yapısal sistemin çubuk elemanlar yanında plak ve/veya kabuk gibi düzlem elemanlar da içermesi durumunda hızla gündeme gelmektedir. İkinci problem ise sistemin sonlu elemanlar modelinin hazırlanması ve gerçekleştirilen çözümleme sonrası üretilen yüksek hacimli tepki çıktılarının değerlendirilmesi, yorumlanması ve tasarıma temel teşkil edecek tepkiler haline dönüştürülmesi aşamalarında ortaya çıkmaktadır.

Sonlu elemanlar modelinin hazırlanması aşamasında karşılaşılan temel güçlükler şöyle özetlenebilir.

- Sonlu elemanlar ağını tanımlayacak düğüm noktalarının uzaydaki konumlarının hesaplanması ve bunun sisteme tanıtılması.
- Belirlenen düğüm noktaları bazında sistemin sonlu elemanlara bölünmesi ve her bir sonlu eleman ile düğüm noktaları arasındaki ilişkinin tanımlanması. Sürekli ortamın geometrik olarak multiply connected (boşluklu perde veya delikli döşeme gibi) olması durumunda güçlükler daha da artmaktadır. Bu aşamada karşılaşılan diğer bir problem de her bir yapısal eleman için ayrı ayrı oluşturulan sonlu elemanlar modelindeki sınır düğüm noktalarının elemanların diğer yapısal elemanlarla oluşturduğu ara yüzler boyunca ortak olması zorunluluğundan kaynaklanan güçlüktür.
- Yapısal sistemin maruz kaldığı dış etkilerin çoğunlukla sonlu elemanlar modelini oluşturan düğüm noktaları referans alınarak tanımlanması zorunluluğu. Bu zorunluluk mühendisin bir dizi ön hesaplama yapması gereğini doğurmaktadır.

Mühendisin büyük bir enerji ve sabrını gerektiren bu işlemler son derecede sıkıcı ve bir o kadar da maddi hata yapmaya elverişli bir süreçtir.

Sonuçların değerlendirilmesi de genellikle bir dizi son işlemi gerektirmektedir. Bu gereksinim yapısal sistemin boyutu ve karmaşıklığı, modelde kullanılan sonlu elemanlar ağının yoğunluğu ve kullanılan sonlu eleman yapısının karmaşıklığı ile hızla artmaktadır. Plak/Kabuk sonlu elemanları doğası gereği bu gereksinimi zorlayan bir yapıya sahiptir ve bu yüzden analizcinin enerjisinin büyük bir bölümü de bu aşamada harcanmaktadır.

Özellikle birleşimleri kaçınılmaz olarak yarı-rijit olan ve bundan ötürü de deprem gibi ciddi yanal zorlamalar altında oldukça esnek davranıp aşırı deformasyonlara maruz kalması beklenen prefabrik sistemlerin bu zaafını gidermek üzere yapısal sistemin perde duvarlarla takviyesi çoğu zaman gerekli görülmektedir. Raporun

bu bölümünde prfabrik yapısal sistemlerde olduğu gibi kısmen veya tünel kalıp tekniği ile üretilen yapılar gibi tamamen düzlem panel taşıyıcı duvarlardan oluşan yapısal sistemlerin sonlu elemanlar yöntemi ile çözümlenmesinde karşılaşılan birinci grup probleme bir çözüm getirmek üzere geliştirilen bir Panel Makro Elemanın [8] formülasyonu ve bir yapısal analiz programına yerleştirmesi anlatılmaktadır. Panel makro elemanının ayrıntıları bu proje kapsamında A.Hakan Argeşo[9] ve H.Bahadır Bahat[10] tarafından gerçekleştirilen iki ayrı yüksek lisans tez çalışmasının raporlarında verilmiştir.

## 2.2 Panel Makro (Süper) Elemanı

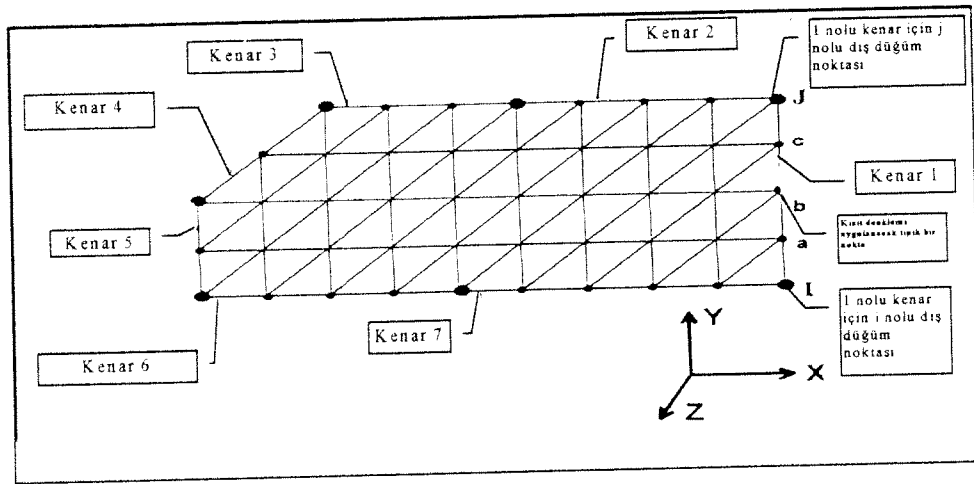
Makro eleman yalnızca elemanı oluşturan dış kenarları tanımlayan noktalarda aktif serbestlikler içerecek şekilde tanımlanmış olup, üçgen kabuk elemanlarıyla tanımlı bir alt yapıdan statik yoğunlaştırma yöntemi ile önce panel iç düğüm noktalarındaki serbestliklerin elimine edilerek yapısal eleman rijitliğinin eleman sınır düğümlerine yoğunlaştırılması, daha sonra da eleman kenarları boyunca oluşturulacak kısıt şartları uygulanarak elemanın yapı içinde diğer elemanlarla ara yüzünü oluşturan kenarları tanımlayan köşe noktalarına yoğunlaştırılması fikrine dayanmaktadır.

Makro elemanın temel yapı taşı Kirchhoff plak eğilme teorisine dayanan DKT[11-15] plak eğilme elemanı ile üçgen geometrili ve burulma serbestliği içeren Allman[16] düzlem gerilme (membran) elemanının sayısal olarak süperpoze edilmesi ile elde edilen üçgen bir kabuk sonlu elemanıdır. Temel eleman olarak üçgen eleman topolojisinin seçimi ise başlıca iki nedene dayanmaktadır.

1. Üçgen geometrili elemanlar düzgün olmayan geometrili düzlemlerin tanımlanmasında dörtgen elemanlara göre daha esnek bir yapıya sahiptirler. Geometrik olarak doğrusal kenarlı herhangi bir poligon düzlemi matematiksel olarak tamamıyla bir dizi üçgen alt düzlemlerin birleşimiyle elde etmek mümkündür.

2. Düzlem içinde verilen N adet nokta ile oluşturulacak üçgenlerin belirlenmesi problemi matematikçiler tarafından da üzerinde çokça çalışılmış bir matematiksel problem olup literatürde bu konuda güçlü ve oturmuş algoritmalar mevcuttur [17-19]. Dolayısı ile düzlem kabukların üçgen elemanlara bölünmesi işlemini etkin bir şekilde gerçekleştirmek mümkündür. Burada ilginç olan bir nokta da matematiksel çözümün ürettiği üçgenlerin sonlu elemanlar yöntemi açısından da arzu edilen ve verilen N adet noktanın uzaydaki dağılımı göz önüne alındığında mümkün olan en uygun topolojik özelliklere sahip olduğu hususudur.

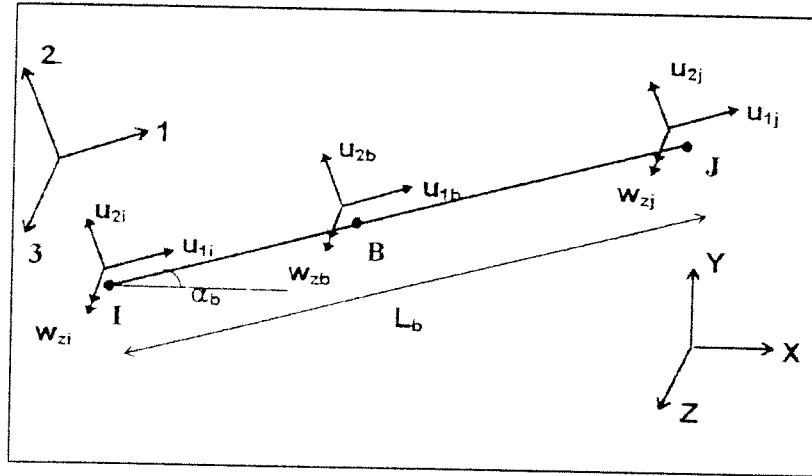
Tipik bir makro eleman aşağıda Şekil 2.1 de görülmektedir. Şekilde daha koyu ve büyük olarak gösterilen ve I ve J olarak etiketlenen noktalar makro elemanın tipik bir kenarı boyunca tanımlanmış ana düğüm noktalarını (Bağlantı Düğüm Noktaları) göstermektedir. Tipik bir kenar boyunca a,b,c olarak etiketlenen ve sayıları makro eleman için seçilen sonlu elemanlar ağı yoğunluğuna bağlı olarak değişen noktalar ise kısıt şartları empoze edilerek elimine edilecek sınır ara düğüm noktalarını göstermektedir. Makro eleman için seçilen sonlu elemanlar ağı yoğunluğuna bağlı olarak mevcut diğer tüm düğüm noktaları ise ilk aşamada statik yoğunlaştırma ile elimine edilecek iç düğüm noktalarıdır.



Şekil 2.1 Tipik bir Panel Makro Elemanı ve topolojik tanımlar.

## 2.2.1 Makro Eleman Kenar Deplasman Fonksiyonları

Kartezyen XY düzlemi içinde tanımlanmış makro elemana ait tipik bir kenar boyunca oluşan deplasman ve dönmeler bu kenarın uçları I ve J düğüm noktalarında tanımlı toplam 12 serbestliğin fonksiyonları olarak ifade edilebilir. Bu amaçla önce tipik bir kenar için lokal bir kartezyen 123 koordinat sistemi tanımlayalım. Panel kenarını I düğümünden J düğümüne tanımlanmış bir vektör olarak düşünersek bu doğrultuyu lokal 1 yönü olarak, panel düzleminin normalini ise lokal 3 yönü olarak tanımlayalım. Bu durumda 3 yönü panel koordinat sisteminin Z doğrultusu olup lokal 2 doğrultusu sağ el kuralına göre tanımlanmış olur. Bu lokal koordinat sisteminde panelin I ve J uç düğüm noktalarında membran moduna ait düzlem içi ötelenme ve burulma serbestlikleri ( $U_1$ ,  $U_2$  ve  $\theta_3$ ) Şekil 2.2 de görülmektedir.



Şekil 2.2 Tipik bir kenar boyunca panel membran modu serbestlikleri

Kenar boyunca aksenal deplasmanların,  $U_1$ , doğrusal olarak değiştiği kabul edilirse

$$U_1(\xi) = N_1 U_{1I} + N_2 U_{1J} \quad (2.1)$$

$$N_1(\xi) = 1 - \xi \quad (2.2)$$

$$N_2(\xi) = \xi \quad (2.3)$$

$$\xi = s / L_B \quad (2.4)$$

olarak ifade edilebilir. Burada  $N_1$  ve  $N_2$  interpolasyon fonksiyonları,  $s$  kenar boyunca  $I$  düğüm noktasından ölçülen uzaklık,  $L_B$  ise kenarın toplam boyudur.

Düzlem içi  $U_2$  ötelenmesi için kübik bir değişim ön görülmektedir. Kenarın  $I$  ve  $J$  düğüm noktalarındaki yanal ötelenmeler ve düzlem dışı dönmeler (burulma serbestlikleri) cinsinden ve birinci derece Hermit polinomları [20] yardımı ile bu kübik değişim

$$U_2(\xi) = N_3 U_{2I} + N_4 \theta_{3I} + N_5 U_{2J} + N_6 \theta_{3J} \quad (2.5)$$

olarak ifade edilebilir. Burada  $N_3$ ,  $N_4$ ,  $N_5$  ve  $N_6$  interpolasyon fonksiyonları birinci derece Hermit polinomları olup

$$N_3(\xi) = (2\xi^3 - 3\xi^2 + 1) \quad (2.6)$$

$$N_4(\xi) = L_B(\xi^3 - 2\xi^2 + \xi) \quad (2.7)$$

$$N_5(\xi) = (-2\xi^3 + 3\xi^2) \quad (2.8)$$

$$N_6(\xi) = L_B(\xi^3 - \xi^2) \quad (2.9)$$

olarak tanımlanmıştır.

$\theta_3 = dU_2/ds$  olup zincir kuralı uygulanarak  $\theta_3(\xi) = dU_2(\xi)/d\xi \cdot d\xi/ds$  ve  $d(\xi)/ds = 1/L_B$  olduğu göz önüne alınırsa

$$\theta_3(\xi) = \underline{N}_3 U_{2I} + \underline{N}_4 \theta_{3I} + \underline{N}_5 U_{2J} + \underline{N}_6 \theta_{3J} \quad (2.10)$$



$$\underline{N}_3(\xi) = (6\xi^2 - 6\xi) / L_B \quad (2.11)$$

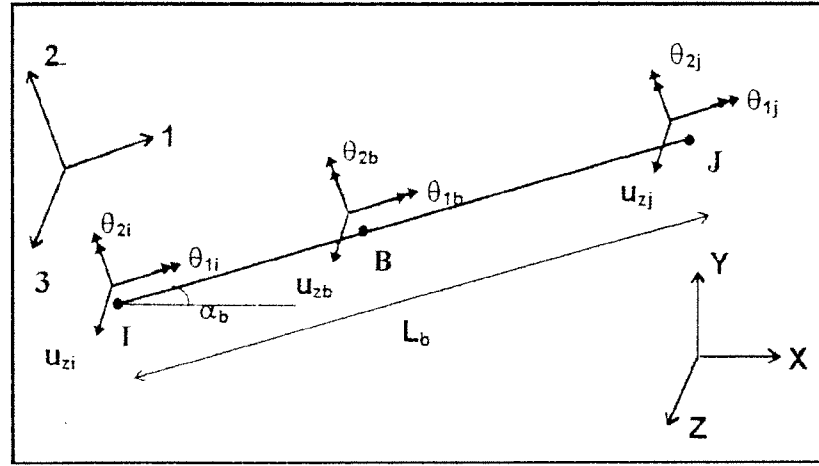
$$\underline{N}_4(\xi) = (3\xi^2 - 4\xi + 1) \quad (2.12)$$

$$\underline{N}_5(\xi) = (-6\xi^2 + 6\xi) / L_B \quad (2.13)$$

$$\underline{N}_6(\xi) = (3\xi^2 - 2\xi) \quad (2.14)$$

elde edilir.

Lokal koordinat sisteminde panelin I ve J uç düğüm noktalarında plak eğilme moduna ait düzlem dışı ötelenme ve düzlem içi dönme serbestlikleri ( $U_3$ ,  $\theta_1$  ve  $\theta_2$ ) de Şekil 2.3 de görülmektedir.



Şekil 2.3 Tipik bir kenar boyunca panelin plak eğilme modu serbestlikleri

Kenar boyunca aksenal dönmelerin,  $\theta_1$ , doğrusal olarak değiştiği kabul edilirse

$$\theta_1(\xi) = N_1\theta_{1I} + N_2\theta_{1J} \quad (2.15)$$

olarak ifade edilebilir.

Düzlem dışı  $U_3$  ötelenmesi için kübik bir değişim ön görülmektedir. Kenarın I ve J düğüm noktalarındaki düzlem dışı ötelenmeler ve kenarın düzlem içi normal

doğrultusundaki dönmeler cinsinden ve yine birinci derece Hermit polinomları yardımı ile bu kübik değişim (2.5) ifadesine benzer şekilde

$$U_3(\xi) = N_3 U_{3I} - N_4 \theta_{2I} + N_5 U_{3J} - N_6 \theta_{2J} \quad (16)$$

olarak ifade edilebilir. Burada interpolasyon fonksiyonları  $N_3$ ,  $N_4$ ,  $N_5$  ve  $N_6$  yukarıda (2.6-2.9) da tanımlanan birinci derece Hermit polinomlarıdır.

$\theta_3 = -dU_3/ds$  olup zincir kuralı uygulanarak  $\theta_3(\xi) = -dU_3(\xi)/d\xi \cdot d\xi/ds$  ve  $d(\xi)/ds = 1/L_B$  olduğu göz önüne alınırsa (2.10) ifadesine benzer şekilde

$$\theta_2(\xi) = -\underline{N}_3 U_{3I} + \underline{N}_4 \theta_{2I} - \underline{N}_5 U_{3J} + \underline{N}_6 \theta_{2J} \quad (17)$$

elde edilir. Burada interpolasyon fonksiyonları  $\underline{N}_3$ ,  $\underline{N}_4$ ,  $\underline{N}_5$  ve  $\underline{N}_6$  daha önce yukarıda (2.11-2.14) de tanımlandığı gibidir.

Panel Makro Elemanı kenarları boyunca köşe düğüm noktaları (Bağlantı Düğüm Noktaları) arasında yer alan düğüm noktaları deplasmanlarının yukarıda (2.1), (2.5), (2.10), (2.15), (2.16) ve (2.17) de verilen deplasman değişimlerine uymaya zorlanması sureti ile elde edilecektir. Bu şekilde iç düğüm noktaları rijitliklerinin statik olarak çevre düğüm noktalarına yoğunlaştırılması işlemine ilave olarak bu koşulların (Kısıt Denklemleri) da empoze edilmesi ile panel eleman rijitliği tamamen köşe düğüm noktalarına yoğunlaşmış olacak ve eleman dış dünyaya karşı sadece Bağlantı Düğüm Noktaları ile tanımlanmış bir Panel Makro Elemanı olarak görülecektir.

## 2.2.2 Kısıt Denklemlerinin Global Sistemde İfadesi

Yukarıda panelin bir kenarını tanımlayan iki köşe düğüm noktası ve bu kenar üzerinde yer alan herhangi bir nokta arasında tanımlanan kısıt denklemleri kenarın lokal eksen takımı 123 baz alınarak ifade edilmiştir. Kısıt denklemlerinin Panel Makro Elemanı rijitlik matrisinin oluşturulmasında kullanılabilmesi için makro eleman global eksen takımına göre ifadesi gerekmektedir.

Herhangi bir noktadaki membran moduna ait deplasmanları lokal koordinat sisteminde  $\{\bar{U}_M\}$  global koordinat sisteminde ise  $\{U_M\}$  ile gösterirsek panel üzerindeki herhangi bir düğüm noktasında

$$\{\bar{U}_M\} = \begin{Bmatrix} U_1 \\ U_2 \\ \theta_3 \end{Bmatrix} \quad \text{ve} \quad \{U_M\} = \begin{Bmatrix} U_x \\ U_y \\ \theta_z \end{Bmatrix} \quad (2.18)$$

vektörleri yazılabilir. Benzer şekilde panelin plak eğilme moduna ait deplasmanlar için de

$$\{\bar{U}_P\} = \begin{Bmatrix} U_3 \\ \theta_1 \\ \theta_2 \end{Bmatrix} \quad \text{ve} \quad \{U_P\} = \begin{Bmatrix} U_z \\ \theta_x \\ \theta_y \end{Bmatrix} \quad (2.19)$$

ifadelerini yazabiliriz. Bu durumda herhangi bir düğüm noktasına ait deplasman vektörünü kenar lokal eksen takımı 123 ve panel global eksen takımı XYZ bazında

$$\{\bar{U}\} = \begin{Bmatrix} \bar{U}_M \\ \bar{U}_P \end{Bmatrix} \quad \text{ve} \quad \{U\} = \begin{Bmatrix} U_M \\ U_P \end{Bmatrix} \quad (2.20)$$

olarak ifade edebiliriz. Dolayısı ile panelin I ve J bağlantı düğüm noktaları arasında tanımlı tipik bir kenarı üzerinde yer alan herhangi bir nokta için

yukarıda (2.1), (2.5), (2.10), (2.15), (2.16) ve (2.17) de verilen deplasman bağıntıları

$$\{\bar{U}_z\} = [\bar{T}_{IJ}] \{\bar{U}_{IJ}\} \quad (2.21)$$

şeklinde ifade edilebilir. Burada  $\{\bar{U}_{IJ}\}$  kenarın **I** ve **J** düğüm noktalarına ait deplasmanlar vektörü

$$\{\bar{U}_{IJ}\} = \begin{Bmatrix} \bar{U}_I \\ \bar{U}_J \end{Bmatrix} \quad (2.22)$$

olup **IJ** kenarına ait dönüşüm matrisi  $[\bar{T}_{IJ}]$

$$[\bar{T}_{IJ}] = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & N_4 & 0 & 0 & 0 & 0 & N_5 & N_6 & 0 & 0 & 0 \\ 0 & N_3 & N_4 & 0 & 0 & 0 & 0 & N_5 & N_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & N_3 & 0 & -N_4 & 0 & 0 & 0 & N_5 & 0 & -N_6 \\ 0 & 0 & 0 & 0 & N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 \\ 0 & 0 & 0 & -N_3 & 0 & N_4 & 0 & 0 & 0 & -N_5 & 0 & N_6 \end{bmatrix} \quad (2.23)$$

olarak ifade edilebilir. Panel düğüm noktası serbestlikleri panelin global XYZ ve kenara ait lokal 123 koordinat sistemleri arasında rotasyon matrisleri  $[R_M]$  ve  $[R_p]$  ile aşağıdaki şekilde ilişkilendirilebilir.

$$\{\bar{U}_M\} = [R_M] \{U_M\} \quad ; \quad [R_M] = \begin{bmatrix} \cos \alpha_y & \sin \alpha_y & 0 \\ -\sin \alpha_y & \cos \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

ve

$$\{\bar{U}_p\} = [R_p] \{U_p\} \quad ; \quad [R_p] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_y & \sin \alpha_y \\ 0 & -\sin \alpha_y & \cos \alpha_y \end{bmatrix} \quad (2.25)$$

Bu durumda panelin **I** ve **J** düğüm noktaları arasında tanımlı kenarı üzerindeki herhangi bir nokta için tanımlanan (2.21) ifadesi XYZ eksen takımına göre

$$\{U_i\} = [T_{ij}] \{U_j\} \quad (2.26)$$

şeklinde yazılabilir. Burada global eksen takımına göre dönüşüm matrisi  $[T_{ij}]$

$$[T_{ij}] = [R_{ij}]^T [\bar{T}_{ij}] [R_{ij}] \quad (2.27)$$

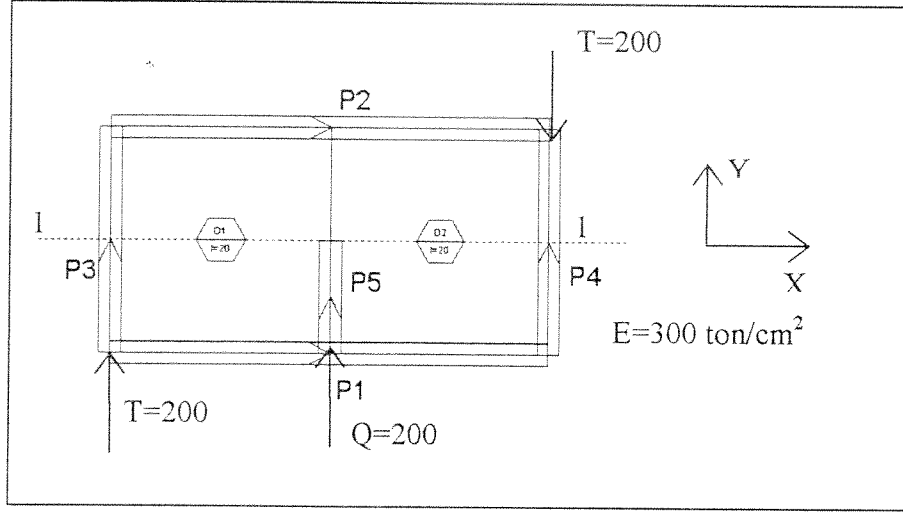
olup rotasyon matrisi  $[R_{ij}]$

$$[R_{ij}] = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \quad \text{ve} \quad [R] = \begin{bmatrix} R_x & 0 \\ 0 & R_y \end{bmatrix} \quad (2.28)$$

olarak tanımlanmıştır.

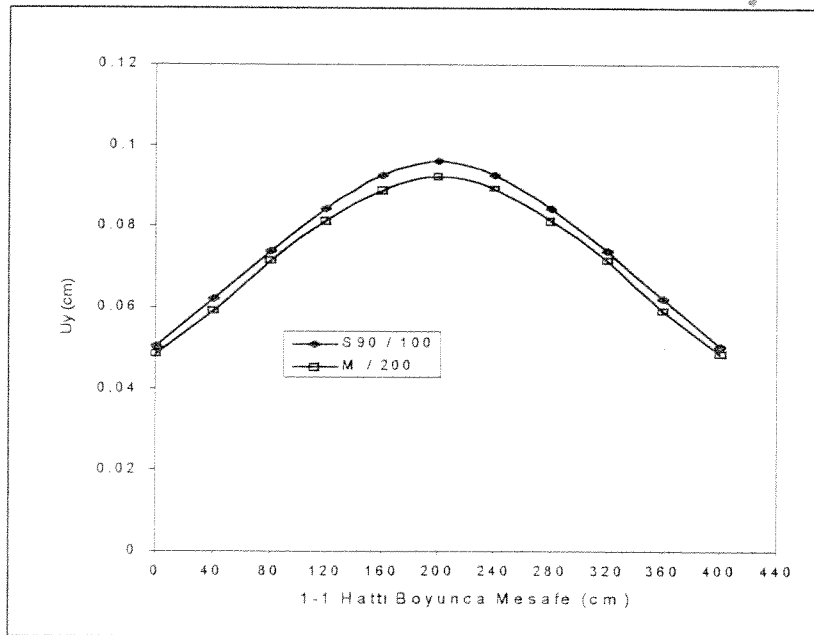
### 2.3 SAYISAL TEST PROBLEMİ

Geliştirilen panel makro elemanın kompleks zorlama ve deformasyon koşulları altındaki davranışını test etmek üzere Şekil 2.4 te görülen ve tümü ile perde duvarlardan ve elastik döşemelerden oluşan, tek katlı ve iki gözlü kapalı kutu tipi yapı kat seviyesinde kütle ve rijitlik merkezinden geçen 200 tonluk yanal yük (Q yüklemesi) ile yine kat seviyesinde, P3 ve P4 perdeleri boyunca kat burulması yaratacak şekilde birbirlerine ters yönlerde etkiyen herbiri 200 tonluk yükler (T yüklemesi) altında incelenmiştir. Yapısal sistem, P5 perdesi hariç tüm perde ve döşemeleri ikişer, P5 perdesi ise tek panel makro elemanı olmak üzere toplam 13 adet makro eleman ile modellenmiştir. Her makro eleman 20cm kalınlıkta olup iç yapı olarak 200 adet üçgen kabuk elemandan oluşmaktadır. Sonuçları popüler ve yerleşmiş SAP90[21] sonlu elemanlar programı sonuçları ile karşılaştırmak üzere yapıyı oluşturan panellerin her biri 100 adet dörtgen kabuk elemanı ile modellenmiştir.

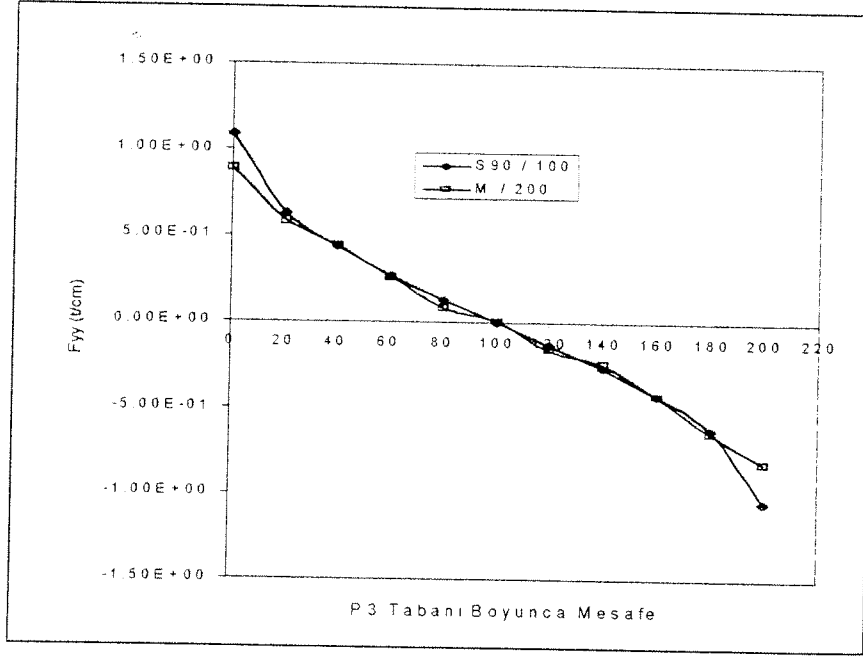


Şekil 2.4 İki göznlü kapalı kutu tipi test yapısı; plan görünüşü ve yüklemeler (Q ve T).

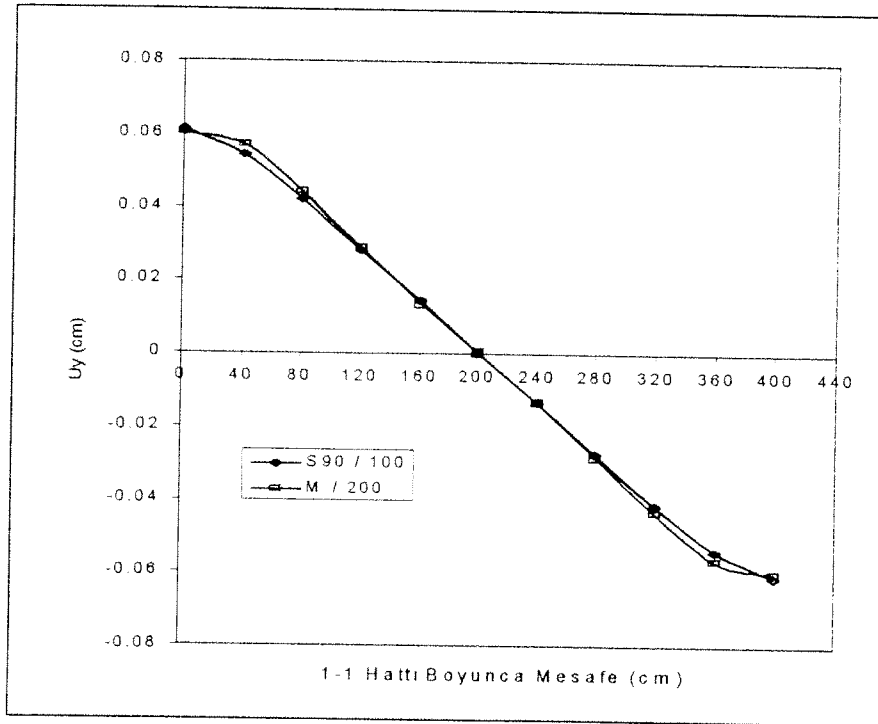
Q yüklemesi altında kat seviyesinde 1-1 hattı boyunca oluşan yanal deplasmanlar ile P3 paneli tabanında eğilmeden dolayı oluşan aksenal kuvvet dağılımı örnek tepkiler olarak Şekil 2.5 ve 2.6 da, T yüklemesi altında oluşan benzer tepkiler ise Şekil 2.7 ve 2.8 de gösterilmiştir.



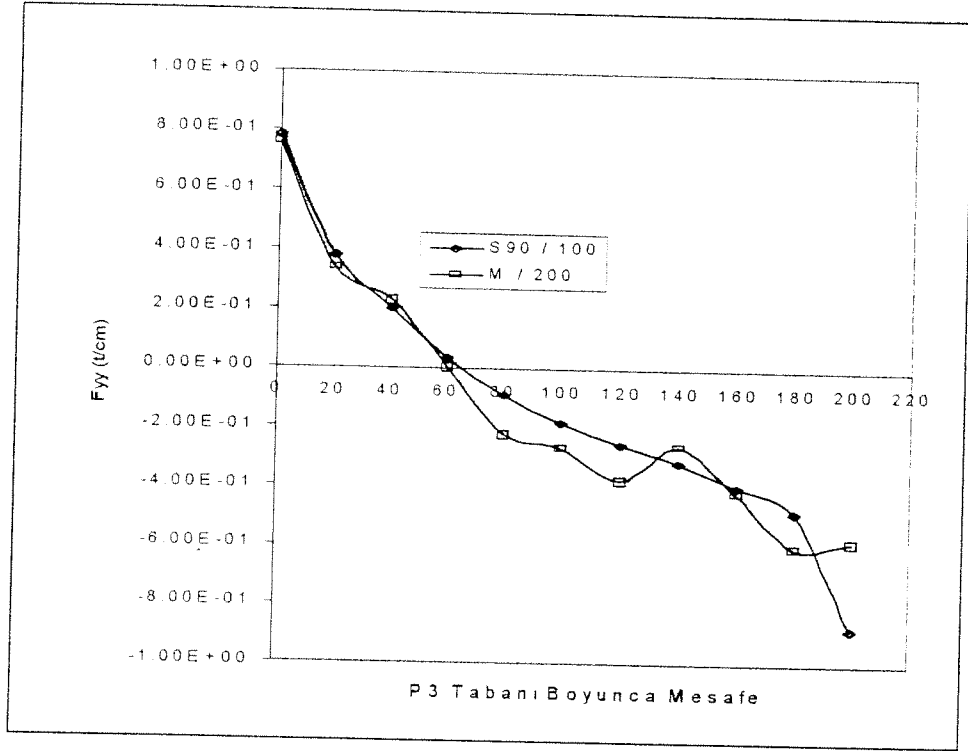
Şekil 2.5 Q yüklemesi altında 1-1 hattı boyunca oluşan yanal deplasmanlar.



Şekil 2.6 Q yüklemesi altında P3 tabanı boyunca oluşan aksel kuvvet dağılımı.



Şekil 2.7 T yüklemesi altında 1-1 hattı boyunca oluşan yanıl deplasmanlar.



Şekil 2.8 T yüklemesi altında P3 tabanı boyunca oluşan aksenal kuvvet dağılımı.

Sayısal test problemleri ile gerçekleştirilen incelemeler sonucunda panel makro elemanlar kullanılarak modellenen yapısal sistemlerde deplasman tepkisini yakalama yeteneğinin son derece yüksek olduğu görülmektedir. Bununla birlikte panel kenarları boyunca aksenal deplasman ve burulma dönmeleri açısından beklendiği gibi bir zaaf gözlenmektedir. Sorun makro elemanın formülasyonu aşamasında yapılan, aksenal deplasman ve burulma dönmelerinin panel kenarları boyunca doğrusal değişeceği, varsayımından kaynaklanmaktadır. Bu durum makro elemanın 4 köşe düğüm noktası yerine çevresi boyunca 8 veya daha fazla düğüm noktası ile tanımlanması durumunda büyük ölçüde giderilmektedir. Bu yüzden uygulamada mümkün olduğunca 8 veya daha fazla düğüm noktalı panel makro elemanları tercih edilmelidir. Panel makro elemanın yanıl yükler altındaki (eğilme) davranışının standart sonlu elemanlar modeli davranışına son derece yakın olduğu görülmektedir. Benzer şekilde makro elemanın kendi düzlemi



içindeki eğilme davranışı da standart sonlu elemanlar modeli davranışına oldukça yakındır. Eksênel deformasyon ve burulma dönmelerinin panel eleman kenarları boyunca doğrusal değişeceği kabulü ve sonuçta böyle bir değişime zorlanması elemanın kenarları boyunca bir şerit içinde oluşan kesme gerilmelerinde normal dışı bir değişime neden olmaktadır. Ancak bu türbülans kenarlardan içeriye doğru ilerledikçe hızla azalıp kaybolmaktadır.

## BÖLÜM 3

### XSPlate: SONLU ELEMANLAR TABANLI BİR PLAK İŞLEMCİSİ

#### 3.1 Giriş

Yapısal sistemlerin çözümlenmesi aşamasında karşılaşılan zorluklardan biri de düzgün olmayan bir geometriye sahip ve/veya düzensiz yüklemeye maruz plakların analizidir. Matematiksel olarak kesin olan analitik çözümlerin sadece dörtgen veya dairesel plak gibi düzgün geometrili ve birtakım özel yükleme durumları için var olduğunu düşünürsek bu tür sistemlerin çözümü için sayısal yöntemlere yönelmek kaçınılmaz olmaktadır. Sayısal yöntem olarak da sonlu elemanlar yöntemi en uygun alternatif olarak ortaya çıkmaktadır. Son yıllarda hızla gelişip hızlanan ve kolayca ulaşılabilir bir araç haline gelen bilgisayarlar sayesinde, geçmişte sadece özel bazı yapısal sistemler için başvurulan, sistemin sonlu elemanlar yöntemi kullanılarak modellenmesi ve ardından ayrıntılı analizleri yaklaşımı günlük hayatta her an kullanılabilir bir yöntem haline gelmiştir. Bu sayede sonlu elemanlar modeli hazırlanan her türlü yapısal sistemin çözümlenmesi mümkün olmaktadır. Ancak yapısal sistem analizlerinin sonlu elemanlar yöntemi kullanılarak gerçekleştirilmesi durumunda başlıca iki ciddi sorunla karşılaşılmaktadır. Bunlardan birincisi sonlu elemanlar modelinin hazırlanarak sistemin sayısal olarak analiz yazılımına tanıtılması aşamasında ikincisi de analiz sonrası üretilen yüksek hacimdeki sayısal sonuçların

değerlendirilmesi, yorumlanması ve özümlenmesi aşamalarında ortaya çıkmaktadır. \*

Sistemin sonlu elemanlar modelinin hazırlanması aşamasındaki en önemli güçlük yapının sayısallaştırılabilmesi için gerekli düğüm noktaları ve bu noktalar yardımı ile ortamın sonlu elemanlar ağının tanımlanması ardından da oluşturulan bu ağ ve düğüm noktaları bazında yüklerin tanımlanması olarak karşımıza çıkmaktadır. Analizcinin büyük bir enerjisini gerektiren bu aşama aynı zamanda maddi hata yapmaya da son derecede elverişli bir süreçtir.

Sonuçların değerlendirilmesi de genellikle bir dizi son işlemleri gerektirmektedir. Bu gereksinim yapısal sistemin boyutu ve karmaşıklığı ile, modelde kullanılan sonlu elemanlar ağının yoğunluğu ile ve kullanılan sonlu eleman yapısının karmaşıklığı ile hızla artmaktadır. Plak sonlu elemanları doğası gereği bu gereksinimi zorlayan bir yapıya sahiptir ve bu yüzden analizcinin enerjisinin büyük bir bölümü de bu aşamada harcanmaktadır.

Bu çalışmada temel amaç günlük uygulamalarda kolayca kullanılacak sonlu elemanlar tabanlı bir plak analiz sistemi geliştirmektir. Bu amaç doğrultusunda ilerlerken aşağıda özetlenen başlıca üç husus göz önünde bulundurulacaktır.

*Genellik:* Geliştirilecek sistem plak geometrisi, yükleme ve mesnetlenme şartları açısından uygulamada olası her durumu içerecek ve çözümlenecek bir yapıda olmalıdır.

*Esnelik:* Sistem kullanıcı mühendisten minimum girdi gerektirmeli ve kullanıcı mühendisi genellikle son derecede sıkıcı ve hata yapmaya yatkın olan sonlu elemanlar modeli veri dosyası hazırlama işleminden kurtarmalıdır.

*Etkinlik:* Sistem analiz sonucu üretilen yüksek hacimdeki sayısal çıktıları mühendisin yapısal sistem davranışını kolayca algılayabileceği ve tasarımda direkt olarak kullanabileceği formda sunarak yorumlama-değerlendirme enerjisini mümkün olan en az düzeye indirmelidir.

Ancak geliştirilecek sistem bu hedeflere ulaşırken hiçbir şekilde analizi gerçekleştiren mühendis adına bazı kararlar vererek kendisini analizci konumuna koymamalı ve mühendisin model üzerindeki kontrolünü sınırlamamalıdır.

### 3.2 XSPlate Plak Analiz Sisteminin Ana Hatları

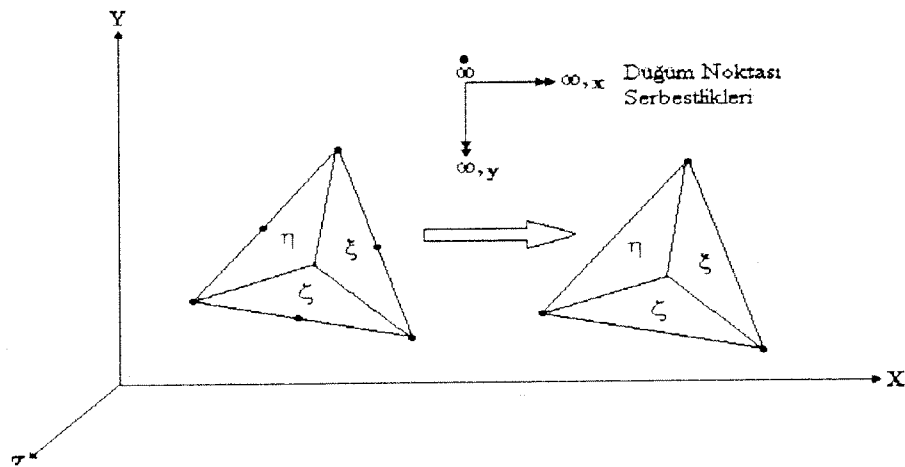
Belirlenen hedefler doğrultusunda oluşturulan XSPlate[22] plak analiz sisteminin ana hatları başlıklar halinde şöyle özetlenebilir:

- Sistem DKT (Discrete Kirchhoff Triangle) sonlu eleman tabanlıdır.
- Plak birçok boşluk içerebilir ancak gerek plak gerekse de olası boşlukların geometrisi son derece genel olup geometri konusunda bir kısıtlama yoktur.
- Plak mesnetlenme durumu plak kenarları boyunca tanımlanmış olup herhangi bir kenarı boyunca sabit (ankastre), mafsalsal veya serbest sınır şartına sahip olabilir.
- Sonlu elemanlar ağı otomatik olarak üretilir ancak oluşturulacak sonlu elemanlar ağı yoğunluğunu kullanıcı belirler.
- Plak üzerinde düzgün yayılı basınç yüklemesi, herhangi iki nokta arasında tanımlı şerit yük ve herhangi bir noktada tekil yük tanımlanabilir. Yüklerin tanımı kullanılan sonlu elemanlar ağından tamamen bağımsızdır.
- Sistem analiz sonrası plak tepkilerinin sorgulanabilmesi ve sonuçların değerlendirilerek tasarım için temel tepkilerin belirlenebilmesi için etkin sorgulama fonksiyonları içermektedir.

Sistem için temel eleman olarak DKT[11-15] plak eğilme sonlu elemanı seçilmiştir. Kirchhoff plak eğilme teorisine dayanan bu üçgen elemanın seçimindeki başlıca kriterler şöyle özetlenebilir.

- Eleman hesaplama açısından son derecede ucuz ve etkilidir.
- Uzun bir süredir kullanımda olan eleman zamanın testinden başarı ile çıkmıştır[12].
- Eleman üçgen geometriye sahiptir ve üçgen geometrili elemanlar düzgün olmayan geometrili plakların tanımlanmasında dörtgen elemanlara göre daha esnek bir yapıya sahiptirler.
- Düzlem içinde verilen N adet nokta ile oluşturulacak üçgenlerin belirlenmesi problemi matematikçiler tarafından da üzerinde çokça çalışılmış bir matematiksel problem olup bu konuda geliştirilmiş güçlü ve sağlam algoritmalar mevcuttur. Dolayısı ile plakların üçgen elemanlara bölünmesi işlemini etkin bir şekilde gerçekleştirmek mümkündür. Burada ilginç olan nokta matematiksel çözümün ürettiği üçgenlerin sonlu elemanlar yöntemi açısından da arzu edilen ve verilen N adet noktanın uzaydaki dağılımı göz önüne alındığında mümkün olan en uygun topolojik özelliklere sahip olduğu hususudur.

### 3.3 DKT Plak Eğilme Elemanı



Şekil 3.1 DKT plak eğilme elemanı ilk ve son topolojisi

Literatürde ilk kez 1969 yılında, [11], görülen DKT elemanı Kirchhoff plak eğilme teorisine dayanmakta olup sonlu eleman formülasyonunun başlangıç

noktası kenar orta noktalarında da düğüm noktaları içeren toplam 6 düğüm noktalı ve doğrusal kenarlı bir üçgendir (Şekil 3.1).

Düğüm noktası serbestlikleri olarak düzlem içi dönmeler ( $\theta_x$ ,  $\theta_y$ ) ile düzlem dışı ötelenme ( $\omega$ ) göz önüne alınmıştır. Kirchhoff teorisinin direkt sonucu olarak kayma deformasyonları ihmal edildiğinden dönmeler  $\theta_x$  ve  $\theta_y$  sırasıyla  $\omega_y$  ve  $\omega_x$  ile değiştirilebilir. Kenarlar boyunca deplasman fonksiyonu köşe noktaların deplasman ve eğimleri cinsinden kübik bir fonksiyon olarak ifade edilirse kenar orta noktaları deplasman ve normal doğrultudaki dönmelerini köşe noktaları deplasman ve dönmeleri cinsinden yazmak mümkündür. Elemanın her bir kenarı boyunca kenar doğrultusundaki dönmelerin de ilgili köşe noktalarındaki kenar boyu dönme değerleri arasında doğrusal olarak değiştiği varsayılırsa elemanın serbestliklerini tamamen köşe noktalarındaki 3 serbestliğe ( $\omega, \omega_x$  ve  $\omega_y$ ) indirmek mümkün olmaktadır. Bu eleman için ayrıntılı bir sonlu elemanlar formülasyonu Batoz[13] tarafından verilmektedir.

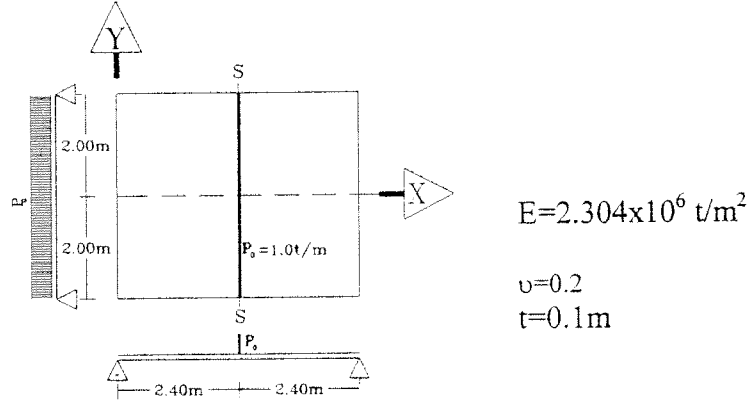
### 3.4 Sayısal Test Problemleri

**XSPlate** tarafından üretilen sonuçları test etmek için biri literatürde analitik çözümü de bulunan dikdörtgen plak diğeri de geometrisi oldukça düzensiz ve boşluklu bir plak seçilmiştir. İkinci plak için analitik çözüm bulunmadığından sonuçlar mühendislik dünyasında oldukça kabul görüp yaygın olarak kullanılan **SAP90**[21] sonlu elemanlar programı ile elde edilen sonuçlarla karşılaştırılmıştır.

#### 3.4.1 Duvar Yüküne Maruz Kenarları Boyunca Basit Mesnetli Dikdörtgen Plak

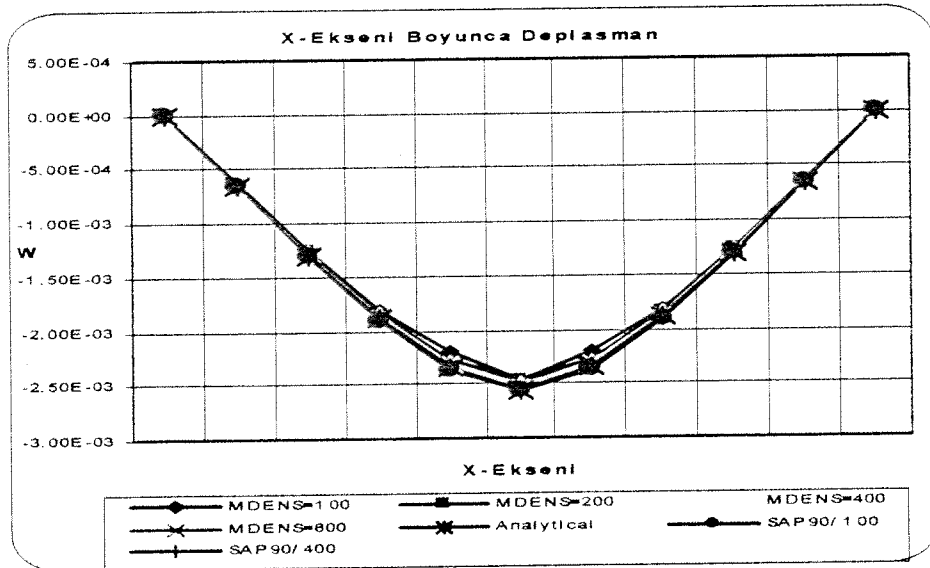
**XSPlate** plak analiz sistemi önce literatürde analitik çözümü de mevcut bir plak eğilme problemi ile test edilmiştir. İncelenen plak özellikleri ve yükleme Şekil 3.2 de gösterilmiştir. Dört kenarı boyunca basit mesnetli ve şerit yükleme

altındaki bu plak eğilme problemine ait analitik çözüm Szilard [23] tarafından verilmektedir. XSPlate kullanılarak elde edilen sonuçlar Şekil 3.3 de gösterilmekte ve sonuçlar analitik çözüm yanında SAP90[21] çözümüyle de karşılaştırılmaktadır.

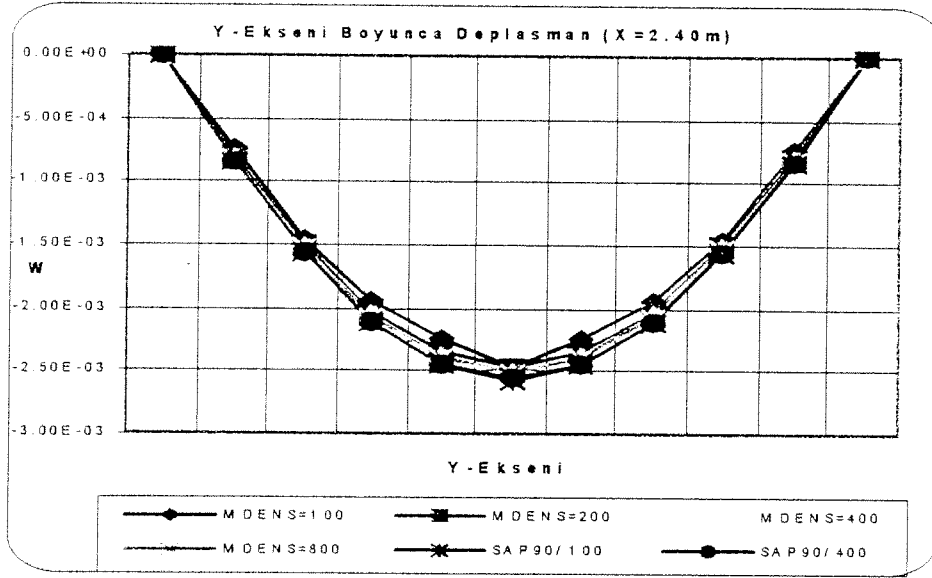


Şekil 3.2. Kenarları boyunca basit mesnetli ve duvar yüküne maruz dikdörtgen

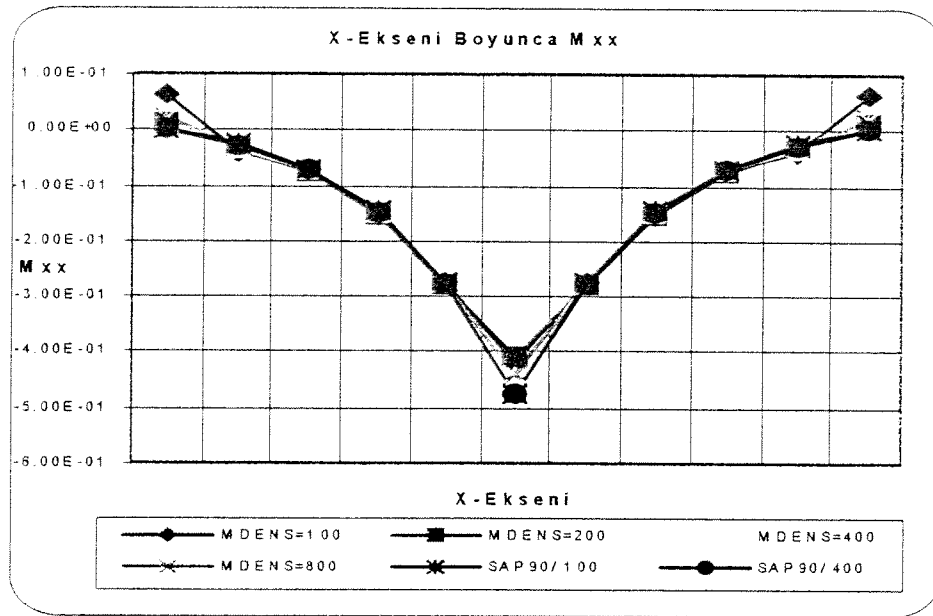
Dört kenarı boyunca basit mesnetli ve şerit yükleme altındaki bu plak eğilme problemine ait analitik çözüm Szilard [23] tarafından verilmektedir. XSPlate kullanılarak elde edilen sonuçlar Şekil 3.3-3.8 de gösterilmekte ve sonuçlar analitik çözüm yanında SAP90 [21] çözümüyle de karşılaştırılmaktadır.



Şekil 3.3 X-Eksenli boyunca oluşan plak deplasmanları (Y=0.0)

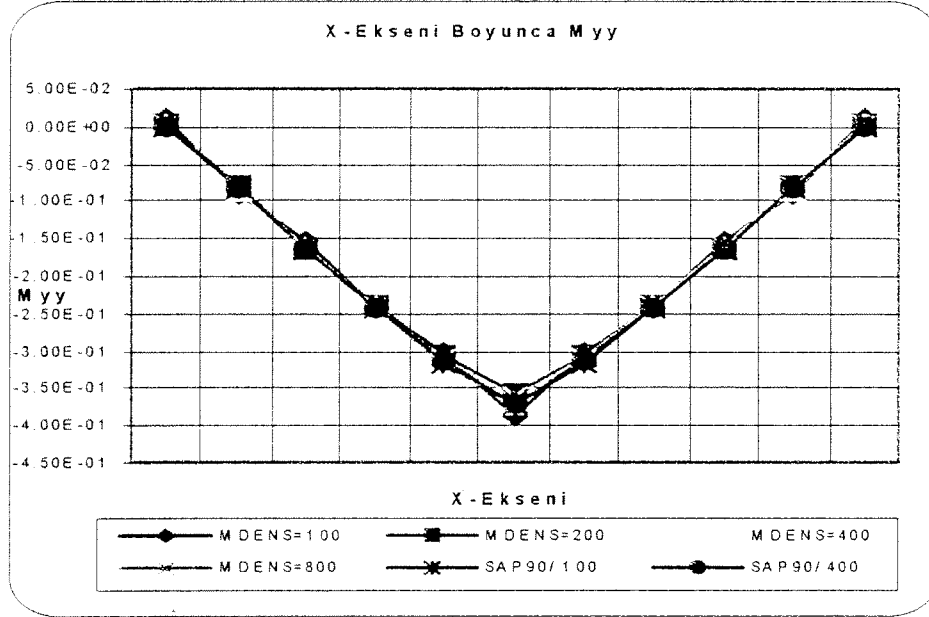


Şekil 3.4 Y-Ekseni boyunca oluşan plak deplasmanları (X=2.40)

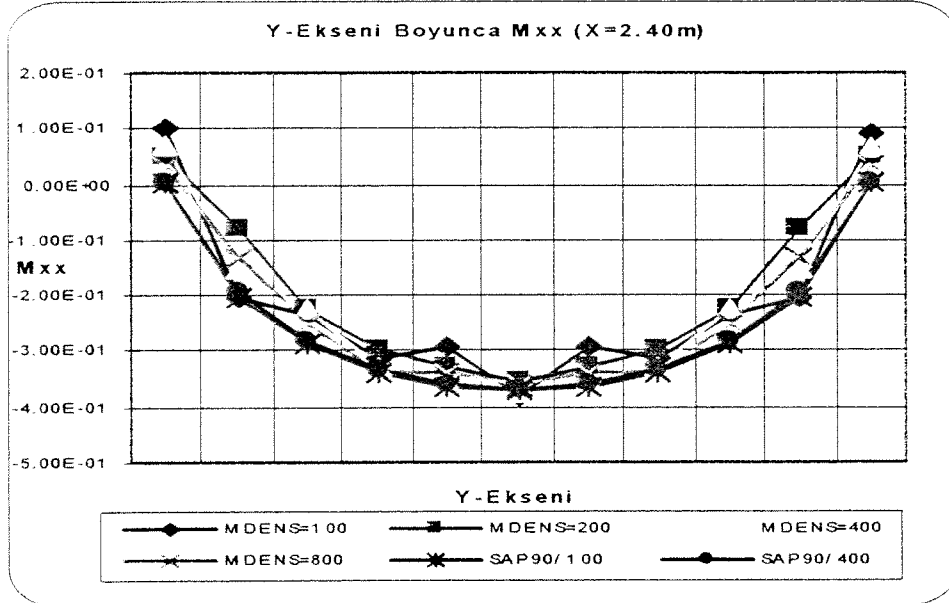


Şekil 3.5 X-Ekseni boyunca Mxx değişimi (Y=0.0)

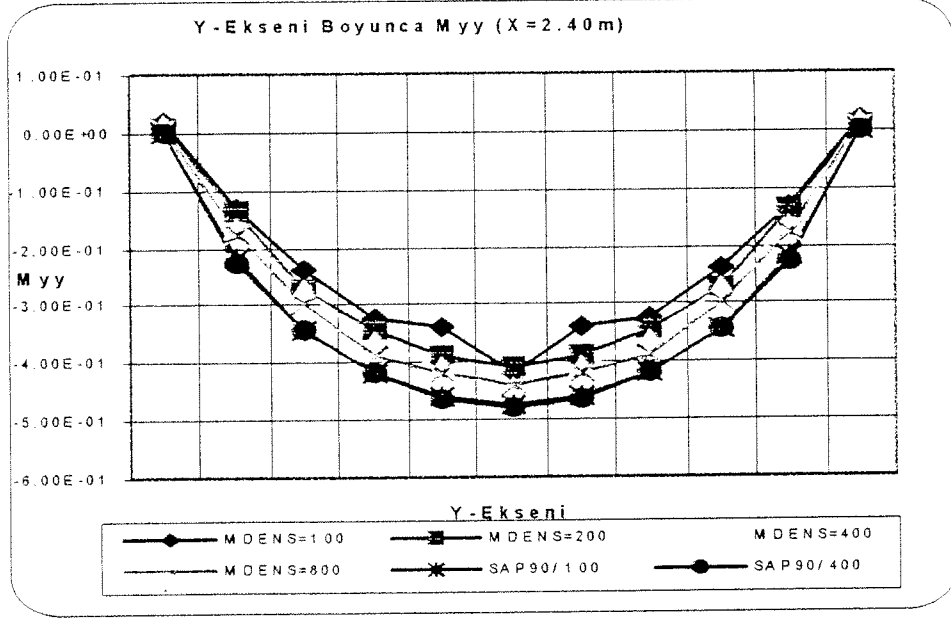




Şekil 3.6 X-Eksenli boyunca M<sub>yy</sub> değışimi (Y=0.0)



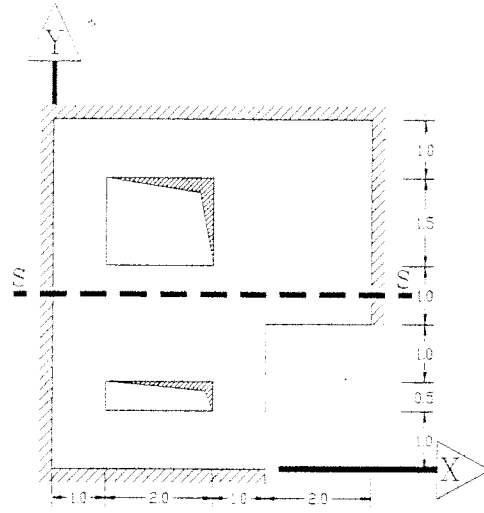
Şekil 3.7 Y-Eksenli boyunca M<sub>xx</sub> değışimi (X=2.40)



Şekil 3.8 Y-Ekseni boyunca M<sub>yy</sub> değişimi (X=2.40)

### 3.4.2 Geometrisi Düzensiz Boşluklu Döşeme

Test problemi olarak seçilen ikinci plak özellikleri Şekil 3.9 da görülmektedir. İki adet boşluk içeren 6 kenarlı bu döşeme dört komşu kenarı boyunca ankastre komşu diğer iki kenarı ise serbest sınır şartına sahiptir. Düzgün yayılı yük altındaki döşemeye ait sonuçlar SAP90[20] çözümü ile birlikte Şekil 3.10-3.12 de verilmektedir.

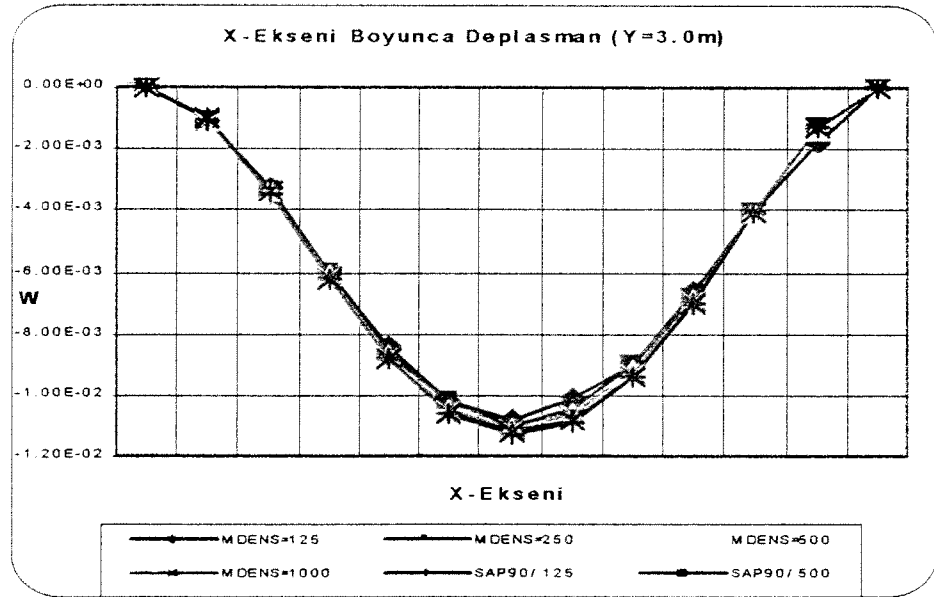


$$E=2.6 \times 10^6 \text{ t/m}^2$$

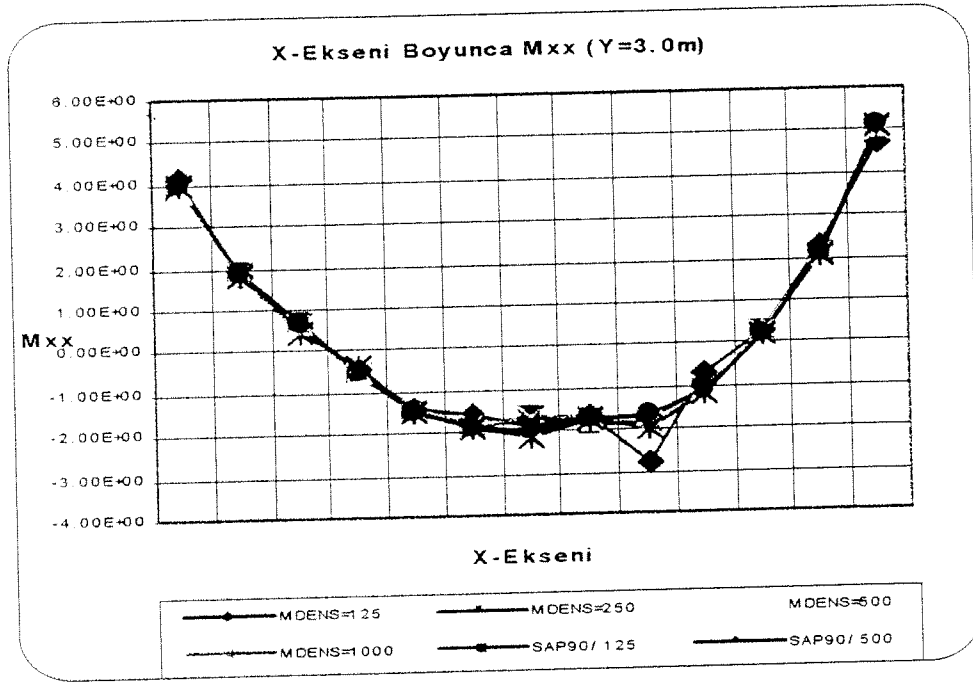
$$\nu=0.2$$

$$t=0.12$$

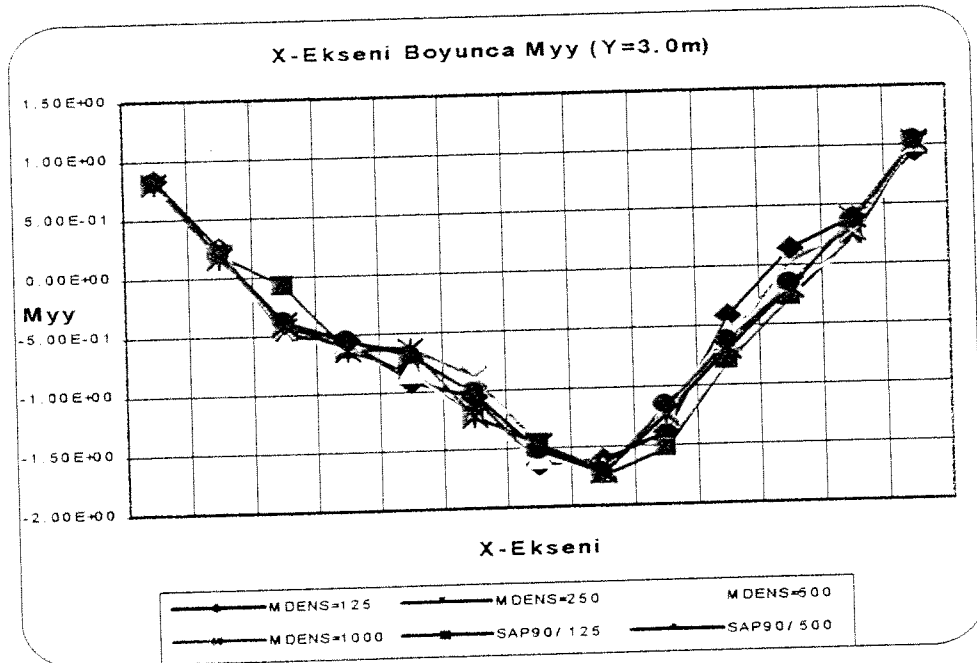
Şekil 3.9. Geometrisi düzensiz boşluklu döşeme



Şekil 3.10 X-Eksenli boyunca deplasman (Y=3.0m)



Şekil 3.11 X-Ekseni boyunca Mxx değişimi (Y=3.0m)



Şekil 3.12 X-Ekseni boyunca Myy değişimi (Y=3.0m)

**XSPlate** kullanımı ile gerçekleştirilen döşeme analizleri ve elde edilen sonuçlar ışığında aşağıdaki gözlemler yapılabilir.

- Geliştirilen sistem geometrik ve yükleme olarak düzensiz ince plak döşemelerin analizleri için son derece esnek ve etkili olan sonlu elemanlar tabanlı bir araçtır.
- Elde edilen sonuçlar gerek analitik gerekse de alternatif sonlu elemanlar tabanlı çözümler ile iyi bir uyum içindedir.
- Sistem uygulamada döşeme tasarımlarında kullanılan ancak düzgün sayılabilecek plak geometrileri için mevcut moment katsayıları ve döşeme yüklerinin çevre kirişlere aktarımında kullanılan akma hattı yaklaşımlarına sonlu elemanlar tabanlı bir alternatif oluşturmaktadır.
- Sistem mevcut yapısal analiz ve tasarım yazılımlarına kolaylıkla adapte edilebilecek bir yapıya sahiptir.

## KAYNAKLAR

1. G.R.Monforton, and T.S.Wu, “*Matrix Analysis of Semi-Rigidly Connected Steel Frames*”, ASCE Structural Division, Vol. 89, pp. 13-42, 1963.
2. Ö.Yüncü, A Finite Element Based Reinforced Concrete Interface Element, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1998
3. F.J.Vecchio, and M.P.Collins, “*Predicting the Response of Reinforced Concrete Beams Subjected to Shear Using Modified Compression Field Theory*”, ACI Structural Journal, Vol. 85, pp. 258-268, 1988.
4. W.Ritter, “*Die Bauweise Hennebique*”, Schweizerische Bauzeitung (Zürich), Feb. 1899.
5. E.Mörsch, Concrete Steel Construction (Der Eisenbetonbau), McGraw-Hill Book Co., New York, 1909, 368p.
6. U.Ersoy, “*Reinforced Concrete*”, Middle East Technical University, 1994.
7. F.J.Vecchio, and M.B.Emara, “*Shear Deformations in Reinforced Concrete Frames*”, ACI Structural Journal, Vol. 89, No. 1, pp. 46-56, 1992.
8. M.U.Polat, H.B.Bahat, A.H.Argeşo, “*Düzlem Kabuk Yapısal Sistemlerin Analizi İçin Bir Panel Makro Elemanı*”, III.Ulusal Hesaplmalı Mekanik Konferansı, İstanbul, pp.203-210, 16-18 Kasım 1998
9. A.H.Argeşo, A Panel Macro Element for the Analysis of Structural Wall Systems, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1996

10. **H.B.Bahat**, *Analysis of Structural Wall Systems*, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1998
11. **J.H.Stricklin**, **W.Haisler**, **P.Tisdale**, **K.Gunderson**, “ A Rapidly Converging Triangular Plate Element”, *JAIJA Journal*, 7, pp.180-181, 1969.
12. **J.L.Batoz**, **K.J.Bathe**, and **L.W.Ho**, “A Study of Three-Node Triangular Plate Bending Elements”, *Int. J. Num. Meth. Engng.*, 18, pp.1771-1812, 1980.
13. **L.J.Batoz**, “An Explicit Formulation for an Efficient Plate Bending Element”, *Int. J. Num. Meth. Engng.*, 21, pp.1077-1089, 1982.
14. **C.Jeyachandrabose**, **C.Kirkhope**, **C.R.Babu**, “An Alternative Explicit Formulation for the DKT Plate-Bending Element”, *Int. J. Num. Meth. Engng.*, 21, pp.1289-1293, 1985.
15. **R.D.Cook**, **S.D.Malkus**, and **M.E.Plesha**, *Concepts and Applications of Finite Element Analysis*, Wiley, 1989.
16. **D.J.Allman**, “A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Analysis”, *Computers and Structures*, 19, pp.1-8, 1984.
17. **D.F. Watson**, “Computing the N-Dimensional Delunay Triangulation with Application to Voronoi Polytopes”, *The Computer Journal*, 24, pp.167-172, 1981.
18. **B.Joe** and **R.B.Simpson**, “Triangular Meshes for Regions of Complicated Shape”, *Int. J. Num. Meth. Engng.*, 23, pp.751-768, 1986.
19. **S.W.Sloan**, “A Fast Algorithm for Constructing Delunay Triangulations in Plane”, *Adv. Eng. Software*, 9, pp.34-55, 1987.
20. **O.C.Zienkiewicz** and **R.L.Taylor**, *The Finite Element Method*, Volume II, McGraw-Hill, 1991.
21. **A.Habibullah**. and **E.L.Wilson**, “SAP90-A General Structural Analysis Program User’s Guide”, *Computers and Structures, Inc.*, Berkely, California, 1989.

22. M.U.Polat, Ö.Kurç, “XSPlate: Sonlu Elemanlar Tabanlı Bir Plak İşlemcisi”, III.Ulusal Hesaplamalı Mekanik Konferansı, İstanbul, pp.227-233, 16-18 Kasım 1998
23. R.Szilar, *Theory and Analysis of Plates*, Prentice-Hall, Englewood Cliffs, NJ, 1974.



## EK A

# BİRLEŞİMLERİ ELASTO-PLASTİK MAFSALLI ÇERÇEVE SİSTEMLERİN ANALİZİ PROGRAMI TANIMI VE LİSTESİ

Birleşimleri elasto-plastik mafsallı prefabrik çerçeve yapısal sistemlerin analizlerini gerçekleştirmek üzere Object Pascal programlama dili kullanılarak geliştirilen sonlu elemanlar yazılımı obje tabanlı olup bu objelerin tanımları ile veri dosyası yapısı bu bölümde özetlenmektedir.

### A.1 NODE Objesi

NODE objesi yapısal sistemin düğüm noktalarının davranışını ve sınır şartlarını tanımlamaktadır.

```
unit Node;  
  
interface  
  
uses Classes, SysUtils, Windows, Matrix;  
  
const  
    DOFPerNode = 6;  
  
type  
    Restraints = (rtX, rtY, rtZ, rtRX, rtRY, rtRZ);  
    Trestraint = set of Restraints;  
  
const  
    rsFree: Trestraint = [];  
    rsFixed: Trestraint = [rtX, rtY, rtZ, rtRX, rtRY, rtRZ];  
    rsFreeXY: Trestraint = [rtZ, rtRX, rtRY];  
  
type  
    Tnode = class  
    private  
        FX, FY, FZ: Double;  
        Use: Cardinal;  
        procedure SetX(NewX: Double);  
        procedure SetY(NewY: Double);  
        procedure SetZ(NewZ: Double);
```

```

protected
public
  No: Integer;
  DOF: array [1..DOFPerNode] of cardinal;
  Disp: array [1..DOFPerNode] of Double;
  Restraint: Trestraint;
  constructor Create(aNo: Cardinal; aX, aY, aZ: Double; Arestraint: Trestraint);
  property X: Double read FX write SetX;
  property Y: Double read FY write SetY;
  property Z: Double read FZ write SetZ;
end;

Tload = class
protected
  procedure GetLoadVector(LoadVector: TdoubleVector); virtual; abstract;
end;

type
  TnodalLoad = class(Tload)
private
  procedure GetLoadVector(LoadVector: TdoubleVector); override;
public
  Node: Tnode;
  FX, FY, FZ, MX, MY, MZ: Double;
  constructor Create(aNode: Tnode; aFX, aFY, aFZ, aMX, aMY, aMZ: Double);
end;

type
  TnodeList = class(TobjectList)
protected
public
  N dof: Cardinal;
  procedure AssignDOF;
  function CreateParams(aNo: Integer; aX, aY, aZ: Double; R: Trestraint): Tnode;
  procedure AssignDisplacements(UU: TdoubleVector);
end;

  TloadList = class(TobjectList)
public
  procedure GetLoadVector(LoadVector: TdoubleVector);
end;

type
  TslaveNode = class(Tnode)
private
  Fmaster: Tnode;
public
  constructor Create(aNo: Cardinal; aX, aY, aZ: Double; Arestraint: Trestraint;
Amaster: Tnode);
  property Master: Tnode read Fmaster;
end;

implementation

{ Tnode }
constructor Tnode.Create(aNo: Cardinal; aX, aY, aZ: Double; Arestraint: Trestraint);
begin
  inherited Create;
  No := aNo;
  X := aX;
  Y := aY;
  Z := aZ;
  Restraint := Arestraint;
end;

procedure Tnode.SetX(NewX: Double);
begin
  if NewX <> FX then
    begin
      FX := NewX;
    end;
end;

```

```

procedure Tnode.SetY(NewY: Double);
begin
  if NewY <> FY then
  begin
    FY := NewY;
  end;
end;

procedure Tnode.SetZ(NewZ: Double);
begin
  if NewZ <> FZ then
  begin
    FZ := NewZ;
  end;
end;

(TnodeList)
procedure TnodeList.AssignDOF;
{ Designed for 3D }
var
  I: Cardinal;
  R: Trestraint;
begin
  NDOF := 0;
  for I:=0 to Count-1 do
  with Tnode(Items[I]) do
  begin
    R := Restraint;
    if not(rtX in R) then begin
      Inc(NDOF);
      DOF[1] := NDOF;
    end;
    if not(rtY in R) then begin
      Inc(NDOF);
      DOF[2] := NDOF;
    end;
    if not(rtZ in R) then begin
      Inc(NDOF);
      DOF[3] := NDOF;
    end;
    if not(rtrX in R) then begin
      Inc(NDOF);
      DOF[4] := NDOF;
    end;
    if not(rtrY in R) then begin
      Inc(NDOF);
      DOF[5] := NDOF;
    end;
    if not(rtrZ in R) then begin
      Inc(NDOF);
      DOF[6] := NDOF;
    end;
  end;
end;

function TnodeList.CreateParams(aNo: Integer; aX, aY, aZ: Double; R: Trestraint):
Tnode;
var
  N: Tnode;
begin
  N := Tnode.Create(aNo, aX, aY, aZ, R);
  Add(N);
  Result := N;
end;

procedure TnodeList.AssignDisplacements(UU: TdoubleVector);
var IL, J: Cardinal;
begin
  for IL:=1 to Count do
  with Tnode(Items[IL-1]) do
  begin
    for J:=1 to DOFPerNode do
    begin
      if DOF[J]=0 then Disp[J]:=0

```

```

        else Disp[J] := UU[DOF[J]];
    end;
end;
end;

{ TnodalLoad }
constructor TnodalLoad.Create(aNode: Tnode; aFX, aFY, aFZ, aMX, aMY, aMZ: Double);
begin
    Node := aNode;
    FX := aFX;
    FY := aFY;
    FZ := aFZ;
    MX := aMX;
    MY := aMY;
    MZ := aMZ;
end;

procedure TnodalLoad.GetLoadVector(LoadVector: TdoubleVector);
var TheDof: Integer;
begin
    if Node=nil then raise Exception.Create('Load has no nodes. ');
    TheDof := Node.DOF[1];
    if TheDof > 0 then LoadVector[TheDof] := LoadVector[TheDof] + FX;
    TheDof := Node.DOF[2];
    if TheDof > 0 then LoadVector[TheDof] := LoadVector[TheDof] + FY;
    TheDof := Node.DOF[3];
    if TheDof > 0 then LoadVector[TheDof] := LoadVector[TheDof] + FZ;
    TheDof := Node.DOF[4];
    if TheDof > 0 then LoadVector[TheDof] := LoadVector[TheDof] + MX;
    TheDof := Node.DOF[5];
    if TheDof > 0 then LoadVector[TheDof] := LoadVector[TheDof] + MY;
    TheDof := Node.DOF[6];
    if TheDof > 0 then LoadVector[TheDof] := LoadVector[TheDof] + MZ;
end;

{ TloadList }
procedure TloadList.GetLoadVector(LoadVector: TdoubleVector);
var IL: Cardinal;
begin
    for IL:=1 to Count do Tload[Items[IL-1]].GetLoadVector(LoadVector);
end;

{ TslaveNode }
constructor TslaveNode.Create(aNo: Cardinal; aX, aY, aZ: Double; Arestraint:
Treststraint; Amaster: Tnode);
begin
    inherited Create(aNo, aX, aY, aZ, Arestraint);
    Fmaster := Amaster;
end;

end.

```

## A.2 FRAME Objesi

FRAME objesi Element objesinin bir türevi olup düzlem çerçeve yapısal elemanın davranışını tanımlamaktadır.

```

unit Frame;

interface

uses
    Classes, Element, Matrix, Node, SysUtils;

type
    Tprismatic3D = class(Telement)
    private
        Flength: Double;
        function GetStrain: Double;
    end;

```

```

function GetLength: Double;
function GetCurvature: Double;
protected
  procedure GetRotation(RotMatrix: TdoubleMatrix); override;
  procedure GetStress(TotalStress: TdoubleVector); override;
public
  ElemStress: TdoubleVector;
  Alpha: Double;
  constructor Create(aNo: Integer; Jt1, Jt2: Tnode; Mat: Tmaterial; Sec: Tsection;
Rot: Double);
  destructor Destroy; override;
  property Strain: Double read GetStrain;
  property Length: Double read Flength;
  property Curvature: Double read GetCurvature;
end;

Tframe3D = class(Tprismatic3D)
protected
  procedure GetStiffness(ElemStif: TdoubleMatrix); override;
public
  procedure WriteProperties(Output: Tstrings); override;
  procedure WriteStress(Output: Tstrings); override;
end;

Ttruss3D = class(Tprismatic3D)
protected
  Yielded: Boolean;
  procedure GetStiffness(ElemStif: TdoubleMatrix); override;
  procedure GetStress(TotalStress: TdoubleVector); override;
public
  constructor Create(aNo: Integer; Jt1, Jt2: Tnode; Mat: Tmaterial; Sec: Tsection;
Rot: Double);
  procedure WriteProperties(Output: Tstrings); override;
  procedure WriteStress(Output: Tstrings); override;
end;

const
  MaxSteelLayers = 4;

type
  TconnectionInfo = record
    Height, Width: double;
    Offset: array[1..2] of double;
    Nlayers, Nsteel: Integer;
    A: array[1..4] of double;
    Concrete, Steel: Tmaterial;
    StLoc, StArea: array[1..MaxSteelLayers] of Double;
    Node1, Node2: Tnode;
  end;

  Tconnection = class(Tprismatic3D)
protected
  Info: TconnectionInfo;
  FelementList: TelementList;
  FsectionList: TsectionList;
  FnodeList: TnodeList;
  IntDOF: Integer;
  procedure GetStiffness(ElemStif: TdoubleMatrix); override;
  procedure GetStress(TotalStress: TdoubleVector); override;
  procedure GetTransformation(Transformation: TdoubleMatrix);
public
  constructor Create(aNo: Integer; AconnInfo: TconnectionInfo);
  destructor Destroy; override;
  procedure WriteProperties(Output: Tstrings); override;
  procedure WriteStress(Output: Tstrings); override;
end;

implementation

{ Tprismatic3D }
constructor Tprismatic3D.Create(aNo: Integer; Jt1, Jt2: Tnode; Mat: Tmaterial; Sec:
Tsection; Rot: Double);
begin
  inherited Create(aNo, Mat, Sec);

```

```

    Nnodes := 2;
    Nodes[1] := Jt1;
    Nodes[2] := Jt2;
    Alpha := Rot;
    ElemStress := TdoubleVector.Create(12);
    Flength := GetLength;
end;

destructor Tprismatic3D.Destroy;
begin
    ElemStress.Free;
    inherited Destroy;
end;

function Tprismatic3D.GetStrain: Double;
var
    XL, YL, ZL, L, DL: Double;
begin
    L := Length;
    XL := (Nodes[2].X + Nodes[2].Disp[1]) - (Nodes[1].X + Nodes[1].Disp[1]);
    YL := (Nodes[2].Y + Nodes[2].Disp[2]) - (Nodes[1].Y + Nodes[1].Disp[2]);
    ZL := (Nodes[2].Z + Nodes[2].Disp[3]) - (Nodes[1].Z + Nodes[1].Disp[3]);
    DL := Sqrt(Sqr(XL) + Sqr(YL) + Sqr(ZL)) - L;
    Result := DL / L;
end;

function Tprismatic3D.GetLength: Double;
var
    XL, YL, ZL: Double;
begin
    XL := Nodes[2].X - Nodes[1].X;
    YL := Nodes[2].Y - Nodes[1].Y;
    ZL := Nodes[2].Z - Nodes[1].Z;
    Result := Sqrt(Sqr(XL) + Sqr(YL) + Sqr(ZL));
end;

function Tprismatic3D.GetCurvature: Double;
begin
    Result := (Nodes[2].Disp[6] - Nodes[1].Disp[6]) / Length;
end;

procedure Tprismatic3D.GetRotation(RotMatrix: TdoubleMatrix);
var
    I, J, K: Integer;
    CX, CY, CZ, CXZ, XL, YL, ZL, L: Double;
begin
    XL := Nodes[2].X - Nodes[1].X;
    YL := Nodes[2].Y - Nodes[1].Y;
    ZL := Nodes[2].Z - Nodes[1].Z;
    L := Sqrt(Sqr(XL) + Sqr(YL) + Sqr(ZL));
    CX := XL/L;
    CY := YL/L;
    CZ := ZL/L;
    CXZ := Sqrt(CX*CX + CZ*CZ);
    RotMatrix.Clear;
    if CXZ < 0.001 then
    begin
        RotMatrix[1,2] := CY;
        RotMatrix[2,1] := -CY;
        RotMatrix[3,3] := 1.0;
        if Alpha > 0 then
        begin
            RotMatrix[2,1] := -CY * Cos(Alpha);
            RotMatrix[2,3] := Sin(Alpha);
            RotMatrix[3,1] := -CY * Sin(Alpha);
            RotMatrix[3,3] := Cos(Alpha);
        end;
    end else begin
        RotMatrix[1,1] := CX;
        RotMatrix[1,2] := CY;
        RotMatrix[1,3] := CZ;
        RotMatrix[2,1] := -CX*CY/CXZ;
        RotMatrix[2,2] := CXZ;
        RotMatrix[2,3] := -CY*CZ/CXZ;
    end;
end;

```

```

RotMatrix[3,1]:= -CZ/CXZ;
RotMatrix[3,3]:= CX/CXZ;
if Alpha > 0 then
begin
  RotMatrix[2,1]:= (-CX*CY*Cos(Alpha)-CZ*Sin(Alpha))/CXZ;
  RotMatrix[2,2]:= CXZ * Cos(Alpha);
  RotMatrix[2,3]:= (-CY*CZ*Cos(Alpha)+CX*Sin(Alpha))/CXZ;
  RotMatrix[3,1]:= (CX*CY*Sin(Alpha)-CZ*Cos(Alpha))/CXZ;
  RotMatrix[3,2]:= -CXZ * Sin(Alpha);
  RotMatrix[3,3]:= (CY*CZ*Sin(Alpha)+CX*Cos(Alpha))/CXZ;
end;
end;
for K := 1 to 3 do
  for I := 1 to 3 do
    for J := 1 to 3 do
      RotMatrix[3*K+I, 3*K+J] := RotMatrix[I, J];
    end;
  end;
end;

procedure Tprismatic3D.GetStress(TotalStress: TdoubleVector);
var
  ElemStif, RotMatrix: TdoubleMatrix;
  GelemStress: TdoubleVector;
  GelemDisp, ElemDisp: TdoubleVector;
  Indices: TintegerVector;
  TheDOF, I: Integer;
begin
  ElemStif := TdoubleMatrix.Create(12, 12);
  RotMatrix := TdoubleMatrix.Create(12, 12);
  GelemDisp := TdoubleVector.Create(12);
  ElemDisp := TdoubleVector.Create(12);
  GelemStress := TdoubleVector.Create(12);
  Indices := TintegerVector.Create(12);
  try
    GetStiffness(ElemStif);
    GetRotation(RotMatrix);
    for I:=1 to 6 do
      begin
        GelemDisp[I] := Nodes[1].Disp[I];
        GelemDisp[I + 6] := Nodes[2].Disp[I];
      end;
    MultiplyMatrixVector(RotMatrix, GelemDisp, ElemDisp);
    MultiplyMatrixVector(ElemStif, ElemDisp, GelemStress);
    RotMatrix.Transpose;
    MultiplyMatrixVector(RotMatrix, GelemStress, TotalStress);
    GetIndices(Indices);
    for I:=1 to 12 do
      begin
        TheDOF := Indices[I];
        if TheDOF>0 then TotalStress[TheDOF] := TotalStress[TheDOF] + GelemStress[I];
      end;
    finally
      Indices.Free;
      GelemStress.Free;
      ElemDisp.Free;
      GelemDisp.Free;
      RotMatrix.Free;
      ElemStif.Free;
    end;
  end;
end;

{ Tframe3D }
procedure Tframe3D.GetStiffness(ElemStif: TdoubleMatrix);
var
  L: Double;
  SCM1A, SCM1B, SCM2Y, SCM3Y, SCM4Y, SCM2Z, SCM3Z, SCM4Z: Double;
  I, J: Integer;
begin
  L := Length;
  if L<=0 then raise Exception.CreateFmt('Element %d has negative length!', [No]);
  ElemStif.Clear;
  SCM1A := Material.GetE(0.0) * Section.Area / L;
  SCM1B := Material.G * Section.Ixx / L;
  SCM2Y := 4 * Material.GetE(0.0) * Section.Iyy / L;
  SCM3Y := 1.5 * SCM2Y / L;

```

```

SCM4Y := 2 * SCM3Y / L;
SCM2Z := 4 * Material.GetE(0.0) * Section.Izz / L;
SCM3Z := 1.5 * SCM2Z / L;
SCM4Z := 2 * SCM3Z / L;
ElemStif[1,1] := SCM1A;
ElemStif[1,7] := -SCM1A;
ElemStif[2,2] := SCM4Z;
ElemStif[2,6] := SCM3Z;
ElemStif[2,8] := -SCM4Z;
ElemStif[2,12] := SCM3Z;
ElemStif[3,3] := SCM4Y;
ElemStif[3,5] := -SCM3Y;
ElemStif[3,9] := -SCM4Y;
ElemStif[3,11] := -SCM3Y;
ElemStif[4,4] := SCM1B;
ElemStif[4,10] := -SCM1B;
ElemStif[5,5] := SCM2Y;
ElemStif[5,9] := SCM3Y;
ElemStif[5,11] := SCM2Y * 0.5;
ElemStif[6,6] := SCM2Z;
ElemStif[6,8] := -SCM3Z;
ElemStif[6,12] := SCM2Z * 0.5;
ElemStif[7,7] := SCM1A;
ElemStif[8,8] := SCM4Z;
ElemStif[8,12] := -SCM3Z;
ElemStif[9,9] := SCM4Y;
ElemStif[9,11] := SCM3Y;
ElemStif[10,10] := SCM1B;
ElemStif[11,11] := SCM1Y;
ElemStif[12,12] := SCM2Z;
for I:=1 to 11 do
  for J:=I+1 to 12 do
    ElemStif[J,I] := ElemStif[I,J];
end;

procedure Tframe3D.WriteProperties(Output: Tstrings);
begin
  Output.Add(Format('%7d FRAME %7d%8d %-10s %-10s%9.2f',[No, Nodes[1].No,
Nodes[2].No, Section.Name, Material.Name, Alpha]));
end;

procedure Tframe3D.WriteStress(Output: Tstrings);
begin
  Output.Add(Format('%4d%13.5e%13.5e%13.5e%13.5e%13.5e%13.5e',
  {No, ElemStress[1], ElemStress[2], ElemStress[3], ElemStress[4],
  ElemStress[5], ElemStress[6]}));
  Output.Add(Format('%13.5e%13.5e%13.5e%13.5e%13.5e%13.5e',
  {ElemStress[7], ElemStress[8], ElemStress[9], ElemStress[10],
  ElemStress[11], ElemStress[12]}));
end;

```

### A.3 ELEMENT Objesi

ELEMENT objesi her türlü sonlu elemanın soyut temel sınıfıdır.

```

unit Element;

interface

uses
  Node, Matrix, Classes, SysUtils;

const
  MaxNodePerElem = 4;

type
  TmaterialInfo = record
    Name: String;
    Npoints: Cardinal;
    Strain, Stress: array [1..10] of Double;

```



```

    G, YieldStrain: Double;
end;

Tmaterial = class
private
    Info: TmaterialInfo;
public
    No, Count: Integer;
    constructor Create(AInfo: TmaterialInfo);
    function GetE(Astrain: Double): Double; virtual;
    function GetStress(Astrain: Double; var Failed: Boolean): Double; virtual;
    property G: Double read Info.G;
    property Name: String read Info.Name;
end;

Tsection = class
public
    Name: String;
    Area, Ixx, Iyy, Izz: Double;
    No, Count: Integer;
    constructor Create(aArea, aIxx, aIyy, aIzz: Double; aName: String);
end;

Telement = class
private
    Fmaterial: Tmaterial;
    Fsection: Tsection;
    procedure SetMaterial(NewMaterial: Tmaterial);
    procedure SetSection(NewSection: Tsection);
    function GetBandwidth: Integer;
protected
    Nnodes: Byte;
    procedure GetIndices(Index: TintegerVector); virtual;
    procedure GetStiffness(ESTIF: TdoubleMatrix); virtual; abstract;
    procedure GetRotation(RotMatrix: TdoubleMatrix); virtual; abstract;
    procedure GetStress(TotalStress: TdoubleVector); virtual; abstract;
public
    No: Integer;
    Nodes: array [1..MaxNodePerElem] of Tnode;
    constructor Create(aNo: Integer; Mat: Tmaterial; Sec: Tsection);
    destructor Destroy; override;
    procedure WriteProperties(Output: Tstrings); virtual; abstract;
    procedure WriteStress(Output: Tstrings); virtual; abstract;
    property Material: Tmaterial read Fmaterial write SetMaterial;
    property Section: Tsection read Fsection write SetSection;
    property BandWidth: Integer read GetBandWidth;
end;

type
TelementList = class(TobjectList)
private
    function GetBandwidth: Integer;
protected
    procedure AddStiffness(Global, Stiffness: TdoubleMatrix; Indices: TintegerVector);
public
    procedure GetGlobalStiffness(Global: TdoubleMatrix);
    procedure CalculateStresses(TotalStress: TdoubleVector);
    property BandWidth: Integer read GetBandWidth;
end;

type
TsectionList = class(TobjectList);
TmaterialList = class(TobjectList);

type
Tconcrete = class(Tmaterial)
private
    FC, Fct, Sco, Scu, Ec: Double;
public
    constructor Create(afc: Double);
    function GetE(Astrain: Double): Double; override;
    function GetStress(Astrain: Double; var Failed: Boolean): Double; override;
end;

```

```

implementation

{ Tsection }
constructor Tsection.Create(aArea, aIxx, aIyy, aIzz: Double; aName: String);
begin
    inherited Create;
    Count := 0;
    Area := aArea;
    Ixx := aIxx;
    Iyy := aIyy;
    Izz := aIzz;
    Name := aName;
end;

{ Tmaterial }
constructor Tmaterial.Create(Ainfo: TmaterialInfo);
begin
    inherited Create;
    Count := 0;
    Info := Ainfo;
end;

function Tmaterial.GetE(Astrain: Double): Double;
var
    I: Integer;
begin
    Astrain := Abs(Astrain);
    with Info do begin
        I := Npoints;
        if Astrain > Strain[I] then
            Result := (Stress[I] - Stress[I - 1]) / (Strain[I] - Strain[I - 1])
        else begin
            while Astrain < Strain[I] do Dec(I);
            Result := (Stress[I+1] - Stress[I]) / (Strain[I+1] - Strain[I])
        end;
    end;
end;

function Tmaterial.GetStress(Astrain: Double; var Failed: Boolean): Double;
var
    I: Integer;
begin
    Result := 0;
    I := 1;
    Failed := False;
    if Astrain > Info.Strain[Info.Npoints] then Failed := True
    else
        repeat
            if Astrain > Info.Strain[I+1] then
                begin
                    Result := Result + GetE(Info.Strain[I]) * Info.Strain[I+1];
                    Astrain := Astrain - Info.Strain[I+1];
                end
            else
                begin
                    Result := Result + GetE(Info.Strain[I]) * Astrain;
                    Astrain := -I;
                end;
            Inc(I);
        until Astrain <= 0;
end;

{ Telement }
constructor Telement.Create(aNo: Integer; Mat: Tmaterial; Sec: Tsection);
begin
    inherited Create;
    Material := Mat;
    Section := Sec;
    No := aNo;
end;

destructor Telement.Destroy;
begin
    if Assigned(Fmaterial) then Fmaterial.Count := Fmaterial.Count - 1;
end;

```

```

    if Assigned(Fsection) then Fsection.Count := Fsection.Count - 1;
    inherited Destroy;
end;

procedure Telement.SetMaterial(NewMaterial: Tmaterial);
begin
    if Assigned(Fmaterial) then Fmaterial.Count := Fmaterial.Count-1;
    Fmaterial := NewMaterial;
    if Assigned(Fmaterial) then Fmaterial.Count := Fmaterial.Count+1;
end;

procedure Telement.SetSection(NewSection: Tsection);
begin
    if Assigned(Fsection) then Fsection.Count := Fsection.Count-1;
    Fsection := NewSection;
    if Assigned(Fsection) then Fsection.Count := Fsection.Count+1;
end;

procedure Telement.GetIndices(Index: TintegerVector);
var I, J: Integer;
begin
    for I := 1 to Nnodes do
        with Nodes[I] as Tnode do
            for J := 1 to DOFPerNode do
                Index[(I-1)*DOFPerNode + J] := DOF[J];
            end;
        end;
end;

function Telement.GetBandWidth: Integer;
var
    I, J, Max, Min: Integer;
begin
    Max := 0;
    Min := High(Integer);
    for I := 1 to Nnodes do
        with Tnode(Nodes[I]) do
            for J := 1 to DOFPerNode do
                if DOF[J]>0 then
                    begin
                        if DOF[J]>Max then Max := DOF[J];
                        if DOF[J]<Min then Min := DOF[J];
                    end;
                end;
            end;
        end;
    if Min>Max then Min := Max;
    Result := Max - Min + 1;
end;

{ TelementList }
procedure TelementList.AddStiffness(Global, Stiffness: TdoubleMatrix; Indices:
TintegerVector);
var I, J, IG, JG: Integer;
begin
    for I:=1 to Stiffness.Rows do
        begin
            IG := Indices[I];
            if IG>0 then
                for J:=1 to Stiffness.Cols do
                    begin
                        JG := Indices[J] - IG + 1;
                        if JG>0 then Global[IG,JG]:=Global[IG,JG]+Stiffness[I,J];
                    end;
                end;
            end;
        end;
end;

function TelementList.GetBandWidth: Integer;
var IL, BW: Integer;
begin
    Result := 0;
    for IL:=1 to Count do
        begin
            BW := Telement(Items[IL-1]).Bandwidth;
            if BW>Result then Result := BW;
        end;
    end;
end;

procedure TelementList.GetGlobalStiffness(Global: TdoubleMatrix);

```

```

var
  IL: Integer;
  ElemStif, RotMatrix, TempMatrix: TdoubleMatrix;
  Indices: TintegerVector;
begin
  ElemStif := TdoubleMatrix.Create(12, 12);
  RotMatrix := TdoubleMatrix.Create(12, 12);
  TempMatrix := TdoubleMatrix.Create(12, 12);
  Indices := TintegerVector.Create(12);
  try
    for IL:=0 to Count-1 do
      with Telement(Items[IL]) do
        begin
          GetStiffness(ElemStif);
          GetRotation(RotMatrix);
          GetIndices(Indices);
          MultiplyTranspose(RotMatrix, ElemStif, TempMatrix);
          MultiplyMatrix(TempMatrix, RotMatrix, ElemStif);
          AddStiffness(Global, ElemStif, Indices);
        end;
      finally
        Indices.Free;
        ElemStif.Free;
        RotMatrix.Free;
        TempMatrix.Free;
      end;
    end;
end;

procedure TelementList.CalculateStresses(TotalStress: TdoubleVector);
var IL: Integer;
begin
  for IL:=1 to Count do
    Telement(Items[IL-1]).GetStress(TotalStress);
  end;

  ( Tconcrete )
  constructor Tconcrete.Create(Afc: Double);
  begin
    Fc := Afc;
    Scu := 0.0038;
    Ec := 12680 + 460 * Fc;
    Sco := 2 * Fc / Ec;
    Fct := 0.35 * Sqrt(Fc);
    Info.Name := Format('BS%.0f', [Fc]);
    Info.G := 8000000.0;
  end;

  function Tconcrete.GetStress(Astrain: Double; var Failed: Boolean): Double;
  begin
    Result := 0.0;
    Failed := False;
    Astrain := -Astrain;
    if Astrain < -0.0002 then Failed := True
    else if Astrain < -0.0001 then Result := -3000*Fct * (Astrain + 0.0001) - Fct
    else if Astrain < 0.0 then Result := 10000*Fct * Astrain
    else if Astrain < Sco then Result := Fc*(2*Astrain/Sco - Sqr(Astrain/Sco))
    else if Astrain < Scu then Result := (0.15*Fc)/(Scu-Sco) * (Sco - Astrain) + Fc
    else Failed := True;
    Result := -Result*1000;
  end;

  function Tconcrete.GetE(Astrain: Double): Double;
  begin
    Result := (12680 + 460*Fc) * 1000;
  end;
end.

```

## A.4 TRUSS Objesi

TRUSS objesi ELEMENT objesinin bir türevi olup düzlem çubuk elemanların davranışını tanımlamaktadır.

```
{ Ttruss3D }
constructor Ttruss3D.Create(aNo: Integer; Jt1, Jt2: Tnode; Mat: Tmaterial; Sec:
Tsection; Rot: Double);
begin
  inherited Create(aNo, Jt1, Jt2, Mat, Sec, Rot);
  Yielded := False;
end;

procedure Ttruss3D.GetStiffness(ElemStif: TdoubleMatrix);
var
  L: Double;
  SCMIa: Double;
  I, J: Integer;
begin
  ElemStif.Clear;
  if Not Yielded then begin
    L := Length;
    if L<=0 then raise Exception.CreateFmt('Element %d has negative length!', (No));
    SCMIa := Material.GetE(Strain) * Section.Area / L;
    ElemStif[1,1] := SCMIa;
    ElemStif[1,7] := -SCMIa;
    ElemStif[7,7] := SCMIa;
    for I:=1 to 11 do
      for J:=I+1 to 12 do
        ElemStif[J,I] := ElemStif[I,J];
      end;
    end;
  end;
end;

procedure Ttruss3D.GetStress(TotalStress: TdoubleVector);
var
  TheStress: Double;
  Failed: Boolean;
  RotMatrix: TdoubleMatrix;
  GelemStress: TdoubleVector;
  Indices: TintegerVector;
  TheDOF, I: Cardinal;
begin
  TheStress := Material.GetStress(Strain, Failed);
  if Failed then
    begin
      Yielded := True;
      TheStress := 0;
    end
  else TheStress := TheStress * Section.Area;
  ElemStress[1] := -TheStress;
  ElemStress[7] := TheStress;
  RotMatrix := TdoubleMatrix.Create(12, 12);
  Indices := TintegerVector.Create(12);
  GelemStress := TdoubleVector.Create(12);
  try
    GetRotation(RotMatrix);
    RotMatrix.Transpose;
    MultiplyMatrixVector(RotMatrix, ElemStress, GelemStress);
    GetIndices(Indices);
    for I:=1 to 12 do
      begin
        TheDOF := Indices[I];
        if TheDOF>0 then TotalStress[TheDOF] := TotalStress[TheDOF] + GelemStress[I];
      end;
    end;
  finally
    GelemStress.Free;
    Indices.Free;
    RotMatrix.Free;
  end;
end;
end;
```

```

procedure Ttruss3D.WriteProperties(Output: Tstrings);
begin
  Output.Add(Format('%7d TRUSS%8d%8d %-12s%-10s%9.2f', [No, Nodes[1].No, Nodes[2].No,
Section.Name, Material.Name, Alpha]));
end;

procedure Ttruss3D.WriteStress(Output: Tstrings);
var Failed: Boolean;
begin
  if Yielded then Output.Add(Format('%4d YIELDED', [No]))
  else
  begin
    Output.Add(Format('%4d%13.5e%13.5e%13.5e%13.5e%13.5e%13.5e',
      [No, ElemStress[1], ElemStress[2], ElemStress[3], ElemStress[4],
      ElemStress[5], ElemStress[6]]));
    Output.Add(Format('%13.5e%13.5e%13.5e%13.5e%13.5e%13.5e',
      [ElemStress[7], ElemStress[8], ElemStress[9], ElemStress[10],
      ElemStress[11], ElemStress[12]]));
    Output.Add(Format('%4d %15.4e%15.4e%15.4e',
      [No, Material.GetStress(Strain, Failed), Strain, Material.GetE(Strain)]));
  end;
end;

end.

```

## A.5 CONNECTION Objesi

CONNECTION objesi de ELEMENT objesinin bir türevi olup elasto-plastik yarı-rijit bağlantı modeli (arayüz elemanı) davranışını tanımlamaktadır

```

{ Tconnection }
constructor Tconnection.Create(aNo: Integer; AconnInfo: TconnectionInfo);
var
  I: Integer;
  N1, N2, N3, N4: Tnode;
  ElemHeight, TempH, TheArea: Double;
begin
  Info := AconnInfo;
  inherited Create(aNo, Info.Node1, Info.Node2, nil, nil, 0);
  ElementList := TelementList.Create;
  FsectionList := TsectionList.Create;
  FnodeList := TnodeList.Create;
  with Info do
  begin
    if Nlayers < 1 then raise Exception.Create('At least one layer should be
defined.');
```

```

    ElemHeight := Height / Nlayers;
    for I := 1 to 4 do FsectionList.Add(Tsection.Create(A[I], 0, 0, 0,
'Bracing'+IntToStr(I)));
    TheArea := ElemHeight * Width - (1.05 * A[1] / Nlayers);
    FsectionList.Add(Tsection.Create(TheArea, 0, 0, 0, 'Conc.Layer'));
    for I := 1 to Nsteel do FsectionList.Add(Tsection.Create(StArea[I], 0, 0, 0,
'Steel'+IntToStr(I)));
    { Create nodes }
    FnodeList.Add(TslaveNode.Create(1, Offset[1], Height*0.5, 0, rsFreeXY, Nodes[1]));
    FnodeList.Add(TslaveNode.Create(2, Length-Offset[2], Height*0.5, 0, rsFreeXY,
Nodes[2]));
    FnodeList.Add(TslaveNode.Create(3, Offset[1], -Height*0.5, 0, rsFreeXY, Nodes[1]));
    FnodeList.Add(TslaveNode.Create(4, Length-Offset[2], -Height*0.5, 0, rsFreeXY,
Nodes[2]));
    for I := 1 to Nlayers do
    begin
      TempH := (Height*0.5) - ElemHeight*(I-0.5);
      FnodeList.Add(TslaveNode.Create(2*I+3, Offset[1], TempH, 0, rsFreeXY, Nodes[1]));
      FnodeList.Add(TslaveNode.Create(2*I+4, (Length-Offset[2]), TempH, 0, rsFreeXY,
Nodes[2]));
    end;
  end;
end;

```

```

    for I := 1 to Nsteel do
    begin
        FnodeList.Add(TslaveNode.Create(2*(I+Nlayers)+3, Offset[1], StLoc[I], 0,
rsFreeXY, Nodes[1]));
        FnodeList.Add(TslaveNode.Create(2*(I+Nlayers)+4, (Length-Offset[2]), StLoc[I], 0,
rsFreeXY, Nodes[2]));
    end;
    { Create Layers }
    for I:= 1 to Nlayers do
        FelementList.Add(Ttruss3D.Create(I, FnodeList[2*I+2], FnodeList[2*I+3], Concrete,
FsectionList[4], 0));
        for I:= 1 to Nsteel do
            FelementList.Add(Ttruss3D.Create(I+Nlayers, FnodeList[2*I+2+2*Nlayers],
FnodeList[2*I+3+2*Nlayers], Steel, FsectionList[I+4], 0));
        { Create bracings }
        N1 := FnodeList[0];
        N2 := FnodeList[1];
        N3 := FnodeList[2];
        N4 := FnodeList[3];
        FelementList.Add(Ttruss3D.Create(Nlayers+Nsteel+1, N1, N4, Concrete,
FsectionList[0], 0));
        FelementList.Add(Ttruss3D.Create(Nlayers+Nsteel+2, N1, N4, Steel, FsectionList[1],
0));
        FelementList.Add(Ttruss3D.Create(Nlayers+Nsteel+3, N3, N2, Concrete,
FsectionList[2], 0));
        FelementList.Add(Ttruss3D.Create(Nlayers+Nsteel+4, N3, N2, Steel, FsectionList[3],
0));
    end;
    FnodeList.AssignDOF; { Assign internal DOF }
    IntDOF := FnodeList.NdoF;
end;

destructor Tconnection.Destroy;
begin
    FelementList.Free;
    FsectionList.Free;
    FnodeList.Free;
    inherited Destroy;
end;

procedure Tconnection.GetTransformation(Transformation: TdoubleMatrix);
var
    I: Integer;
    TheMaster: Tnode;
    TheSlave: TslaveNode;
begin
    for I:= 1 to FnodeList.Count do
    begin
        TheSlave := FnodeList[I-1];
        if TheSlave is TslaveNode then with TheSlave as TslaveNode do
        begin
            TheMaster := TheSlave.Master;
            if TheMaster = Nodes[1] then
            begin
                Transformation[TheSlave.DOF[1],1] := 1;
                Transformation[TheSlave.DOF[1],6] := -TheSlave.Y;
                Transformation[TheSlave.DOF[2],2] := 1;
                Transformation[TheSlave.DOF[2],6] := Info.Offset[1];
                Transformation[TheSlave.DOF[6],6] := 1;
            end else if TheMaster = Nodes[2] then
            begin
                Transformation[TheSlave.DOF[1],7] := 1;
                Transformation[TheSlave.DOF[1],12] := -TheSlave.Y;
                Transformation[TheSlave.DOF[2],8] := 1;
                Transformation[TheSlave.DOF[2],12] := -Info.Offset[2];
                Transformation[TheSlave.DOF[6],12] := 1;
            end;
        end;
    end;
end;

procedure Tconnection.GetStiffness(ElemStif: TdoubleMatrix);
var
    I, J: Integer;

```

```

    LargeElemStif, Transformation, Temp: TdoubleMatrix;
begin
    LargeElemStif := TdoubleMatrix.Create(IntDOF, IntDOF);
    Transformation := TdoubleMatrix.Create(IntDOF, 12); { Rigid links at both ends }
    Temp := TdoubleMatrix.Create(12, IntDOF);
    try
        ElemStif.Clear;
        FelementList.GetGlobalStiffness(LargeElemStif); { Banded }
        for I:= 2 to IntDOF do
            for J:= IntDOF downto I do
                LargeElemStif[I, J] := LargeElemStif[I, J-I+1]; { UnBand it }
            for I:=2 to IntDOF do
                for J:=1 to I do
                    LargeElemStif[I,J] := LargeElemStif[J, I];
                GetTransformation(Transformation);
                MultiplyTranspose(Transformation, LargeElemStif, Temp);
                MultiplyMatrix(Temp, Transformation, ElemStif);
            finally
                LargeElemStif.Free;
                Transformation.Free;
                Temp.Free;
            end;
        end;
end;

procedure Tconnection.WriteProperties(Output: Tstrings);
var I: Integer;
begin
    Output.Add(Format('%7d CONN %8d%8d', [No, Nodes[1].No, Nodes[2].No]));
    Output.Add('Internal elements with internal nodes');
    for I:=1 to FelementList.Count do
        with Telement(FelementList[I-1]) do
            Output.Add(Format('%7d TRUSS%8d%8d %-12s%-10s%9.5f', [No, Nodes[1].No,
                Nodes[2].No, Section.Name, Material.Name, Section.Area]));
            Output.Add('');
            Output.Add('ELEMENT TYPE NODE_I NODE_J SECTION MATERIAL ALPHA');
        end;
end;

procedure Tconnection.GetStress(TotalStress: TdoubleVector);
var
    ElemStif, RotMatrix: TdoubleMatrix;
    GelemStress: TdoubleVector;
    GelemDisp, ElemDisp: TdoubleVector;
    Transformation: TdoubleMatrix;
    InternDisp, InternForce: TdoubleVector;
    Indices: TintegerVector;
    TheDOF, I: Integer;
begin
    ElemStif := TdoubleMatrix.Create(12, 12);
    RotMatrix := TdoubleMatrix.Create(12, 12);
    GelemStress := TdoubleVector.Create(12);
    GelemDisp := TdoubleVector.Create(12);
    ElemDisp := TdoubleVector.Create(12);
    Transformation := TdoubleMatrix.Create(IntDOF, 12);
    InternDisp := TdoubleVector.Create(IntDOF);
    InternForce := TdoubleVector.Create(IntDOF);
    Indices := TintegerVector.Create(12);
    try
        for I:=1 to 6 do
            begin
                GelemDisp[I] := Nodes[1].Disp[I];
                GelemDisp[I + 6] := Nodes[2].Disp[I];
            end;
        GetRotation(RotMatrix);
        MultiplyMatrixVector(RotMatrix, GelemDisp, ElemDisp);
        GetTransformation(Transformation);
        MultiplyMatrixVector(Transformation, ElemDisp, InternDisp);
        FnodeList.AssignDisplacements(InternDisp);
        FelementList.CalculateStresses(InternForce);
        MultiplyMatrixVectorTrans(Transformation, InternForce, ElemStress);
        RotMatrix.Transpose;
        MultiplyMatrixVector(RotMatrix, ElemStress, GelemStress);
        GetIndices(Indices);
        for I:=1 to 12 do
            begin

```



```

        TheDOF := Indices[I];
        if TheDOF>0 then TotalStress[TheDOF] := TotalStress[TheDOF] + GelemStress[I];
        end;
    finally
        Indices.Free;
        InternForce.Free;
        InternDisp.Free;
        Transformation.Free;
        ElemDisp.Free;
        GelemDisp.Free;
        GelemStress.Free;
        RotMatrix.Free;
        ElemStif.Free;
    end;
end;

procedure Tconnection.WriteStress(Output: Tstrings);
var
    I: Cardinal;
    Truss: Ttruss3D;
    TheStress: Double;
    Failed: Boolean;
    Temp: String;
begin
    Output.Add(Format('%4d Connection Element', {No}));
    Output.Add(Format('%4d%13.5e%13.5e%13.5e%13.5e%13.5e%13.5e',
        {No, ElemStress[1], ElemStress[2], ElemStress[3], ElemStress[4],
        ElemStress[5], ElemStress[6]}));
    Output.Add(Format('%13.5e%13.5e%13.5e%13.5e%13.5e%13.5e',
        {ElemStress[7], ElemStress[8], ElemStress[9], ElemStress[10],
        ElemStress[11], ElemStress[12]}));
    Output.Add('Axial Stresses of Internal Elements: (Tension(+))');
    Output.Add('ELEM TYPE          STRESS          STRAIN          M.OF ELAS. ');
    for I:=1 to FelementList.Count do
    begin
        Truss := Ttruss3D(FelementList[I-1]);
        TheStress := Truss.Material.GetStress(Truss.Strain, Failed);
        Temp := Format('%4d %-10.10s %15.4e%15.4e%15.4e',
            {Truss.No, Truss.Section.Name, TheStress, Truss.Strain,
            Truss.Material.GetE(Truss.Strain)});
        if Failed then Temp := Temp + ' Cracked';
        Output.Add(Temp);
    end;
    for I:=1 to FnodeList.Count do
        with Tnode(FnodeList[I-1]) do
            Output.Add(Format('%4d%13.5e%13.5e%13.5e%13.5e%13.5e',
                {I, Disp[1], Disp[2], Disp[3], Disp[4], Disp[5], Disp[6]}));
    end;
end;

```

## A.6 STRUCTURE Objesi

STRUCTURE objesi NODE ve ELEMENT objelerinden oluşan tüm sistemin davranışını tanımlamaktadır.

```

unit Struct;

interface

uses Classes, SysUtils, Element, Node;

type
    Tstructure = class
    public
        Title: String;
        SectionList: TsectionList;
        MaterialList: TmaterialList;
        ElementList: TelementList;
    end;

```

```

    NodeList: TnodeList;
    Loads: TloadList;
    constructor Create;
    destructor Destroy; override;
    procedure Clear;
end;

implementation

uses Frame;

function ValidStr(const V: OpenString): String;
begin
    if V='' then Result := '0'
    else if V='-\' then Result := '0'
    else Result := V;
end;

( Tstructure )
constructor Tstructure.Create;
begin
    inherited Create;
    NodeList := TnodeList.Create;
    ElementList := TelementList.Create;
    SectionList := TsectionList.Create;
    MaterialList := TmaterialList.Create;
    Loads := TloadList.Create;
end;

destructor Tstructure.Destroy;
begin
    NodeList.Free;
    ElementList.Free;
    SectionList.Free;
    MaterialList.Free;
    Loads.Free;
    inherited Destroy;
end;

procedure Tstructure.Clear;
begin
    NodeList.Free;
    ElementList.Free;
    SectionList.Free;
    MaterialList.Free;
    Loads.Free;
    NodeList := TnodeList.Create;
    ElementList := TelementList.Create;
    SectionList := TsectionList.Create;
    MaterialList := TmaterialList.Create;
    Loads := TloadList.Create;
end;

end.

```

## A.7 SOLVER Objesi

SOLVER objesi çözüm algoritmalarının işleyişini tanımlamaktadır.

```

unit Solver;

interface

uses Classes, Struct, SysUtils, Matrix, Node, Element, Frame;

type
    Tsolver = class
    private

```

```

    Fstructure: Tstructure;
    Foutput: Tstrings;
    Fstatus: string;
    FonProgress: TnotifyEvent;
    procedure WriteOut(Str: String);
    procedure PrintStructure;
    procedure PrintLoads;
    procedure DecomposeBand(SS: TdoubleMatrix);
    procedure SolveBand(SS: TdoubleMatrix; PP, UU: TdoubleVector);
    procedure PrintDisplacements;
    procedure PrintForces(TS: TdoubleVector);
    procedure SetStatus(NewStatus: string);
public
    constructor Create(Astructure: Tstructure);
    destructor Destroy; override;
    procedure CheckInput;
    procedure Calculate; virtual;
    property Output: Tstrings read Foutput write Foutput;
    property Structure: Tstructure read Fstructure write Fstructure;
    property Status: string read Fstatus write SetStatus;
    property OnProgress: TnotifyEvent read FonProgress write FonProgress;
end;

TlinearSolver = class(Tsolver);

TnonLinearSolver = class(Tsolver)
private
    Niter, Nstep: Integer;
end;

TNRSolver = class(TnonLinearSolver)
public
    constructor Create(Astructure: Tstructure; Aiter: Integer);
    procedure Calculate; override;
end;

TMNRSolver = class(TNRSolver)
public
    procedure Calculate; override;
end;

TstepSolver = class(TNRSolver)
public
    constructor Create(Astructure: Tstructure; Aiter, Astep: Integer);
    procedure Calculate; override;
end;

TarcLengthSolver = class(TNRSolver)
public
    constructor Create(Astructure: Tstructure; Aiter, Astep: Integer);
    procedure Calculate; override;
end;

const
    ElemSize = 12;

implementation

uses Math;

{ Tsolver }
constructor Tsolver.Create;
begin
    inherited Create;
    Fstructure := Astructure;
    Foutput := TStringList.Create;
end;

destructor Tsolver.Destroy;
begin
    Foutput.Free;
    inherited Destroy;
end;

```

```

procedure Tsolver.SetStatus(NewStatus: string);
begin
  if NewStatus <> Fstatus then
    begin
      Fstatus := NewStatus;
      if Assigned(FonProgress) then FonProgress(Self);
    end;
end;

procedure Tsolver.CheckInput;
begin
  with Fstructure do
    begin
      if NodeList.Count=0 then raise Exception.Create('No nodes defined.');
```

```

      if SectionList.Count=0 then raise Exception.Create('No sections defined.');
```

```

      if MaterialList.Count=0 then raise Exception.Create('No materials defined.');
```

```

      if ElementList.Count=0 then raise Exception.Create('No elements defined.');
```

```

    end;
end;

procedure Tsolver.WriteOut(Str: String);
begin
  if Assigned(Foutput) then Foutput.Add(Str);
end;

procedure Tsolver.PrintStructure;
var
  IBC: array[1..6] of Byte;
  I, J: Integer;
  R: Trestraint;
begin
  { Input nodal coordinates and number the d.o.f at each node. }
  with Fstructure do
    begin
      WriteOut(Title);
      WriteOut(' ');
      WriteOut('SYSTEM DATA');
      WriteOut('*****');
      WriteOut(' ');
      WriteOut(Format('Number of nodal points =%4d',[NodeList.Count]));
      WriteOut(Format('Number of elements =%4d',[ElementList.Count]));
      WriteOut(Format('Number of sections =%4d',[SectionList.Count]));
      WriteOut(Format('Number of materials =%4d',[MaterialList.Count]));
      WriteOut(' ');
      WriteOut('NODAL POINT DATA');
      WriteOut('*****');
      WriteOut(' ');
      WriteOut('NODE X-COORD Y-COORD Z-COORD DX DY DZ RX RY RZ');
      for I:=1 to NodeList.Count do
        with Tnode(NodeList[I-1]) do
          begin
            for J:=1 to 6 do IBC[J]:=0;
            R := Restraint;
            if rtX in R then IBC[1] := 1;
            if rtY in R then IBC[2] := 1;
            if rtZ in R then IBC[3] := 1;
            if rtRX in R then IBC[4] := 1;
            if rtRY in R then IBC[5] := 1;
            if rtRZ in R then IBC[6] := 1;
            WriteOut(Format('%4d%9.2f%9.2f%9.2f%4d%4d%4d%4d%4d%4d',
              [No, X, Y, Z, IBC[1], IBC[2], IBC[3], IBC[4], IBC[5], IBC[6]]));
          end;
        end;
      WriteOut(' ');
    end;
  { Input sectional and material data for different Elements. }
  WriteOut(' ');
  WriteOut('ELEMENT SECTION TYPES');
  WriteOut('*****');
  WriteOut(' ');
  WriteOut('NO NAME AREA(A) INERTIA(Ixx) INERTIA(Iyy) INERTIA(Izz)');
  for I:=1 to SectionList.Count do
    with Tsection(SectionList[I-1]) do
      WriteOut(Format('%3d %-10s%11.4e%14.4e%14.4e%14.4e',[I, Name, Area, Ixx, Iyy,
        Izz]));
    end;
  WriteOut(' ');
  WriteOut('ELEMENT MATERIAL TYPES');

```

```

WriteOut('*****');
WriteOut('');
WriteOut('NO NAME MOD. OF ELAST. MOD. OF SHEAR');
for I:=1 to MaterialList.Count do
with Tmaterial(MaterialList[I-1]) do
WriteOut(Format('%3d %-12s%14.4e%16.4e', [I, Name, GetE(0), G]));
( Input for each member the descriptive data. )
WriteOut('');
WriteOut('ELEMENT DEFINITIONS');
WriteOut('*****');
WriteOut('');
WriteOut('ELEMENT TYPE NODE_I NODE_J SECTION MATERIAL ALPHA');
for I:=1 to ElementList.Count do
Telement(ElementList[I-1]).WriteProperties(Foutput);
end;
end;

procedure Tsolver.PrintLoads;
var
I: Integer;
begin
WriteOut('');
WriteOut('NODAL LOADS');
WriteOut('*****');
WriteOut('');
WriteOut('NODE FORCE X FORCE Y FORCE Z MOMENT X MOMENT Y MOMENT Z');
with Fstructure do
for I:=1 to Loads.Count do
with TnodalLoad(Loads[I-1]) do
WriteOut(Format('%4d%10.2f%10.2f%10.2f%10.2f%10.2f%10.2f',
[Node.No, FX, FY, FZ, MX, MY, MZ]));
end;
end;

procedure Tsolver.DecomposeBand(SS: TdoubleMatrix);
var
I, J, K, N, P, Q, IBW: Integer;
Sum, Temp: Double;
begin
Temp := 0;
N := SS.Rows;
IBW := SS.Cols;
for I:=1 to N do
begin
P:=N-I+1;
if P>IBW then P:=IBW;
for J:=1 to P do
begin
Q:=IBW-J;
if Q>I-1 then Q:=I-1;
Sum:=SS[I,J];
for K:=1 to Q do
Sum:=Sum-SS[I-K,K+1]+SS[I-K,J+K];
if J<>1 then
SS[I,J]:=Sum+Temp
else
begin
if Sum<=0 then raise Exception.Create('Structure is unstable');
Temp:=1/Sqrt(Sum);
SS[I,J]:=Temp;
end;
end;
end;
end;

procedure Tsolver.SolveBand(SS: TdoubleMatrix; PP, UU: TdoubleVector);
var
I, J, K, N, IBW: Integer;
Sum: Double;
begin
N := SS.Rows;
IBW := SS.Cols;
for I:=1 to N do
begin
J:=I-IBW+1;

```

```

    if (I+1)<=IBW then J:=1;
    Sum:=PP[I];
    for K:=J to I-1 do
        Sum:=Sum-SS[K,I-K+1]*UU[K];
    UU[I]:=Sum*SS[I,1];
end;
for I:=N downto 1 do
begin
    J:=I+IBW-1;
    if J>N then J:=N;
    Sum:=UU[I];
    for K:=I+1 to J do
        Sum:=Sum-SS[I,K-I+1]*UU[K];
    UU[I]:=Sum*SS[I,1];
end;
end;

procedure Tsolver.PrintDisplacements;
var
    I: Integer;
begin
    WriteOut('');
    WriteOut('NODAL POINT DISPLACEMENTS');
    WriteOut('*****');
    WriteOut('');
    WriteOut('NODE      X-DISP      Y-DISP      Z-DISP      X-ROT      Y-ROT
Z-ROT');
    with Fstructure do
    for I:=1 to NodeList.Count do
    with Tnode(NodeList[I-1]) do
        WriteOut(Format('%4d%13.5e%13.5e%13.5e%13.5e%13.5e',
            [I, Disp[1], Disp[2], Disp[3], Disp[4], Disp[5], Disp[6]]));
end;

procedure Tsolver.PrintForces(TS: TdoubleVector);
var
    IL: Integer;
begin
    WriteOut('');
    WriteOut('ELEMENT END FORCES IN LOCAL DIRECTIONS');
    WriteOut('*****');
    WriteOut('ELEM  NODE1      F1          F2          F3          F4          F5
F6');
    WriteOut('      NODE2      F1          F2          F3          F4          F5
F6');
    with Fstructure do
    for IL:=1 to ElementList.Count do
        Telement(ElementList[IL-1]).WriteStress(Foutput);
end;

procedure Tsolver.Calculate;
var
    N dof, IBW: Cardinal;
    SS: TdoubleMatrix;
    PP, UU: TdoubleVector;
begin
    UU := nil;
    PP := nil;
    SS := nil;
    try
        CheckInput;
    except
        Raise;
    end;
    try
        Foutput.Clear;
        Status := 'Echoing input';
        PrintStructure;
        PrintLoads;
        Status := 'Creating stiffness matrix';
        with Fstructure do
        begin
            NodeList.AssignDOF;
            N dof := NodeList.N dof;

```

```

        IBW := ElementList.Bandwidth;
        SS := TdoubleMatrix.Create(Ndof, IBW);
        PP := TdoubleVector.Create(Ndof);
        UU := TdoubleVector.Create(Ndof);
        ElementList.GetGlobalStiffness(SS);
        Loads.GetLoadVector(PP);
    end;
    WriteOut(' ');
    WriteOut('*****');
    WriteOut('LINEAR SOLUTION');
    WriteOut('*****');
    Status := 'Solving equations';
    DecomposeBand(SS);
    SolveBand(SS, PP, UU);
    Fstructure.NodeList.AssignDisplacements(UU);
    Status := 'Writing output';
    PrintDisplacements;
    Fstructure.ElementList.CalculateStresses(PP);
    PrintForces(PP);
finally
    UU.Free;
    PP.Free;
    SS.Free;
end;
end;

{ TNRSolver }
constructor TNRSolver.Create(Astructure: Tstructure; Aiter: Integer);
begin
    inherited Create(Astructure);
    Niter := Aiter;
end;

procedure TNRSolver.Calculate;
var
    Ndof, IBW, Iteration, I: Cardinal;
    Tolerance: Double;
    TotalStiffness: TdoubleMatrix;
    TotalDisp, DeltaDisp, Residual, TotalLoad, TotalStress: TdoubleVector;
begin
    TotalStiffness := nil;
    TotalDisp := nil;
    DeltaDisp := nil;
    Residual := nil;
    TotalLoad := nil;
    TotalStress := nil;
    try
        CheckInput;
    except
        Raise;
    end;
    try
        Foutput.Clear;
        Status := 'Echoing input';
        PrintStructure;
        PrintLoads;
        Status := 'Creating stiffness matrix';
        with Fstructure do
            begin
                NodeList.AssignDOF;
                Ndof := NodeList.Ndof;
                IBW := ElementList.Bandwidth;
                TotalStiffness := TdoubleMatrix.Create(Ndof, IBW);
                TotalLoad := TdoubleVector.Create(Ndof);
                TotalDisp := TdoubleVector.Create(Ndof);
                DeltaDisp := TdoubleVector.Create(Ndof);
                TotalStress := TdoubleVector.Create(Ndof);
                Residual := TdoubleVector.Create(Ndof);
                Loads.GetLoadVector(TotalLoad);
            end;
        end;
        WriteOut(' ');
        WriteOut('*****');
        WriteOut('NONLINEAR SOLUTION USING NEWTON-RAPHSON METHOD');
        WriteOut('*****');
    end;
end;

```

```

Iteration := 0;
Tolerance := 0.00001;
Residual.Assign(TotalLoad);
repeat
  Inc(Iteration);
  Status := 'Solving equations';
  TotalStiffness.Clear;
  Estructure.ElementList.GetGlobalStiffness(TotalStiffness);
  DecomposeBand(TotalStiffness);
  SolveBand(TotalStiffness, Residual, DeltaDisp);
  WriteOut('');
  WriteOut(Format('ITERATION %d', [Iteration]));
  TotalDisp.Add(DeltaDisp);
  Estructure.NodeList.AssignDisplacements(TotalDisp);
  Status := 'Writing output';
  PrintDisplacements;
  TotalStress.Clear;
  Estructure.ElementList.CalculateStresses(TotalStress);
  PrintForces(TotalStress);
  for I:=1 to NdoF do Residual[I] := TotalLoad[I] - TotalStress[I];
  until (Iteration > Niter) or ((DeltaDisp.Norm / TotalDisp.Norm) < Tolerance) or
(Residual.Norm < Tolerance);
  WriteOut('');
  if Iteration > Niter then WriteOut('Maximum number of iterations is reached.')
  else WriteOut('Iterations converged.');
```

```

finally
  Residual.Free;
  TotalStress.Free;
  DeltaDisp.Free;
  TotalDisp.Free;
  TotalLoad.Free;
  TotalStiffness.Free;
end;
end;

( TMNRSolver )
procedure TMNRSolver.Calculate;
var
  NdoF, IBW, Iteration, I: Cardinal;
  Tolerance: Double;
  TotalStiffness: TdoubleMatrix;
  TotalDisp, DeltaDisp, Residual, TotalLoad, TotalStress: TdoubleVector;
begin
  TotalStiffness := nil;
  TotalDisp := nil;
  DeltaDisp := nil;
  Residual := nil;
  TotalLoad := nil;
  TotalStress := nil;
  try
    CheckInput;
  except
    Raise;
  end;
  try
    Foutput.Clear;
    Status := 'Echoing input';
    PrintStructure;
    PrintLoads;
    Status := 'Creating stiffness matrix';
    with Fstructure do
      begin
        NodeList.AssignDOF;
        NdoF := NodeList.NdoF;
        IBW := ElementList.Bandwidth;
        TotalStiffness := TdoubleMatrix.Create(NdoF, IBW);
        TotalLoad := TdoubleVector.Create(NdoF);
        TotalDisp := TdoubleVector.Create(NdoF);
        DeltaDisp := TdoubleVector.Create(NdoF);
        TotalStress := TdoubleVector.Create(NdoF);
        Residual := TdoubleVector.Create(NdoF);
        ElementList.GetGlobalStiffness(TotalStiffness);
        Loads.GetLoadVector(TotalLoad);
      end;
    end;
  end;
end;

```



```

WriteOut('');
WriteOut('*****');
WriteOut('NONLINEAR SOLUTION USING MODIFIED NEWTON-RAPHSON METHOD');
WriteOut('*****');
Iteration := 0;
Tolerance := 0.00001;
Residual.Assign(TotalLoad);
DecomposeBand(TotalStiffness);
repeat
  Inc(Iteration);
  Status := 'Solving equations';
  SolveBand(TotalStiffness, Residual, DeltaDisp);
  WriteOut('');
  WriteOut(Format('ITERATION %d', [Iteration]));
  TotalDisp.Add(DeltaDisp);
  Estructure.NodeList.AssignDisplacements(TotalDisp);
  Status := 'Calculating displacements';
  PrintDisplacements;
  Status := 'Calculating stresses';
  TotalStress.Clear;
  Estructure.ElementList.CalculateStresses(TotalStress);
  PrintForces(TotalStress);
  for I:=1 to N dof do Residual[I] := TotalLoad[I] - TotalStress[I];
until (Iteration > Niter) or ((DeltaDisp.Norm / TotalDisp.Norm) < Tolerance) or
(Residual.Norm < Tolerance);
WriteOut('');
if Iteration > Niter then WriteOut('Maximum number of iterations is reached.')
else WriteOut('Iterations converged.');
```

```

finally
  Residual.Free;
  TotalStress.Free;
  DeltaDisp.Free;
  TotalDisp.Free;
  TotalLoad.Free;
  TotalStiffness.Free;
end;
end;

{ TstepSolver }
constructor TstepSolver.Create(Astructure: Tstructure; Aiter, Astep: Integer);
begin
  inherited Create(Astructure, Aiter);
  Nstep := Astep;
  if Nstep < 1 then Nstep := 1;
end;

procedure TstepSolver.Calculate;
var
  N dof, IBW, Iteration, I, J: Cardinal;
  Tolerance: Double;
  TotalStiffness: TdoubleMatrix;
  TotalDisp, Ddisp, Residual, TotalLoad, TotalStress, Dload, EF, Tdisp: TdoubleVector;
begin
  TotalStiffness := nil;
  TotalDisp := nil;
  Ddisp := nil;
  Residual := nil;
  TotalLoad := nil;
  TotalStress := nil;
  try
    Status := 'Reading input';
    CheckInput;
  except
    Raise;
  end;
  try
    Foutput.Clear;
    Status := 'Echoing input';
    PrintStructure;
    PrintLoads;
    Status := 'Setting up matrices';
    with Estructure do
      begin
        NodeList.AssignDOF;

```

```

N dof := NodeList.Ndof;
IBW := ElementList.Bandwidth;
TotalStiffness := TdoubleMatrix.Create(Ndof, IBW);
TotalLoad := TdoubleVector.Create(Ndof);
TotalDisp := TdoubleVector.Create(Ndof);
Ddisp := TdoubleVector.Create(Ndof);
Dload := TdoubleVector.Create(Ndof);
EF := TdoubleVector.Create(Ndof);
Tdisp := TdoubleVector.Create(Ndof);
TotalStress := TdoubleVector.Create(Ndof);
Residual := TdoubleVector.Create(Ndof);
Loads.GetLoadVector(TotalLoad);
for I := 1 to Ndof do Dload[I] := TotalLoad[I] / Nstep;
end;
WriteOut('');
WriteOut('*****');
WriteOut('NONLINEAR SOLUTION USING MODIFIED NEWTON-RAPHSON METHOD WITH STEP
LOADING');
WriteOut('*****');
Tolerance := 0.00001;
for I := 1 to Nstep do
begin
Iteration := 0;
WriteOut('');
WriteOut(Format('LOAD STEP %d', [I]));
Status := 'Creating stiffness matrix, load step = '+IntToStr(I);
TotalStiffness.Clear;
Fstructure.ElementList.GetGlobalStiffness(TotalStiffness);
Status := 'Solving equations, load step = '+IntToStr(I);
DecomposeBand(TotalStiffness);
EF.Add(Dload);
SolveBand(TotalStiffness, Dload, Ddisp);
repeat
Inc(Iteration);
Tdisp.Assign(TotalDisp);
Tdisp.Add(Ddisp);
Status := 'Calculating displacements, load step = '+IntToStr(Nstep);
Fstructure.NodeList.AssignDisplacements(Tdisp);
Status := 'Calculating stresses, load step = '+IntToStr(I);
TotalStress.Clear;
Fstructure.ElementList.CalculateStresses(TotalStress);
for J:=1 to Ndof do Residual[J] := EF[J] - TotalStress[J];
Status := 'Solving equations';
SolveBand(TotalStiffness, Residual, Tdisp);
Ddisp.Add(Tdisp);
until (Iteration > Niter) or ((Tdisp.Norm / Ddisp.Norm) < Tolerance) or
(Residual.Norm < Tolerance);
WriteOut('');
if Iteration > Niter then WriteOut('Maximum number of iterations is reached.')
else WriteOut('Iterations converged. ');
Status := 'Calculating displacements';
TotalDisp.Add(Ddisp);
Fstructure.NodeList.AssignDisplacements(TotalDisp);
PrintDisplacements;
Status := 'Calculating stresses';
TotalStress.Clear;
Fstructure.ElementList.CalculateStresses(TotalStress);
PrintForces(TotalStress);
end;
finally
Residual.Free;
Dload.Free;
EF.Free;
Tdisp.Free;
TotalStress.Free;
Ddisp.Free;
TotalDisp.Free;
TotalLoad.Free;
TotalStiffness.Free;
end;
end;

{ TarLengthSolver }
constructor TarLengthSolver.Create(Astructure: Tstructure; Aiter, Astep: Integer);

```

```

begin
  inherited Create(Astructure, Aiter);
  Nstep := Astep;
  if Nstep < 1 then Nstep := 1;
end;

procedure TarclengthSolver.Calculate;
var
  N dof, IBW, Iteration, I, ID, J, N: Cardinal;
  Psi, L, Labs, Lmax, C, Rho, XI, Eta, DXI, EPS, TheLoad: Double;
  K: TdoubleMatrix;
  A, DA, TA, DAOld, DASTart, FI, DQ, R, FE, Delta, DF, EF: TdoubleVector;
  NewStep, ARC, ORD, QN: Boolean;
begin
  Status := 'Reading input';
  CheckInput;
  try
    Foutput.Clear;
    Status := 'Echoing input';
    PrintStructure;
    PrintLoads;
    Status := 'Setting up matrices';
    with Fstructure do
      begin
        NodeList.AssignDOF;
        N dof := NodeList.N dof;
        IBW := ElementList.Bandwidth;
        K := TdoubleMatrix.Create(N dof, IBW);
        A := TdoubleVector.Create(N dof);
        DA := TdoubleVector.Create(N dof);
        TA := TdoubleVector.Create(N dof);
        DAOld := TdoubleVector.Create(N dof);
        DASTart := TdoubleVector.Create(N dof);
        FI := TdoubleVector.Create(N dof);
        DQ := TdoubleVector.Create(N dof);
        R := TdoubleVector.Create(N dof);
        FE := TdoubleVector.Create(N dof);
        Delta := TdoubleVector.Create(N dof);
        DF := TdoubleVector.Create(N dof);
        EF := TdoubleVector.Create(N dof);
        Loads.GetLoadVector(FE);
        for I := 1 to N dof do DF[I] := FE[I] / N step;
      end;
    WriteOut('');
    WriteOut('*****');
    WriteOut('NONLINEAR SOLUTION USING ARC LENGTH METHOD');
    WriteOut('*****');
    Psi := 1;
    C := 1;
    ID := 100;
    I := 0;
    EPS := 0.001;
    NewStep := True;
    ARC := False;
    ORD := True;
    QN := False;
    for N := 1 to N step do
      begin
        WriteOut('');
        WriteOut(Format('LOAD STEP %d', [N]));
        if NewStep then begin
          Status := 'Generating stiffness matrix';
          K.Clear;
          Fstructure.ElementList.GetGlobalStiffness(K);
          Status := 'Solving equations';
          DecomposeBand(K);
          DAOld.Assign(DA);
          SolveBand(K, DF, DA);
          if (Dot(DAOld, DA) < 0) then begin
            DF.ScaleBy(-1);
            DA.ScaleBy(-1);
          end;
          L := DA.Norm;
          if N=1 then begin

```

```

        Labs := C * L;
        Lmax := Labs;
    end;
    if (I < ID) then Lmax := MinValue([Labs, 2*Lmax]);
    Rho := MinValue([1, Lmax/L]);
    DASTart.Assign(DA);
    DA.ScaleBy(Rho);
    if ARC then XI := Rho;
end;
I := 0;
repeat
    Inc(I);
    FI.Clear;
    TA.Assign(A);
    TA.Add(DA);
    Fstructure.NodeList.AssignDisplacements(TA);
    Fstructure.ElementList.CalculateStresses(FI);
    for J:=1 to Ndef do DQ[J] := FI[J] - FE[J];
    if ORD then XI := Dot(DQ, DA)/Dot(DF, DA);
    TA.Assign(DQ);
    TA.ScaleBy(-1);
    R.Assign(TA);
    TA.Assign(DF);
    TA.ScaleBy(XI);
    R.Add(TA);
    SolveBand(K, R, Delta);
    if (ORD and QN) then begin
        Eta := -Dot(DQ, Delta)/Dot(DQ, DA);
        Delta.ScaleBy(1 / (1 + Eta));
    end;
    DA.Add(Delta);
until (R.Norm<EPS*DF.Norm) or (I>Niter);
if I>Niter then begin
    WriteOut(' ');
    WriteOut('Maximum number of iterations reached');
    NewStep := False;
    Psi := Psi * 0.5;
    DA.Assign(DASTart);
    DA.ScaleBy(Psi * Rho);
    if ARC then XI := Rho * Psi;
end else begin
    WriteOut(' ');
    WriteOut(Format('Converged at %dth iteration', [I]));
    NewStep := True;
    Lmax := Lmax * Psi;
    Psi := 1;
    A.Add(DA);
    TA.Assign(DF);
    TA.ScaleBy(XI);
    FE.Add(TA);
    Status := 'Writing output';
    Fstructure.NodeList.AssignDisplacements(A);
    PrintDisplacements;
    Fstructure.ElementList.CalculateStresses(FI);
    PrintForces(FI);
end;
end;
finally
    if N<Nstep then begin
        WriteOut(' ');
        WriteOut(Format('Unstable at %dth iteration', [I]));
    end;
    EF.Free;
    DF.Free;
    Delta.Free;
    FE.Free;
    R.Free;
    DQ.Free;
    FI.Free;
    DASTart.Free;
    DAOld.Free;
    TA.Free;
    DA.Free;
    A.Free;

```

```

    K.Free;
  end;
end;

end.

```

## A.8 PROBLEM Objesi

PROBLEM objesi yazılımın ana sınıfı olup STRUCTURE objesi ile bunun analizini kontrol etmektedir.

```

unit Problem;

interface

uses
  Classes, SysUtils, Struct, Solver;

type
  TSolutionMethod = (Linear, NewtonRaphson, ModNewtonRaphson, StepLoad,
    OrthogonalResidual);

  TProblem = class
  private
    FStructure: TStructure;
    FSolver: TSolver;
    InFile, OutFile: TFileName;
    FMethod: TSolutionMethod;
    FIter, FStep: Integer;
    procedure ReadFromFile;
  public
    constructor Create(AInFileName, AOutFileName: TFileName);
    procedure Run;
  end;

implementation

uses
  Node, Element, Frame, Parser;

{ TProblem }
constructor TProblem.Create;
begin
  inherited Create;
  InFile := AInFileName;
  OutFile := AOutFileName;
end;

procedure TProblem.Run;
begin
  FStructure := TStructure.Create;
  try
    ReadFromFile;
    case FMethod of
      Linear: FSolver := TLinearSolver.Create(FStructure);
      NewtonRaphson: FSolver := TNRSolver.Create(FStructure, FIter);
      ModNewtonRaphson: FSolver := TMNRSolver.Create(FStructure, FIter);
      StepLoad: FSolver := TStepSolver.Create(FStructure, FIter, FStep);
      OrthogonalResidual: FSolver := TArcLengthSolver.Create(FStructure, FIter, FStep);
    end;
    FSolver.OnProgress := SolverOnProgress;
    FSolver.Calculate;
  finally
    if Assigned(FSolver) then FSolver.Output.SaveToFile(OutFile);
    if Assigned(FSolver) then FSolver.Free;
    FStructure.Free;
  end;
end;
end;

```

```

procedure TProblem.ReadFromFile;
var
  I, No, I1, I2, I3, I4, I5, I6, CurrentPart: Integer;
  D1, D2, D3, A1, A2: Double;
  S, S1: String;
  Inf: Text;
  Rest: TRestraint;
  Mat: TMaterial;
  Sect: TSection;
  Elem: TPrismatic3D;
  ConnInfo: TConnectionInfo;
  MatInfo: TMaterialInfo;
  NodalLoad: TNodalLoad;
  N1, N2: TNode;
  StrParser: TStrParser;
begin
  StrParser := TStrParser.Create('');
  try
    AssignFile(Inf, InFile);
    Reset(Inf);
    with FStructure do
    begin
      CurrentPart := 1;
      Rest := rsFree;
      while not Eof(Inf) do
      begin
        ReadLn(Inf, S);
        if (Trim(S)='') or (S[1]=' ') then Continue;
        if (S[1]='C') and (S[2]=' ') then Continue;
        StrParser.Source := S;
        S1 := StrParser.GetStringByOrder(1, '');
        if S1='SYSTEM' then CurrentPart := 2
        else if S1='NODES' then CurrentPart := 3
        else if S1='RESTRAINTS' then CurrentPart := 4
        else if S1='MATERIALS' then CurrentPart := 5
        else if S1='SECTIONS' then CurrentPart := 6
        else if S1='FRAME' then CurrentPart := 7
        else if S1='TRUSS' then CurrentPart := 8
        else if S1='CONNECTION' then CurrentPart := 9
        else if S1='LOADS' then CurrentPart := 10
        else with StrParser do
          case CurrentPart of
            1: begin { TITLE }
                  Title := S;
                end;
            2: begin { SYSTEM }
                  try
                    FMethod := TSolutionMethod(GetIntegerByName('METHOD', 1, 0));
                    FIter := GetIntegerByName('NITER', 1, 100);
                    FStep := GetIntegerByName('NSTEP', 1, 1);
                    A1 := GetDoubleByName('A', 1, 0.15);
                    A2 := GetDoubleByName('A', 2, 0.20);
                  except
                    raise Exception.Create('Error in SYSTEM block');
                  end;
                end;
            3: begin { NODES }
                  try
                    No := GetIntegerByOrder(1, 0);
                    D1 := GetDoubleByOrder(2, 0);
                    D2 := GetDoubleByOrder(3, 0);
                    D3 := GetDoubleByOrder(4, 0);
                    NodeList.CreateParams(No, D1, D2, D3, Rest);
                  except
                    raise Exception.Create('Error in NODES block');
                  end;
                end;
            4: begin { RESTRAINTS }
                  Rest := rsFree;
                  try
                    No := GetIntegerByOrder(1, 0);
                    I1 := GetIntegerByName('R', 1, 0);
                    I2 := GetIntegerByName('R', 2, 0);
                    I3 := GetIntegerByName('R', 3, 0);

```

```

I4 := GetIntegerByName('R', 4, 0);
I5 := GetIntegerByName('R', 5, 0);
I6 := GetIntegerByName('R', 6, 0);
if I1 = 1 then Include(Rest, rtX);
if I2 = 1 then Include(Rest, rtY);
if I3 = 1 then Include(Rest, rtZ);
if I4 = 1 then Include(Rest, rtRX);
if I5 = 1 then Include(Rest, rtRY);
if I6 = 1 then Include(Rest, rtRZ);
TNode(NodeList[No-1]).Restraint := Rest;
except
raise Exception.Create('Error in RESTRAINTS block');
end;
end;
5: begin { MATERIALS }
try
D1 := GetDoubleByName('CONCRETE', 1, 0);
if D1 > 0 then
Mat := TConcrete.Create(D1)
else begin
with MatInfo do begin
No := GetIntegerByOrder(1, 0);
G := GetDoubleByName('G', 1, 0);
Name := GetStringByName('NAME', 1, 'Material'+IntToStr(No));
NPoints := GetIntegerByName('NP', 1, 0);
if NPoints > 1 then begin
for I := 1 to NPoints do ReadLn(Inf, Strain[I], Stress[I]);
YieldStrain := Strain[NPoints];
end;
end;
Mat := TMaterial.Create(MatInfo);
end;
MaterialList.Add(Mat);
except
raise Exception.CreateFmt('Error in reading material %d', [No]);
end;
end;
6: begin { SECTIONS }
try
No := GetIntegerByOrder(1, 0);
Sect := TSection.Create(GetDoubleByName('A', 1, 0),
GetDoubleByName('I', 1, 0), GetDoubleByName('I', 2, 0),
GetDoubleByName('I', 3, 0),
GetStringByName('NAME', 1, 'Section'+IntToStr(No)));
SectionList.Add(Sect);
except
raise Exception.CreateFmt('Error in reading section %d', [No]);
end;
end;
7: begin { FRAME }
try
No := GetIntegerByOrder(1, 0);
I1 := GetIntegerByOrder(2, 0);
I2 := GetIntegerByOrder(3, 0);
I3 := GetIntegerByName('M', 1, 0);
I4 := GetIntegerByName('S', 1, 0);
D1 := GetDoubleByName('ALPHA', 1, 0);
N1 := NodeList[I1-1];
N2 := NodeList[I2-1];
Mat := MaterialList[I3-1];
Sect := SectionList[I4-1];
Elem := TFrame3D.Create(No, N1, N2, Mat, Sect, D1);
ElementList.Add(Elem);
except
raise Exception.CreateFmt('Error in reading frame element %d', [No]);
end;
end;
8: begin { TRUSS }
try
No := GetIntegerByOrder(1, 0);
I1 := GetIntegerByOrder(2, 0);
I2 := GetIntegerByOrder(3, 0);
I3 := GetIntegerByName('M', 1, 0);
I4 := GetIntegerByName('S', 1, 0);

```

```

D1 := GetDoubleByName('ALPHA', 1, 0);
N1 := NodeList[I1-1];
N2 := NodeList[I2-1];
Mat := MaterialList[I3-1];
Sect := SectionList[I4-1];
Elem := TTrouss3D.Create(No, N1, N2, Mat, Sect, D1);
ElementList.Add(Elem);
except
  raise Exception.CreateFmt('Error in reading truss element %d', [No]);
end;
end;
9: begin { CONNECTION }
  try
    No := GetIntegerByOrder(1, 0);
    I1 := GetIntegerByOrder(2, 0);
    I2 := GetIntegerByOrder(3, 0);
    I3 := GetIntegerByName('MCON', 1, 0);
    I4 := GetIntegerByName('MST', 1, 0);
    with ConnInfo do begin
      Nodel := NodeList[I1-1];
      Node2 := NodeList[I2-1];
      Concrete := MaterialList[I3-1];
      Steel := MaterialList[I4-1];
      Offset[1] := GetDoubleByName('RE', 1, 0);
      Offset[2] := GetDoubleByName('RE', 2, 0);
      Width := GetDoubleByName('W', 1, 0);
      Height := GetDoubleByName('H', 1, 0);
      NLayers := GetIntegerByName('NL', 1, 0);
      NSteel := GetIntegerByName('NL', 2, 0);
      for I := 1 to NSteel do begin
        StLoc[I] := GetDoubleByName('ST', I*2-1, 0);
        StArea[I] := GetDoubleByName('ST', I*2, 0);
      end;
      D1 := GetDoubleByName('TIE', 1, 0);
      D2 := GetDoubleByName('TIE', 2, 0);
      A[1] := GetDoubleByName('A', 1, A1);
      A[2] := GetDoubleByName('A', 2, A2);
      A[3] := GetDoubleByName('A', 3, A[1]);
      A[4] := GetDoubleByName('A', 4, A[2]);
      if (D1 <> 0) and (D2 <> 0) then begin
        A[1] := Width * Height * A[1];
        A[2] := D1 * Height * A[2] / D2;
        A[3] := A[1];
        A[4] := A[2];
      end;
    end;
    Elem := TConnection.Create(No, ConnInfo);
    ElementList.Add(Elem);
  except
    raise Exception.CreateFmt('Error in reading connection %d', [No]);
  end;
end;
10: begin { LOADS }
  try
    No := GetIntegerByOrder(1, 0);
    N1 := NodeList[No-1];
    NodalLoad := TNodalLoad.Create(N1, GetDoubleByName('F', 1, 0),
      GetDoubleByName('F', 2, 0), GetDoubleByName('F', 3, 0),
      GetDoubleByName('F', 4, 0), GetDoubleByName('F', 5, 0),
      GetDoubleByName('F', 6, 0));
    Loads.Add(NodalLoad);
  except
    raise Exception.CreateFmt('Error in reading nodal load %d', [No]);
  end;
end;
end;
end;
finally
  StrParser.Free;
  CloseFile(Inf);
end;
end;
end.

```



## A.9 Veri Dosyası Yapısı

Analizi gerçekleştirilecek yapının programa tanıtılması serbest formatta hazırlanan bir ASCII veri dosyası aracılığı ile yapılmaktadır. Veri dosyasının ilk satırı problemin başlığı olup veri dosyasındaki bölümleri birbirinden ayırarak daha okunaklı hale getirmek üzere boş satırlar kullanılabilir. Herhangi bir satırın birinci kolonuna C harfi koyarak ve bunu takip eden ikinci kolonu da boş bırakarak kullanıcılara yönelik uyarı ve açıklayıcı bilgiler de veri dosyasına yerleştirilebilir.

Veri dosyası SYSTEM, NODES, RESTRAINTS, MATERIALS, SECTIONS, FRAME, TRUSS, CONNECTION, ve LOADS bölümleri olmak üzere başlıca 9 ana bölüm ve bunlara bağlı alt bölümlerden oluşmaktadır.

### SYSTEM

METHOD=m NITER=ni NSTEP=ns [A=a1,a2]

m: çözüm yöntemi. (0=Linear, 1=Newton-Raphson, 2=Modified Newton-Raphson, 3=Modified Newton-Raphson with Step Loading, 4=Arc-Length )

ni: İterasyon sayısı

ni: Yükleme adımı sayısı

a1,a2: Betonarme arayüz elemanı için  $\alpha_c$  ve  $\alpha_s$  parametreleri. (otomatik değerler 0.15 ve 0.20 dir)

### NODES

no x y

no: Düğüm numarası

x,y: Düğüm noktası koordinatları

## RESTRAINTS

no R= r<sub>1</sub>, r<sub>2</sub>, r<sub>3</sub>, r<sub>4</sub>, r<sub>5</sub>, r<sub>6</sub>

no: Dügüm numarası

r<sub>1</sub>-r<sub>6</sub>: Serbestlik kodu (0=serbest, 1=tutulu)

## MATERIALS

no CONCRETE=c

veya

no NP=np G=g NAME=name

no: Malzeme tip numarası

c: Beton dayanımı ( $f_c$ )

np: Malzeme davranış eğrisinin tanımı için gerekli nokta sayısı

g: Kesme modülü

name: Malzeme adı

np adet nokta için gerilme, birim deformasyon değerleri bu satırı takip etmelidir.

## SECTIONS

no A=a I= i<sub>x</sub>, i<sub>y</sub>, i<sub>z</sub>

no: Kesit numarası

i<sub>x</sub>, i<sub>y</sub>, i<sub>z</sub>: X, Y ve Z eksenleri etrafındaki kesit atalet momentleri (düzlem çerçeve sistemler için i<sub>x</sub> ve i<sub>y</sub> sıfırdır)

## FRAME ve TRUSS

no  $n_1$   $n_2$  M=m S=s

no: Çerçeve eleman numarası

$n_1, n_2$ : Eleman uç düğün noktaları

m: Malzeme no.

s: Kesit no.

## CONNECTION

no  $n_1$   $n_2$  MCON=mc MST=ms RE=  $r_1, r_2$  W=w H=h NL= $l_c, l_s$  ST= $d_1, a_1, d_2, a_2, \dots$  TIE= $a_t, s$

no: Betonarme arayüz elemanı numarası

$n_1, n_2$ : Eleman uç düğün noktaları

mc: Beton malzeme no.

ms: Çelik malzeme no.

$r_1, r_2$ : Eleman uçlarındaki rijit bölge genişlikleri

w: Kesit genişliği

h: Kesit yüksekliği

$l_c$ : Beton katman adedi

$l_s$ : Boyuna donatı çubukları seviye adedi

$a_t$ : Etriye tek kolunun kesit alanı (program halihazırda etriyelerin iki kollu olduğunu vrsaymaktadır)

s: Etriye aralığı

$d_1, a_1, \dots$ : Kesit merkezinden herbir boyuna donatı seviyesine olan uzaklık ve bu seviyedeki toplam donatı alanı

## LOADS

no F=  $f_x, f_y, f_z, m_x, m_y, m_z$

no: Yüklü düğüm noktası no.

$f_x, f_y, f_z, m_x, m_y, m_z$ : Yapısal eksenler doğrultusundaki yük ve moment değerleri

## EK B

# DENEYSEL ÇERÇEVE ANALİZLERİ VERİ VE ÇIKTI DOSYALARI

### B.1 Veri Dosyası

Nonlinear Plane Frame with Semi-Rigid Ends  
SYSTEM  
METHOD=4 NITER=100 NSTEP=10

#### NODES

1 0.0 0.0  
2 3.5 0.0  
3 0.0 2.2  
4 3.5 2.2  
5 0.0 4.2  
6 3.5 4.2  
7 0.0 0.6  
8 0.0 1.6  
9 3.5 0.6  
10 3.5 1.6  
11 0.6 2.2  
12 2.9 2.2  
13 0.0 2.8  
14 0.0 3.6  
15 3.5 2.8  
16 3.5 3.6  
17 0.6 4.2  
18 2.9 4.2

#### RESTRAINTS

1 R=1,1,1,1,1,1  
2 R=1,1,1,1,1,1  
3 R=0,0,1,1,1,0  
4 R=0,0,1,1,1,0  
5 R=0,0,1,1,1,0  
6 R=0,0,1,1,1,0  
7 R=0,0,1,1,1,0  
8 R=0,0,1,1,1,0  
9 R=0,0,1,1,1,0  
10 R=0,0,1,1,1,0  
11 R=0,0,1,1,1,0  
12 R=0,0,1,1,1,0  
13 R=0,0,1,1,1,0  
14 R=0,0,1,1,1,0  
15 R=0,0,1,1,1,0  
16 R=0,0,1,1,1,0  
17 R=0,0,1,1,1,0  
18 R=0,0,1,1,1,0

#### MATERIALS

1 CONCRETE=30

```

2 NP=4 G=80000000.0 NAME=STEEL
  0.0 0.0
  0.00217 418000.0
  0.0093 418000.0
  0.06672 596000.0

```

```

SECTIONS
1 A=0.12 I=0.0,0.0,0.0

```

```

FRAME
1 7 8 M=1 S=1
2 9 10 M=1 S=1
3 11 12 M=1 S=1
4 13 14 M=1 S=1
5 15 16 M=1 S=1
6 17 18 M=1 S=1

```

```

CONNECTION
7 1 7 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
8 3 8 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
9 2 9 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
10 4 10 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
11 3 11 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
12 4 12 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
13 3 13 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
14 5 14 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
15 4 15 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
16 6 16 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
17 5 17 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125
18 6 18 MCON=1 MST=2 RE=0.2,0 W=0.3 H=0.4 NL=10,2 ST=0.15,0.0012,-0.15,0.0012
TIE=0.0001,0.125

```

```

LOADS
5 F=500.0,-700.0
6 F=0.0,-700.0

```

## B.2 Çıktı Dosyası

Nonlinear Plane Frame with Semi-Rigid Ends

```

SYSTEM DATA
*****

```

```

Number of nodal points = 18
Number of elements     = 19
Number of sections     = 1
Number of materials    = 2

```

```

NODAL POINT DATA
*****

```

NODE	X-COORD	Y-COORD	Z-COORD	DX	DY	DZ	RX	RY	RZ
1	0.00	0.00	0.00	1	1	1	1	1	1
2	3.50	0.00	0.00	1	1	1	1	1	1
3	0.00	2.20	0.00	0	0	1	1	1	0
4	3.50	2.20	0.00	0	0	1	1	1	0
5	0.00	4.20	0.00	0	0	1	1	1	0
6	3.50	4.20	0.00	0	0	1	1	1	0
7	0.00	0.60	0.00	0	0	1	1	1	0

8	0.00	1.60	0.00	0	0	1	1	1	0
9	3.50	0.60	0.00	0	0	1	1	1	0
10	3.50	1.60	0.00	0	0	1	1	1	0
11	0.60	2.20	0.00	0	0	1	1	1	0
12	2.90	2.20	0.00	0	0	1	1	1	0
13	0.00	2.90	0.00	0	0	1	1	1	0
14	0.00	3.60	0.00	0	0	1	1	1	0
15	3.50	2.90	0.00	0	0	1	1	1	0
16	3.50	3.60	0.00	0	0	1	1	1	0
17	0.60	4.20	0.00	0	0	1	1	1	0
18	2.90	4.20	0.00	0	0	1	1	1	0

ELEMENT SECTION TYPES

NO	NAME	AREA(A)	INERTIA(Ixx)	INERTIA(Iyy)	INERTIA(Izz)
1	Section1	1.200E+01	0.000E+00	0.000E+00	1.600E-03

ELEMENT MATERIAL TYPES

NO	NAME	MOD. OF ELAST.	MOD. OF SHEAR
1	BS30	2.649E+07	8.000E+07
2	STEEL	1.926E+08	8.000E+07

ELEMENT DEFINITIONS

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
1	FRAME	7	9	Section1	BS30	0.00
2	FRAME	9	10	Section1	BS30	0.00
3	FRAME	11	12	Section1	BS30	0.00
4	FRAME	13	14	Section1	BS30	0.00
5	FRAME	15	16	Section1	BS30	0.00
6	FRAME	17	18	Section1	BS30	0.00
7	CONN	1	7			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	9	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
8	CONN	3	9			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	9	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
9	CONN	2	3			
Internal elements with internal nodes						
1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
10	CONN	4	10			
Internal elements with internal nodes						
1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
11	CONN	3	11			
Internal elements with internal nodes						
1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
12	CONN	4	12			
Internal elements with internal nodes						
1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120



12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
13	CONN	3	13			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
14	CONN	5	14			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
15	CONN	4	15			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
16	CONN	6	16			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011

6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
17	CONN	5	17			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
18	CONN	6	18			

Internal elements with internal nodes

1	TRUSS	5	6	Conc.Layer	BS30	0.01011
2	TRUSS	7	8	Conc.Layer	BS30	0.01011
3	TRUSS	9	10	Conc.Layer	BS30	0.01011
4	TRUSS	11	12	Conc.Layer	BS30	0.01011
5	TRUSS	13	14	Conc.Layer	BS30	0.01011
6	TRUSS	15	16	Conc.Layer	BS30	0.01011
7	TRUSS	17	18	Conc.Layer	BS30	0.01011
8	TRUSS	19	20	Conc.Layer	BS30	0.01011
9	TRUSS	21	22	Conc.Layer	BS30	0.01011
10	TRUSS	23	24	Conc.Layer	BS30	0.01011
11	TRUSS	25	26	Steel1	STEEL	0.00120
12	TRUSS	27	28	Steel2	STEEL	0.00120
13	TRUSS	1	4	Bracing1	BS30	0.01800
14	TRUSS	1	4	Bracing2	STEEL	0.00006
15	TRUSS	3	2	Bracing3	BS30	0.01800
16	TRUSS	3	2	Bracing4	STEEL	0.00006

ELEMENT	TYPE	NODE_I	NODE_J	SECTION	MATERIAL	ALPHA
---------	------	--------	--------	---------	----------	-------

NODAL LOADS  
\*\*\*\*\*

NODE	FORCE X	FORCE Y	FORCE Z	MOMENT X	MOMENT Y	MOMENT Z
5	500.00	-700.00	0.00	0.00	0.00	0.00
6	0.00	-700.00	0.00	0.00	0.00	0.00

\*\*\*\*\*  
NONLINEAR SOLUTION USING ARC LENGTH METHOD  
\*\*\*\*\*

LOAD STEP 1

Converged at 24th iteration

NODAL POINT DISPLACEMENTS  
\*\*\*\*\*

NODE	X-DISP	Y-DISP	Z-DISP	X-ROT	Y-ROT	Z-ROT
1	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000

2	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
3	1.4541E-003	-9.8214E-008	0.0000E+000	0.0000E+000	0.0000E+000	-8.0885E-004
4	1.5450E-003	-6.5448E-005	0.0000E+000	0.0000E+000	0.0000E+000	-8.3960E-004
5	3.2033E-003	-3.0431E-005	0.0000E+000	0.0000E+000	0.0000E+000	-5.6303E-004
6	3.1955E-003	-1.2062E-004	0.0000E+000	0.0000E+000	0.0000E+000	-4.7877E-004
7	1.5121E-004	1.7086E-005	0.0000E+000	0.0000E+000	0.0000E+000	-5.4507E-004
8	9.1730E-004	4.3613E-006	0.0000E+000	0.0000E+000	0.0000E+000	-8.7149E-004
9	1.6018E-004	-2.9746E-006	0.0000E+000	0.0000E+000	0.0000E+000	-5.4666E-004
10	9.7399E-004	-4.8779E-005	0.0000E+000	0.0000E+000	0.0000E+000	-9.3128E-004
11	1.5011E-003	-2.7872E-004	0.0000E+000	0.0000E+000	0.0000E+000	-5.7557E-005
12	1.4968E-003	2.2678E-004	0.0000E+000	0.0000E+000	0.0000E+000	-7.3565E-005
13	2.0146E-003	-7.1761E-006	0.0000E+000	0.0000E+000	0.0000E+000	-9.5120E-004
14	2.7643E-003	-1.4080E-005	0.0000E+000	0.0000E+000	0.0000E+000	-8.3404E-004
15	2.1077E-003	-7.9088E-005	0.0000E+000	0.0000E+000	0.0000E+000	-9.2805E-004
16	2.8121E-003	-1.0871E-004	0.0000E+000	0.0000E+000	0.0000E+000	-7.5217E-004
17	3.2144E-003	-2.4552E-004	0.0000E+000	0.0000E+000	0.0000E+000	-1.1685E-004
18	3.1912E-003	6.0474E-005	0.0000E+000	0.0000E+000	0.0000E+000	-1.1033E-004

ELEMENT END FORCES IN LOCAL DIRECTIONS

ELEM	NODE1	F1	F2	F3	F4	F5	F6
	NODE2	F1	F2	F3	F4	F5	F6
1	4.3292E+001	2.9396E+001	0.0000E+000	0.0000E+000	0.0000E+000	2.8528E+001	
	-4.3292E+001	-2.9396E+001	0.0000E+000	0.0000E+000	0.0000E+000	8.6784E-001	
2	1.4555E+002	3.8047E+001	0.0000E+000	0.0000E+000	0.0000E+000	3.5319E+001	
	-1.4555E+002	-3.8047E+001	0.0000E+000	0.0000E+000	0.0000E+000	2.7274E+000	
3	6.0008E+000	-2.7424E+001	0.0000E+000	0.0000E+000	0.0000E+000	-3.1242E+001	
	-6.0008E+000	2.7424E+001	0.0000E+000	0.0000E+000	0.0000E+000	-3.1832E+001	
4	7.0716E+001	3.8306E+001	0.0000E+000	0.0000E+000	0.0000E+000	7.9536E+000	
	-7.0716E+001	-3.8306E+001	0.0000E+000	0.0000E+000	0.0000E+000	2.0363E+001	
5	1.1812E+002	3.2046E+001	0.0000E+000	0.0000E+000	0.0000E+000	3.5033E+000	
	-1.1812E+002	-3.2046E+001	0.0000E+000	0.0000E+000	0.0000E+000	2.2133E+001	
6	3.2046E+001	-2.3704E+001	0.0000E+000	0.0000E+000	0.0000E+000	-2.7379E+001	
	-3.2046E+001	2.3704E+001	0.0000E+000	0.0000E+000	0.0000E+000	-2.7139E+001	
7	Connection Element						
7	4.3291E+001	2.9396E+001	0.0000E+000	0.0000E+000	0.0000E+000	4.6167E+001	
	-4.3291E+001	-2.9396E+001	0.0000E+000	0.0000E+000	0.0000E+000	-2.8529E+001	

Axial Stresses of Internal Elements: (Tension(+))

ELEM	TYPE	STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	0.000E+000	2.903E-004	2.648E+007 Cracked
2	Conc.Layer	0.000E+000	2.358E-004	2.648E+007 Cracked
3	Conc.Layer	1.440E+003	1.913E-004	2.648E+007
4	Conc.Layer	1.763E+003	1.268E-004	2.648E+007
5	Conc.Layer	1.399E+003	7.229E-005	2.648E+007
6	Conc.Layer	3.480E+002	1.778E-005	2.648E+007
7	Conc.Layer	-9.646E+002	-3.672E-005	2.648E+007
8	Conc.Layer	-2.867E+003	-9.123E-005	2.648E+007
9	Conc.Layer	-3.735E+003	-1.457E-004	2.648E+007
10	Conc.Layer	-5.069E+003	-2.002E-004	2.648E+007
11	Steel1	4.805E+004	2.494E-004	1.926E+008
12	Steel2	-3.070E+004	-1.534E-004	1.926E+008
13	Bracing1	1.443E+003	7.527E-005	2.648E+007
14	Bracing2	1.450E+004	7.527E-005	1.926E+008
15	Bracing3	-7.944E+002	-3.020E-005	2.648E+007
16	Bracing4	-5.817E+003	-3.020E-005	1.926E+008
8	Connection Element			
8	4.3291E+001	2.9396E+001	0.0000E+000	0.0000E+000 0.0000E+000 1.8505E+001
	-4.3291E+001	-2.9396E+001	0.0000E+000	0.0000E+000 0.0000E+000 -8.6791E-001

Axial Stresses of Internal Elements: (Tension(+))

ELEM	TYPE	STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	3.350E+002	1.748E-005	2.648E+007
2	Conc.Layer	2.160E+002	1.121E-005	2.648E+007
3	Conc.Layer	9.488E+001	4.349E-006	2.648E+007
4	Conc.Layer	-3.480E+001	-1.315E-006	2.648E+007
5	Conc.Layer	-2.003E+002	-7.579E-006	2.648E+007
6	Conc.Layer	-3.654E+002	-1.384E-005	2.648E+007
7	Conc.Layer	-5.301E+002	-2.011E-005	2.648E+007
8	Conc.Layer	-6.942E+002	-2.637E-005	2.648E+007
9	Conc.Layer	-8.579E+002	-3.263E-005	2.648E+007
10	Conc.Layer	-1.021E+003	-3.890E-005	2.648E+007
11	Steel1	2.462E+003	1.278E-005	1.926E+008
12	Steel2	-6.898E+003	-3.420E-005	1.926E+008
13	Bracing1	8.338E+002	4.349E-005	2.648E+007
14	Bracing2	8.378E+003	4.349E-005	1.926E+008

15 Bracing3	-1.409E+003	-5.385E-005	2.648E+007
16 Bracing4	-1.037E+004	-5.385E-005	1.926E+008
9 Connection Element			
9	1.4554E+002	3.8047E+001	0.0000E+000 0.0000E+000 0.0000E+000 5.8148E+001
	-1.4554E+002	-3.8047E+001	0.0000E+000 0.0000E+000 0.0000E+000 -3.5320E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	0.000E+000	2.386E-004	2.648E+007 Cracked
2 Conc.Layer	1.434E+003	1.940E-004	2.648E+007
3 Conc.Layer	1.749E+003	1.293E-004	2.648E+007
4 Conc.Layer	1.431E+003	7.464E-005	2.648E+007
5 Conc.Layer	3.830E+002	1.998E-005	2.648E+007
6 Conc.Layer	-9.115E+002	-3.469E-005	2.648E+007
7 Conc.Layer	-2.319E+003	-8.935E-005	2.648E+007
8 Conc.Layer	-3.692E+003	-1.440E-004	2.648E+007
9 Conc.Layer	-5.031E+003	-1.987E-004	2.648E+007
10 Conc.Layer	-6.334E+003	-2.534E-004	2.648E+007
11 Steel1	3.807E+004	1.976E-004	1.926E+008
12 Steel2	-4.090E+004	-2.124E-004	1.926E+008
13 Bracing1	1.148E+003	5.990E-005	2.648E+007
14 Bracing2	1.154E+004	5.990E-005	1.926E+008
15 Bracing3	-1.754E+003	-6.723E-005	2.648E+007
16 Bracing4	-1.295E+004	-6.723E-005	1.926E+008
10 Connection Element			
10	1.4555E+002	3.8047E+001	0.0000E+000 0.0000E+000 0.0000E+000 2.5556E+001
	-1.4555E+002	-3.8047E+001	0.0000E+000 0.0000E+000 0.0000E+000 -2.7274E+000

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	-2.901E-002	-7.558E-010	2.648E+007
2 Conc.Layer	-2.423E+002	-9.169E-006	2.648E+007
3 Conc.Layer	-4.836E+002	-1.934E-005	2.648E+007
4 Conc.Layer	-7.140E+002	-2.751E-005	2.648E+007
5 Conc.Layer	-9.633E+002	-3.668E-005	2.648E+007
6 Conc.Layer	-1.292E+003	-4.584E-005	2.648E+007
7 Conc.Layer	-1.498E+003	-5.501E-005	2.648E+007
8 Conc.Layer	-1.678E+003	-6.419E-005	2.648E+007
9 Conc.Layer	-1.911E+003	-7.335E-005	2.648E+007
10 Conc.Layer	-2.145E+003	-8.252E-005	2.648E+007
11 Steel1	-1.325E+003	-6.877E-006	1.926E+008
12 Steel2	-1.457E+004	-7.564E-005	1.926E+008
13 Bracing1	7.820E+002	4.079E-005	2.648E+007
14 Bracing2	7.857E+003	4.079E-005	1.926E+008
15 Bracing3	-2.123E+003	-8.166E-005	2.648E+007
16 Bracing4	-1.573E+004	-8.166E-005	1.926E+008
11 Connection Element			
11	6.0131E+000	-2.7419E+001	0.0000E+000 0.0000E+000 0.0000E+000 -4.7694E+001
	-6.0131E+000	2.7419E+001	0.0000E+000 0.0000E+000 0.0000E+000 3.1243E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	-5.555E+003	-2.205E-004	2.648E+007
2 Conc.Layer	-3.726E+003	-1.454E-004	2.648E+007
3 Conc.Layer	-1.931E+003	-7.024E-005	2.648E+007
4 Conc.Layer	9.379E+001	4.892E-006	2.648E+007
5 Conc.Layer	1.534E+003	8.002E-005	2.648E+007
6 Conc.Layer	1.600E+003	1.552E-004	2.648E+007
7 Conc.Layer	9.000E+000	2.303E-004	2.648E+007 Cracked
8 Conc.Layer	9.000E+000	3.054E-004	2.648E+007 Cracked
9 Conc.Layer	9.000E+000	3.905E-004	2.648E+007 Cracked
10 Conc.Layer	9.000E+000	4.557E-004	2.648E+007 Cracked
11 Steel1	-3.162E+004	-1.641E-004	1.926E+008
12 Steel2	7.692E+004	3.993E-004	1.926E+008
13 Bracing1	-3.097E+002	-1.173E-005	2.648E+007
14 Bracing2	-2.259E+003	-1.173E-005	1.926E+008
15 Bracing3	1.748E+003	1.294E-004	2.648E+007
16 Bracing4	2.493E+004	1.294E-004	1.926E+008
12 Connection Element			
12	6.0385E+000	2.7413E+001	0.0000E+000 0.0000E+000 0.0000E+000 4.8280E+001
	-6.0385E+000	-2.7413E+001	0.0000E+000 0.0000E+000 0.0000E+000 -3.1832E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	0.000E+000	4.655E-004	2.648E+007 Cracked
2 Conc.Layer	0.000E+000	3.889E-004	2.648E+007 Cracked
3 Conc.Layer	0.000E+000	3.123E-004	2.648E+007 Cracked
4 Conc.Layer	0.000E+000	2.357E-004	2.648E+007 Cracked

5	Conc.Layer	1.577E+003	1.591E-004	2.648E+007		
6	Conc.Layer	1.581E+003	9.248E-005	2.648E+007		
7	Conc.Layer	1.127E+002	9.881E-006	2.648E+007		
8	Conc.Layer	-1.843E+003	-7.072E-005	2.648E+007		
9	Conc.Layer	-3.774E+003	-1.473E-004	2.648E+007		
10	Conc.Layer	-5.637E+003	-2.239E-004	2.648E+007		
11	Steel1	7.860E+004	4.080E-004	1.926E+008		
12	Steel2	-3.207E+004	-1.665E-004	1.926E+008		
13	Bracing1	1.725E+003	1.333E-004	2.648E+007		
14	Bracing2	2.569E+004	1.333E-004	1.926E+008		
15	Bracing3	-3.286E+002	-1.244E-005	2.648E+007		
16	Bracing4	-2.397E+003	-1.244E-005	1.926E+008		
13	Connection Element					
13	7.0715E+001	3.5396E+001	0.0000E+000	0.0000E+000	0.0000E+000	2.9192E+001
	-7.0715E+001	-3.5396E+001	0.0000E+000	0.0000E+000	0.0000E+000	-7.9537E+000

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE		STRESS	STRAIN	M.OF ELAS.		
1	Conc.Layer	9.983E-002	4.686E-005	2.648E+007		
2	Conc.Layer	6.254E-002	3.263E-005	2.648E+007		
3	Conc.Layer	3.529E+002	1.839E-005	2.648E+007		
4	Conc.Layer	7.965E+001	4.155E-006	2.648E+007		
5	Conc.Layer	-2.663E-002	-1.008E-005	2.648E+007		
6	Conc.Layer	-6.404E-002	-2.432E-005	2.648E+007		
7	Conc.Layer	-1.012E+003	-3.855E-005	2.648E+007		
8	Conc.Layer	-1.381E+003	-5.279E-005	2.648E+007		
9	Conc.Layer	-1.749E+003	-6.702E-005	2.648E+007		
10	Conc.Layer	-2.113E+003	-8.126E-005	2.648E+007		
11	Steel1	6.977E-003	3.618E-005	1.926E+008		
12	Steel2	-1.363E-004	-7.059E-005	1.926E+008		
13	Bracing1	9.582E+002	4.999E-005	2.648E+007		
14	Bracing2	9.629E+003	4.999E-005	1.926E+008		
15	Bracing3	-1.743E+003	-6.680E-005	2.648E+007		
16	Bracing4	-1.287E+004	-6.680E-005	1.926E+008		
14	Connection Element					
14	7.0717E+001	3.5396E+001	0.0000E+000	0.0000E+000	0.0000E+000	4.1601E+001
	-7.0717E+001	-3.5396E+001	0.0000E+000	0.0000E+000	0.0000E+000	-2.0363E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE		STRESS	STRAIN	M.OF ELAS.		
1	Conc.Layer	1.967E+003	1.087E-004	2.648E+007		
2	Conc.Layer	1.563E+003	9.156E-005	2.648E+007		
3	Conc.Layer	1.044E+003	5.446E-005	2.648E+007		
4	Conc.Layer	5.244E+002	2.736E-005	2.648E+007		
5	Conc.Layer	4.895E+000	2.554E-007	2.648E+007		
6	Conc.Layer	-7.067E+002	-2.685E-005	2.648E+007		
7	Conc.Layer	-1.411E+003	-5.395E-005	2.648E+007		
8	Conc.Layer	-2.108E+003	-8.105E-005	2.648E+007		
9	Conc.Layer	-2.795E+003	-1.081E-004	2.648E+007		
10	Conc.Layer	-3.475E+003	-1.352E-004	2.648E+007		
11	Steel1	1.702E-004	8.833E-005	1.926E+008		
12	Steel2	-2.214E+004	-1.149E-004	1.926E+008		
13	Bracing1	9.994E+002	5.213E-005	2.648E+007		
14	Bracing2	1.004E+004	5.213E-005	1.926E+008		
15	Bracing3	-1.701E+003	-6.519E-005	2.648E+007		
16	Bracing4	-1.256E+004	-6.519E-005	1.926E+008		
15	Connection Element					
15	1.1812E+002	3.2046E+001	0.0000E+000	0.0000E+000	0.0000E+000	2.2731E+001
	-1.1812E+002	-3.2046E+001	0.0000E+000	0.0000E+000	0.0000E+000	-3.5034E+000

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE		STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	1.257E+002	6.556E-006	2.648E+007
2	Conc.Layer	-6.059E+001	-2.289E-006	2.648E+007
3	Conc.Layer	-2.941E+002	-1.113E-005	2.648E+007
4	Conc.Layer	-5.268E+002	-1.998E-005	2.648E+007
5	Conc.Layer	-7.585E+002	-2.883E-005	2.648E+007
6	Conc.Layer	-9.893E+002	-3.767E-005	2.648E+007
7	Conc.Layer	-1.219E+003	-4.652E-005	2.648E+007
8	Conc.Layer	-1.448E+003	-5.536E-005	2.648E+007
9	Conc.Layer	-1.676E+003	-6.421E-005	2.648E+007
10	Conc.Layer	-1.903E+003	-7.305E-005	2.648E+007
11	Steel1	-1.502E+001	-7.797E-008	1.926E+008
12	Steel2	-1.279E+004	-6.642E-005	1.926E+008
13	Bracing1	6.736E+002	3.514E-005	2.648E+007
14	Bracing2	6.762E+003	3.514E-005	1.926E+008
15	Bracing3	-1.774E+003	-6.800E-005	2.648E+007

```

16 Bracing4      -1.31E+004      -6.800E-005      1.926E+008
16 Connection Element
16  1.1812E+002  3.2046E-001  0.0000E+000  0.0000E+000  0.0000E+000  4.1361E+001
    -1.1812E+002 -3.2046E-001  0.0000E+000  0.0000E+000  0.0000E+000 -2.2133E+001
Axial Stresses of Internal Elements: (Tension(+))
ELEM TYPE      STRESS      STRAIN      M.OF ELAS.
 1 Conc.Layer    1.738E+003    3.353E-005    2.648E+007
 2 Conc.Layer    1.283E+003    5.819E-005    2.648E+007
 3 Conc.Layer    7.447E+002    3.895E-005    2.648E+007
 4 Conc.Layer    2.206E+002    1.151E-005    2.648E+007
 5 Conc.Layer   -4.177E+002   -1.584E-005    2.648E+007
 6 Conc.Layer   -1.117E+003   -1.817E-005    2.648E+007
 7 Conc.Layer   -1.008E+003   -1.761E-005    2.648E+007
 8 Conc.Layer   -1.007E+003   -1.768E-005    2.648E+007
 9 Conc.Layer   -6.126E+002   -1.253E-004    2.648E+007
10 Conc.Layer   -6.126E+002   -1.253E-004    2.648E+007
11 Steel1       1.147E+004    7.302E-005    1.926E+008
12 Steel2      -2.348E+004   -1.320E-004    1.926E+008
13 Bracing1     7.124E+002    3.716E-005    2.648E+007
14 Bracing2     7.124E+002    3.716E-005    1.926E+008
15 Bracing3    -1.734E+003   -8.647E-005    2.648E+007
16 Bracing4    -1.27E+004    -8.647E-005    1.926E+008
17 Connection Element
17  3.2038E+001 -1.3704E-001  0.0000E+000  0.0000E+000  0.0000E+000 -4.1602E+001
    -3.2038E+001  1.3704E-001  0.0000E+000  0.0000E+000  0.0000E+000  2.7380E+001
Axial Stresses of Internal Elements: (Tension(+))
ELEM TYPE      STRESS      STRAIN      M.OF ELAS.
 1 Conc.Layer    1.147E+003   -1.725E-004    2.648E+007
 2 Conc.Layer    1.147E+003   -1.725E-004    2.648E+007
 3 Conc.Layer    1.147E+003   -1.725E-004    2.648E+007
 4 Conc.Layer    1.147E+003   -1.725E-004    2.648E+007
 5 Conc.Layer    1.147E+003   5.523E-006    2.648E+007
 6 Conc.Layer    7.124E+002    3.716E-005    2.648E+007
 7 Conc.Layer    7.124E+002    3.716E-005    2.648E+007
 8 Conc.Layer    7.124E+002    3.716E-005    2.648E+007
 9 Conc.Layer    7.124E+002    3.716E-005    2.648E+007
10 Conc.Layer    7.124E+002    3.716E-005    2.648E+007 Cracked
11 Steel1      -1.334E+004   -1.334E-004    1.926E+008
12 Steel2     1.334E+004    1.334E-004    1.926E+008
13 Bracing1    -7.124E+002   -3.716E-005    2.648E+007
14 Bracing2    -7.124E+002   -3.716E-005    1.926E+008
15 Bracing3     1.734E+003    8.647E-005    2.648E+007
16 Bracing4     1.27E+004    8.647E-005    1.926E+008
18 Connection Element
18  3.2023E+001  2.3704E-001  0.0000E+000  0.0000E+000  0.0000E+000  4.1364E+001
    -3.2023E+001 -2.3704E-001  0.0000E+000  0.0000E+000  0.0000E+000 -2.7142E+001
Axial Stresses of Internal Elements: (Tension(+))
ELEM TYPE      STRESS      STRAIN      M.OF ELAS.
 1 Conc.Layer    1.477E+003    1.765E-004    2.648E+007
 2 Conc.Layer    1.669E+003    1.397E-004    2.648E+007
 3 Conc.Layer    1.001E+003    1.029E-004    2.648E+007
 4 Conc.Layer    1.266E+002    6.599E-005    2.648E+007
 5 Conc.Layer    6.33E+001     2.915E-005    2.648E+007
 6 Conc.Layer   -2.183E+001   -7.636E-006    2.648E+007
 7 Conc.Layer   -1.113E+003   -4.454E-005    2.648E+007
 8 Conc.Layer   -1.113E+003   -4.454E-005    2.648E+007
 9 Conc.Layer   -1.113E+003   -4.454E-005    2.648E+007
10 Conc.Layer   -5.388E+002   -1.551E-004    2.648E+007
11 Steel1     1.480E+004    1.480E-004    1.926E+008
12 Steel2    -1.480E+004   -1.480E-004    1.926E+008
13 Bracing1    4.528E+001    4.528E-005    2.648E+007
14 Bracing2    4.528E+001    4.528E-005    1.926E+008
15 Bracing3    1.522E+001    1.522E-005    2.648E+007
16 Bracing4    1.522E+001    1.522E-005    1.926E+008

```

Load steps 2 to 4 are given in short form.

LOAD STEP 2

Converged at 37th iteration

NODAL POINT DISPLACEMENTS

\*\*\*\*\*

NODE	X-DISE	Y-DISE	Z-DISE	X-ROT	Y-ROT	Z-ROT
1	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
2	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
3	4.0039E-003	-8.0121E-003	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
4	4.3577E-003	-7.7000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.1401E-003
5	8.8929E-003	-1.5400E-002	0.0000E+000	0.0000E+000	0.0000E+000	-2.2838E-003
6	9.0375E-003	-1.6000E-002	0.0000E+000	0.0000E+000	0.0000E+000	-1.5208E-003
7	5.7210E-003	-1.2000E-002	0.0000E+000	0.0000E+000	0.0000E+000	-1.5764E-003
8	2.6157E-003	-9.1800E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.5898E-003
9	5.0411E-003	-6.7800E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.2628E-003
10	2.8412E-003	-2.7400E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.6892E-003
11	4.1830E-003	-4.8800E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.4879E-003
12	4.1720E-003	-5.7374E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.3636E-004
13	5.4815E-003	-6.0000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.1230E-004
14	7.5634E-003	-8.1300E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.5562E-003
15	5.9645E-003	-6.0000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.4609E-003
16	7.8530E-003	-8.1300E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.5318E-003
17	9.0920E-003	-8.0000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-2.2849E-003
18	8.2346E-003	-7.0000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.6552E-004

LOAD STEP 3

Converged at 315th iteration

NODAL POINT DISPLACEMENTS

NODE	X-DISE	Y-DISE	Z-DISE	X-ROT	Y-ROT	Z-ROT
1	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
2	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
3	9.5411E-003	-1.8771E-002	0.0000E+000	0.0000E+000	0.0000E+000	-5.2575E-003
4	1.0672E-002	-3.5841E-003	0.0000E+000	0.0000E+000	0.0000E+000	-5.7794E-003
5	2.1779E-002	-1.0505E-002	0.0000E+000	0.0000E+000	0.0000E+000	-3.6375E-003
6	2.2159E-002	-1.2771E-002	0.0000E+000	0.0000E+000	0.0000E+000	-3.3850E-003
7	1.2894E-002	-2.3877E-003	0.0000E+000	0.0000E+000	0.0000E+000	-3.6794E-003
8	6.0634E-003	-2.4200E-003	0.0000E+000	0.0000E+000	0.0000E+000	-5.3853E-003
9	1.3800E-002	-1.5000E-002	0.0000E+000	0.0000E+000	0.0000E+000	-4.0765E-003
10	6.7824E-003	-2.6800E-003	0.0000E+000	0.0000E+000	0.0000E+000	-6.0628E-003
11	1.0084E-002	-1.1400E-003	0.0000E+000	0.0000E+000	0.0000E+000	-3.3912E-005
12	1.0062E-002	-1.3840E-003	0.0000E+000	0.0000E+000	0.0000E+000	-6.3952E-005
13	1.3386E-002	-1.2061E-003	0.0000E+000	0.0000E+000	0.0000E+000	-6.5176E-003
14	1.8570E-002	-1.5840E-003	0.0000E+000	0.0000E+000	0.0000E+000	-6.0790E-003
15	1.4513E-002	-1.5000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-6.1861E-003
16	1.9314E-002	-1.7000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-5.4873E-003
17	2.2022E-002	-1.2000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-4.4287E-004
18	2.1025E-002	-1.2000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-3.1139E-004

LOAD STEP 4

Converged at 30th iteration

NODAL POINT DISPLACEMENTS

NODE	X-DISE	Y-DISE	Z-DISE	X-ROT	Y-ROT	Z-ROT
1	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
2	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
3	1.9829E-002	-1.1700E-002	0.0000E+000	0.0000E+000	0.0000E+000	-1.0877E-002
4	2.3002E-002	-5.4500E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.1985E-002
5	4.3698E-002	-3.4100E-003	0.0000E+000	0.0000E+000	0.0000E+000	-8.7690E-003
6	4.5646E-002	-3.4100E-003	0.0000E+000	0.0000E+000	0.0000E+000	-7.7679E-003
7	2.2557E-002	-3.3600E-003	0.0000E+000	0.0000E+000	0.0000E+000	-8.7229E-003
8	1.2990E-002	-8.5700E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.0853E-002
9	3.4426E-002	-8.5700E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.0314E-002
10	1.5226E-002	-8.4000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.2457E-002
11	2.1354E-002	-8.5000E-003	0.0000E+000	0.0000E+000	0.0000E+000	9.0360E-004
12	2.1311E-002	-8.5000E-003	0.0000E+000	0.0000E+000	0.0000E+000	8.9601E-004
13	2.7237E-002	-8.1000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.2434E-002
14	3.7129E-002	-8.7000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.1857E-002
15	3.0543E-002	-8.5400E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.2108E-002
16	3.0809E-002	-8.5000E-003	0.0000E+000	0.0000E+000	0.0000E+000	-1.0747E-002
17	4.4749E-002	-1.2000E-003	0.0000E+000	0.0000E+000	0.0000E+000	2.3742E-004
18	4.4645E-002	-1.2000E-003	0.0000E+000	0.0000E+000	0.0000E+000	2.4476E-004

LOAD STEP 5

Converged at 35th iteration

NODAL JOINT DISPLACEMENTS

NODE	X-DISP	Y-DISP	Z-DISP	X-ROT	Y-ROT	Z-ROT
1	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
2	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000
3	4.3000E-001	1.7000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.6680E-002
4	5.3500E-001	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.9556E-002
5	1.4700E-001	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.5988E-002
6	1.1540E-001	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.4401E-002
7	9.2040E-002	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.3867E-002
8	3.3500E-001	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.6409E-002
9	2.4900E-001	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.9015E-002
10	3.3810E-001	1.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.4401E-002
11	5.4060E-001	-2.1000E-001	0.0000E+000	0.0000E+000	0.0000E+000	3.6478E-003
12	5.3200E-001	-1.2700E-001	0.0000E+000	0.0000E+000	0.0000E+000	3.5922E-003
13	6.7000E-001	-1.7700E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.9232E-002
14	9.0810E-002	-1.3700E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.9042E-002
15	7.6670E-002	-1.1400E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.9429E-002
16	9.0650E-002	-1.2000E-001	0.0000E+000	0.0000E+000	0.0000E+000	-2.7675E-002
17	1.1190E-001	-1.9700E-001	0.0000E+000	0.0000E+000	0.0000E+000	2.9805E-003
18	1.1170E-001	-1.7000E-001	0.0000E+000	0.0000E+000	0.0000E+000	3.0110E-003

ELEMENT END FORCES IN GLOBAL DIRECTIONS

ELEM	NODE1	F1	F2	F3	F4	F5	F6
	NODE2	F1	F2	F3	F4	F5	F6
1	2.3061E+002	1.1811E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	1.6427E+002
		-2.3061E+002	-1.1811E+002	0.0000E+000	0.0000E+000	0.0000E+000	-5.1138E+001
2	7.4412E+002	2.3400E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	1.9031E+002
		-7.4412E+002	-2.3400E+002	0.0000E+000	0.0000E+000	0.0000E+000	4.4681E+001
3	1.0220E+002	-1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	-1.4603E+002
		-1.0220E+002	1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	-1.4807E+002
4	3.5840E+002	1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	7.6374E+001
		-3.5840E+002	-1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	9.6475E+001
5	5.1800E+002	1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	-4.0112E+001
		-5.1800E+002	-1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	1.4576E+002
6	1.3200E+002	-1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	-1.4878E+002
		-1.3200E+002	1.1000E+002	0.0000E+000	0.0000E+000	0.0000E+000	-1.4766E+002
7	Connection Element						
7	2.3061E+002	1.1811E+002	0.0000E+000	0.0000E+000	0.0000E+000	0.0000E+000	2.3214E+002
		-2.3061E+002	-1.1811E+002	0.0000E+000	0.0000E+000	0.0000E+000	-1.6427E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM	TYPE	STRESS	STRAIN	M.OF ELAS.		
1	Conc.Layer	0.000E+000	1.954E-002	2.648E+007 Cracked		
2	Conc.Layer	0.000E+000	1.616E-002	2.648E+007 Cracked		
3	Conc.Layer	0.000E+000	1.377E-002	2.648E+007 Cracked		
4	Conc.Layer	0.000E+000	1.138E-002	2.648E+007 Cracked		
5	Conc.Layer	0.000E+000	9.090E-003	2.648E+007 Cracked		
6	Conc.Layer	0.000E+000	6.612E-003	2.648E+007 Cracked		
7	Conc.Layer	0.000E+000	4.226E-003	2.648E+007 Cracked		
8	Conc.Layer	0.000E+000	1.900E-003	2.648E+007 Cracked		
9	Conc.Layer	-1.100E-004	-5.469E-004	2.648E+007		
10	Conc.Layer	-1.100E-004	-2.013E-003	2.648E+007		
11	Steel1	1.600E+005	1.600E-002	3.100E+006		
12	Steel1	-1.100E+005	-1.100E-003	1.926E+008		
13	Bracing1	0.000E+000	0.164E-003	2.648E+007 Cracked		
14	Bracing2	0.000E+000	0.164E-003	0.000E+000		
15	Bracing3	-1.100E-004	-1.100E-004	2.648E+007		
16	Bracing4	-1.100E-004	-1.100E-004	1.926E+008		
8	Connection Element					
8	2.3061E+002	1.1811E+002	0.0000E+000	0.0000E+000	0.0000E+000	1.6738E+001
		-2.3061E+002	-1.1811E+002	0.0000E+000	0.0000E+000	5.1139E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM	TYPE	STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	-4.008E+003	-1.600E-004	2.648E+007
2	Conc.Layer	-3.414E+003	-1.320E-004	2.648E+007
3	Conc.Layer	-2.737E+003	-1.058E-004	2.648E+007
4	Conc.Layer	-2.059E+003	-7.878E-005	2.648E+007



5	Conc.Layer	-1.394E-003	-7.171E-005	2.648E+007
6	Conc.Layer	-1.394E-003	-7.171E-005	2.648E+007
7	Conc.Layer	-1.394E-003	-7.171E-005	2.648E+007
8	Conc.Layer	-1.394E-003	-7.171E-005	2.648E+007
9	Conc.Layer	-1.394E-003	-7.171E-005	2.648E+007
10	Conc.Layer	-1.394E-003	-7.171E-005	2.648E+007
11	Steel1	-1.394E-003	-7.171E-005	1.926E+008
12	Steel2	-1.394E-003	-7.171E-005	1.926E+008
13	Bracing1	0.000E+000	0.488E-004	2.648E+007 Cracked
14	Bracing2	1.394E-003	0.488E-004	1.926E+008
15	Bracing3	-8.350E-004	-3.350E-004	2.648E+007
16	Bracing4	-6.370E-004	-3.350E-004	1.926E+008
9	Connection Element			
9	7.4406E+002	0.0000E+000	0.0000E+000	0.0000E+000 3.3130E+002
	-7.4406E+002	0.0000E+000	0.0000E+000	0.0000E+000 -1.9030E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	0.000E+000	2.648E+007 Cracked
2	Conc.Layer	0.000E+000	2.648E+007 Cracked
3	Conc.Layer	0.000E+000	2.648E+007 Cracked
4	Conc.Layer	0.000E+000	2.648E+007 Cracked
5	Conc.Layer	0.000E+000	2.648E+007 Cracked
6	Conc.Layer	0.000E+000	2.648E+007 Cracked
7	Conc.Layer	0.000E+000	2.648E+007 Cracked
8	Conc.Layer	0.000E+000	2.648E+007 Cracked
9	Conc.Layer	0.000E+000	2.648E+007
10	Conc.Layer	0.000E+000	2.648E+007 Cracked
11	Steel1	0.000E+000	3.100E+006
12	Steel2	0.000E+000	0.000E+000
13	Bracing1	0.000E+000	2.648E+007 Cracked
14	Bracing2	0.000E+000	0.000E+000
15	Bracing3	-7.441E-004	2.648E+007
16	Bracing4	-7.441E-004	1.926E+008
10	Connection Element		
10	7.4414E+002	0.0000E+000	0.0000E+000 1.8567E+002
	-7.4414E+002	0.0000E+000	0.0000E+000 -4.4680E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	0.000E+000	2.648E+007 Cracked
2	Conc.Layer	0.000E+000	2.648E+007 Cracked
3	Conc.Layer	0.000E+000	2.648E+007
4	Conc.Layer	0.000E+000	2.648E+007
5	Conc.Layer	0.000E+000	2.648E+007
6	Conc.Layer	0.000E+000	2.648E+007
7	Conc.Layer	0.000E+000	2.648E+007
8	Conc.Layer	0.000E+000	2.648E+007
9	Conc.Layer	0.000E+000	2.648E+007
10	Conc.Layer	0.000E+000	2.648E+007
11	Steel1	0.000E+000	1.926E+008
12	Steel2	0.000E+000	1.926E+008
13	Bracing1	0.000E+000	2.648E+007 Cracked
14	Bracing2	0.000E+000	1.926E+008
15	Bracing3	-1.029E-003	2.648E+007
16	Bracing4	-1.029E-003	1.926E+008
11	Connection Element		
11	1.0294E+002	0.0000E+000	0.0000E+000 -2.2275E+002
	-1.0294E+002	0.0000E+000	0.0000E+000 1.4603E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1	Conc.Layer	-3.245E-003	2.648E+007
2	Conc.Layer	-3.245E-003	2.648E+007
3	Conc.Layer	0.000E+000	2.648E+007 Cracked
4	Conc.Layer	0.000E+000	2.648E+007 Cracked
5	Conc.Layer	0.000E+000	2.648E+007 Cracked
6	Conc.Layer	0.000E+000	2.648E+007 Cracked
7	Conc.Layer	0.000E+000	2.648E+007 Cracked
8	Conc.Layer	0.000E+000	2.648E+007 Cracked
9	Conc.Layer	0.000E+000	2.648E+007 Cracked
10	Conc.Layer	0.000E+000	2.648E+007 Cracked
11	Steel1	-3.707E-004	1.926E+008
12	Steel2	4.402E-005	3.100E+006
13	Bracing1	-3.400E-004	2.648E+007
14	Bracing2	-3.400E-004	1.926E+008
15	Bracing3	0.000E+000	2.648E+007 Cracked

16 Bracing4 4.197E+005 1.090E+002 3.100E+006  
 12 Connection Element  
 12 1.0294E+002 1.2797E+002 0.0000E+000 0.0000E+000 0.0000E+000 2.2480E+002  
 -1.0294E+002 -1.2797E+002 0.0000E+000 0.0000E+000 0.0000E+000 -1.4808E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	0.000E+000	1.531E-002	2.648E+007 Cracked
2 Conc.Layer	0.000E+000	1.209E-002	2.648E+007 Cracked
3 Conc.Layer	0.000E+000	1.068E-002	2.648E+007 Cracked
4 Conc.Layer	0.000E+000	1.836E-002	2.648E+007 Cracked
5 Conc.Layer	0.000E+000	1.305E-002	2.648E+007 Cracked
6 Conc.Layer	0.000E+000	0.783E-003	2.648E+007 Cracked
7 Conc.Layer	0.000E+000	0.418E-003	2.648E+007 Cracked
8 Conc.Layer	0.000E+000	0.103E-003	2.648E+007 Cracked
9 Conc.Layer	-0.000E+000	-2.010E-004	2.648E+007
10 Conc.Layer	-0.000E+000	-0.317E-003	2.648E+007
11 Steel1	1.090E+002	1.082E-002	3.100E+006
12 Steel2	-1.029E+002	-1.041E-003	1.926E+008
13 Bracing1	0.000E+000	1.181E-002	2.648E+007 Cracked
14 Bracing2	0.000E+000	1.181E-002	3.100E+006
15 Bracing3	-0.000E+000	-0.308E-004	2.648E+007
16 Bracing4	-0.000E+000	-0.308E-004	1.926E+008

13 Connection Element  
 13 3.5849E+002 2.1646E+002 0.0000E+000 0.0000E+000 0.0000E+000 2.0601E+002  
 -3.5849E+002 -2.1646E+002 0.0000E+000 0.0000E+000 0.0000E+000 -7.6374E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	0.000E+000	1.539E-003	2.648E+007 Cracked
2 Conc.Layer	0.000E+000	1.194E-003	2.648E+007 Cracked
3 Conc.Layer	0.000E+000	1.029E-003	2.648E+007 Cracked
4 Conc.Layer	0.000E+000	7.740E-004	2.648E+007 Cracked
5 Conc.Layer	0.000E+000	5.198E-004	2.648E+007 Cracked
6 Conc.Layer	0.000E+000	2.637E-004	2.648E+007 Cracked
7 Conc.Layer	1.044E+002	9.577E-006	2.648E+007
8 Conc.Layer	-0.000E+000	-2.466E-004	2.648E+007
9 Conc.Layer	-0.000E+000	-0.017E-004	2.648E+007
10 Conc.Layer	-0.000E+000	-7.569E-004	2.648E+007
11 Steel1	1.090E+002	1.084E-003	1.926E+008
12 Steel2	-1.029E+002	-0.055E-004	1.926E+008
13 Bracing1	0.000E+000	1.471E-003	2.648E+007 Cracked
14 Bracing2	0.000E+000	1.471E-003	1.926E+008
15 Bracing3	-0.000E+000	-0.303E-004	2.648E+007
16 Bracing4	-0.000E+000	-0.303E-004	1.926E+008

14 Connection Element  
 14 3.5849E+002 2.1646E+002 0.0000E+000 0.0000E+000 0.0000E+000 2.2611E+002  
 -3.5849E+002 -2.1646E+002 0.0000E+000 0.0000E+000 0.0000E+000 -9.6475E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	0.000E+000	1.858E-003	2.648E+007 Cracked
2 Conc.Layer	0.000E+000	1.553E-003	2.648E+007 Cracked
3 Conc.Layer	0.000E+000	1.247E-003	2.648E+007 Cracked
4 Conc.Layer	0.000E+000	0.418E-004	2.648E+007 Cracked
5 Conc.Layer	0.000E+000	6.365E-004	2.648E+007 Cracked
6 Conc.Layer	0.000E+000	3.311E-004	2.648E+007 Cracked
7 Conc.Layer	4.347E+002	2.581E-005	2.648E+007
8 Conc.Layer	-0.000E+000	-2.705E-004	2.648E+007
9 Conc.Layer	-1.343E+004	-5.249E-004	2.648E+007
10 Conc.Layer	-1.490E+004	-0.902E-004	2.648E+007
11 Steel1	0.000E+000	1.620E-003	1.926E+008
12 Steel2	-1.029E+002	-0.611E-004	1.926E+008
13 Bracing1	0.000E+000	1.549E-003	2.648E+007 Cracked
14 Bracing2	0.000E+000	1.549E-003	1.926E+008
15 Bracing3	-1.444E+004	-0.875E-004	2.648E+007
16 Bracing4	-1.444E+004	-0.875E-004	1.926E+008

15 Connection Element  
 15 6.1627E+002 1.0404E+002 0.0000E+000 0.0000E+000 0.0000E+000 3.9122E+001  
 -6.1627E+002 -1.0404E+002 0.0000E+000 0.0000E+000 0.0000E+000 4.0112E+001

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.
1 Conc.Layer	-0.000E+000	-2.203E-004	2.648E+007
2 Conc.Layer	-0.000E+000	-2.076E-004	2.648E+007
3 Conc.Layer	-4.046E+003	-1.949E-004	2.648E+007
4 Conc.Layer	-4.632E+003	-1.823E-004	2.648E+007
5 Conc.Layer	-4.322E+003	-1.696E-004	2.648E+007

6	Conc.Layer	-4.010E+003	-1.569E-004	2.648E+007		
7	Conc.Layer	-3.697E+003	-1.442E-004	2.648E+007		
8	Conc.Layer	-3.381E+003	-1.315E-004	2.648E+007		
9	Conc.Layer	-3.064E+003	-1.188E-004	2.648E+007		
10	Conc.Layer	-2.748E+003	-1.061E-004	2.648E+007		
11	Steel1	-4.061E+004	-2.108E-004	1.926E+008		
12	Steel2	-2.12E+004	-1.156E-004	1.926E+008		
13	Bracing1	0.000E+000	6.707E-004	2.648E+007	Cracked	
14	Bracing2	1.202E+005	6.707E-004	1.926E+008		
15	Bracing3	-2.648E+003	-3.993E-004	2.648E+007		
16	Bracing4	-7.692E+004	-3.993E-004	1.926E+008		
16 Connection Element						
16	6.1626E+002	1.3206E+002	0.0000E+000	0.0000E+000	0.0000E+000	2.2499E+002
	-6.1626E+002	-1.3206E+002	0.0000E+000	0.0000E+000	0.0000E+000	-1.4576E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.			
1	Conc.Layer	0.000E+000	1.717E-003	2.648E+007	Cracked	
2	Conc.Layer	0.000E+000	1.390E-003	2.648E+007	Cracked	
3	Conc.Layer	0.000E+000	1.062E-003	2.648E+007	Cracked	
4	Conc.Layer	0.000E+000	7.351E-004	2.648E+007	Cracked	
5	Conc.Layer	0.000E+000	4.079E-004	2.648E+007	Cracked	
6	Conc.Layer	1.544E+003	9.055E-005	2.648E+007		
7	Conc.Layer	-6.177E+003	-2.467E-004	2.648E+007		
8	Conc.Layer	-1.327E+004	-5.740E-004	2.648E+007		
9	Conc.Layer	-1.011E+004	-3.013E-004	2.648E+007		
10	Conc.Layer	-2.371E+004	-1.229E-003	2.648E+007		
11	Steel1	2.834E+005	1.471E-003	1.926E+008		
12	Steel2	-1.834E+005	-9.240E-004	1.926E+008		
13	Bracing1	0.000E+000	6.707E-004	2.648E+007	Cracked	
14	Bracing2	1.202E+005	6.707E-004	1.926E+008		
15	Bracing3	-2.648E+003	-3.993E-004	2.648E+007		
16	Bracing4	-7.692E+004	-3.993E-004	1.926E+008		
17 Connection Element						
17	1.3206E+002	-1.2889E+002	0.0000E+000	0.0000E+000	0.0000E+000	-2.2611E+002
	-1.3206E+002	1.2889E+002	0.0000E+000	0.0000E+000	0.0000E+000	1.4878E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.			
1	Conc.Layer	-2.734E+004	-3.173E-003	2.648E+007		
2	Conc.Layer	-6.866E+003	-2.761E-004	2.648E+007		
3	Conc.Layer	0.000E+000	2.621E-003	2.648E+007	Cracked	
4	Conc.Layer	0.000E+000	5.519E-003	2.648E+007	Cracked	
5	Conc.Layer	0.000E+000	9.414E-003	2.648E+007	Cracked	
6	Conc.Layer	0.000E+000	1.131E-002	2.648E+007	Cracked	
7	Conc.Layer	0.000E+000	1.421E-002	2.648E+007	Cracked	
8	Conc.Layer	0.000E+000	1.710E-002	2.648E+007	Cracked	
9	Conc.Layer	0.000E+000	2.000E-002	2.648E+007	Cracked	
10	Conc.Layer	0.000E+000	2.290E-002	2.648E+007	Cracked	
11	Steel1	-1.927E+005	-1.000E-003	1.926E+008		
12	Steel2	4.467E+005	2.073E-002	3.100E+006		
13	Bracing1	-9.405E+003	-3.434E-004	2.648E+007		
14	Bracing2	-6.615E+004	-3.434E-004	1.926E+008		
15	Bracing3	0.000E+000	1.027E-002	2.648E+007	Cracked	
16	Bracing4	4.17E+005	1.027E-002	3.100E+006		
18 Connection Element						
18	1.3206E+002	1.2889E+002	0.0000E+000	0.0000E+000	0.0000E+000	2.2499E+002
	-1.3206E+002	-1.2889E+002	0.0000E+000	0.0000E+000	0.0000E+000	-1.4766E+002

Axial Stresses of Internal Elements: (Tension(+))

ELEM TYPE	STRESS	STRAIN	M.OF ELAS.		
1	Conc.Layer	0.000E+000	2.165E-002	2.648E+007	Cracked
2	Conc.Layer	0.000E+000	1.891E-002	2.648E+007	Cracked
3	Conc.Layer	0.000E+000	1.617E-002	2.648E+007	Cracked
4	Conc.Layer	0.000E+000	1.343E-002	2.648E+007	Cracked
5	Conc.Layer	0.000E+000	1.069E-002	2.648E+007	Cracked
6	Conc.Layer	0.000E+000	7.947E-003	2.648E+007	Cracked
7	Conc.Layer	0.000E+000	5.206E-003	2.648E+007	Cracked
8	Conc.Layer	0.000E+000	2.465E-003	2.648E+007	Cracked
9	Conc.Layer	-6.876E+003	-2.765E-004	2.648E+007	
10	Conc.Layer	-2.779E+004	-3.018E-003	2.648E+007	
11	Steel1	4.432E+005	1.960E-002	3.100E+006	
12	Steel2	-1.853E+005	-9.618E-004	1.926E+008	
13	Bracing1	0.000E+000	9.714E-003	2.648E+007	Cracked
14	Bracing2	4.180E+005	9.714E-003	3.100E+006	
15	Bracing3	-8.405E+003	-3.434E-004	2.648E+007	
16	Bracing4	-6.615E+004	-3.434E-004	1.926E+008	

LOAD STEP 6  
Maximum number of iterations reached

LOAD STEP 7  
Maximum number of iterations reached

LOAD STEP 8  
Maximum number of iterations reached

LOAD STEP 9  
Maximum number of iterations reached

LOAD STEP 10  
Maximum number of iterations reached

## KAYNAKLAR

1. **G.R.Monforton, and T.S.Wu**, “*Matrix Analysis of Semi-Rigidly Connected Steel Frames*”, ASCE Structural Division, Vol. 89, pp. 13-42, 1963.
2. **Ö.Yüncü**, A Finite Element Based Reinforced Concrete Interface Element, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1998
3. **F.J.Vecchio, and M.P.Collins**, “*Predicting the Response of Reinforced Concrete Beams Subjected to Shear Using Modified Compression Field Theory*”, ACI Structural Journal, Vol. 85, pp. 258-268, 1988.
4. **W.Ritter**, “*Die Bauweise Hennebique*”, Schweizerische Bauzeitung (Zürich), Feb. 1899.
5. **E.Mörsch**, Concrete Steel Construction (Der Eisenbetonbau), McGraw-Hill Book Co., New York, 1909, 368p.
6. **U.Ersoy**, “*Reinforced Concrete*”, Middle East Technical University, 1994.
7. **F.J.Vecchio, and M.B.Emara**, “*Shear Deformations in Reinforced Concrete Frames*”, ACI Structural Journal, Vol. 89, No. 1, pp. 46-56, 1992.
8. **M.U.Polat, H.B.Bahat, A.H.Argeşo**, “*Düzlem Kabuk Yapısal Sistemlerin Analizi İçin Bir Panel Makro Elemanı*”, III.Ulusal Hesaplamalı Mekanik Konferansı, İstanbul, pp.203-210, 16-18 Kasım 1998
9. **A.H.Argeşo**, A Panel Macro Element for the Analysis of Structural Wall Systems, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1996
10. **H.B.Bahat**, Analysis of Structural Wall Systems, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1998

11. **J.H.Stricklin, W.Haisler, P.Tisdale, K.Gunderson**, "A Rapidly Converging Triangular Plate Element", *JAIJA Journal*, **7**, pp.180-181, 1969.
12. **J.L.Batoz, K.J.Bathe, and L.W.Ho**, "A Study of Three-Node Triangular Plate Bending Elements", *Int. J. Num. Meth. Engng.*, **18**, pp.1771-1812, 1980.
13. **L.J.Batoz**, "An Explicit Formulation for an Efficient Plate Bending Element", *Int. J. Num. Meth. Engng.*, **21**, pp.1077-1089, 1982.
14. **C.Jeyachandrabose, C.Kirkhope, C.R.Babu**, "An Alternative Explicit Formulation for the DKT Plate-Bending Element", *Int. J. Num. Meth. Engng.*, **21**, pp.1289-1293, 1985.
15. **R.D.Cook, S.D.Malkus, and M.E.Plesha**, *Concepts and Applications of Finite Element Analysis*, Wiley, 1989.
16. **D.J.Allman**, "A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Analysis", *Computers and Structures*, **19**, pp.1-8, 1984.
17. **D.F. Watson**, "Computing the N-Dimensional Delunay Triangulation with Application to Voronoi Polytopes", *The Computer Journal*, **24**, pp.167-172, 1981.
18. **B.Joe and R.B.Simpson**, "Triangular Meshes for Regions of Complicated Shape", *Int. J. Num. Meth. Engng.*, **23**, pp.751-768, 1986.
19. **S.W.Sloan**, "A Fast Algorithm for Constructing Delunay Triangulations in Plane", *Adv. Eng. Software*, **9**, pp.34-55, 1987.
20. **O.C.Zienkiewicz and R.L.Taylor**, *The Finite Element Method, Volume II*, McGraw-Hill, 1991.
21. **A.Habibullah. and E.L.Wilson**, "SAP90-A General Structural Analysis Program User's Guide", *Computers and Structures, Inc., Berkely, California*, 1989.

22. **M.U.Polat, Ö.Kurç**, “*XSPlate: Sonlu Elemanlar Tabanlı Bir Plak İşlemcisi*”, III.Ulusal Hesaplamalı Mekanik Konferansı, İstanbul, pp.227-233, 16-18 Kasım 1998
23. **R.Szilar**, *Theory and Analysis of Plates*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

BİBLİYOGRAFİK BİLGİ FORMU	
1.Proje No: INTAG 542	2. Rapor Tarihi: 15 Aralık 1999
3.Projenin Başlangıç ve Bitiş Tarihleri: 1 Kasım 1994 – 30 Nisan 1998	
4.Projenin Adı: Prefabrik Çerçeve ve Sünek Duvarlardan Oluşan Yapı Sistemlerinin Deprem Davranışının Analitik İncelenmesi	
5.Proje Yürütücüsü ve Yardımcı Araştırmacılar: Prof. Ş.M.Üzümeri Doç.Dr.Uğur Polat Prof.Dr.Uğur Ersoy Prof.Dr.Güney Özcebe	
6.Projenin Yürütüldüğü Kuruluş ve Adresi: Orta Doğu Teknik Üniversitesi, İnşaat Müh. Bölümü	
7.Destekleyen Kuruluşların Adı ve Adresi: Orta Doğu Teknik Üniversitesi, Ankara, Turkey University of Toronto, Toronto, Ontario, Canada	
<p>8. Öz (Abstract): Uygulamada uzay çerçeve yapısal taşıyıcı sistemlerin analizleri genellikle birleşimleri için yapılan rijit birleşim veya mafsallı birleşim kabulü ile gerçekleştirilmektedir. Bu proje kapsamında, prefabrik çerçeve sistemler için pek de gerçekçi ve geçerli olmadığı bilinen bu basitleştirmeyi gereksiz kılmak ve bu tür yapısal sistemlerin analizlerini daha gerçekçi olarak gerçekleştirebilmek için gerekli olan, bu tür sistemlerin yarı rijit birleşimlerini modellemek üzere bir <i>Elasto-Plastik Betonarme Arayüz Elemanı</i> geliştirilmiştir.</p> <p>Boşluklu ve boşluksuz perde duvar ve döşemeler gibi iki boyutlu elemanlar içeren yapısal sistemlerin sonlu elemanlar yöntemi ile analizlerinde karşılaşılan modelleme, bilgisayar kapasitesi ve yüksek hacimdeki tepki çıktılarının tasarına temel teşkil edecek tepkiler haline dönüştürülmesi aşamalarında karşılaşılan güçlükleri gidermek üzere bir <i>Düzlem Panel Makro (Süper) Elemanı</i> geliştirilmiştir.</p> <p>Yapısal sistemlerin düzgün olmayan bir geometriye sahip ve/veya düzensiz yüklemeye maruz döşemelerinin analizlerini sonlu elemanlar yöntemi ile gerçekçi olarak gerçekleştirebilmek, sonuçları tasarımda direkt olarak kullanılabilir hale getirmek ve bu yatay elemanlardan çerçeve taşıyıcı sisteme aktarılan yükleri sağlıklı bir şekilde belirleyebilmek üzere bir Sonlu Elemanlar Tabanlı Plak İşlemcisi bilgisayar yazılımı geliştirilmiştir</p> <p><b>Anahtar Kelimeler:</b> Yarı-rijit birleşim, betonarme arayüz elemanı, Panel Makro Elemanı</p>	
9.Proje ile ilgili Yayın ve Tebliğler: <ul style="list-style-type: none"> <li>• <b>Ö.Yüncü</b>, <i>A Finite Element Based Reinforced Concrete Interface Element</i>, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1998</li> <li>• <b>M.U.Polat, Ö.Kurç</b>, "<i>XSPlate: Sonlu Elemanlar Tabanlı Bir Plak İşlemcisi</i>", III.Ulusal Hesaplamalı Mekanik Konferansı, İstanbul, pp.227-233, 16-18 Kasım 1998</li> <li>• <b>M.U.Polat, H.B.Bahat, A.H.Argeşo</b>, "<i>Düzlem Kabuk Yapısal Sistemlerin Analizi İçin Bir Panel Makro Elemanı</i>", III.Ulusal Hesaplamalı Mekanik Konferansı, İstanbul, pp.203-210, 16-18 Kasım 1998</li> <li>• <b>A.H.Argeşo</b>, <i>A Panel Macro Element for the Analysis of Structural Wall Systems</i>, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1996</li> <li>• <b>H.B.Bahat</b>, <i>Analysis of Structural Wall Systems</i>, M.Sc. Tezi, İnşaat Mühendisliği Bölümü, ODTÜ, 1998</li> </ul>	
10. Bilim Dalı: <p>Doçentlik B.Dalı Kodu:</p> <p>Uzmanlık Alanı Kodu:</p>	
11. Dağıtım: Sınırsız	
12.Raporun Gizlilik Durumu: Gizli Değil	