YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK KATLI YÜKSEK YAPILARA AİT ORTOGONAL KAT ÇERÇEVELERİNİN TRANSFER MATRİS METODUYLA ÇÖZÜMÜ

YÜKSEK LİSANS TEZİ

İNŞ.MÜH.ERKAN ALTAY

TEZİ YÖNETEN
PROF.SİNAN ÇAĞDAŞ

İSTANBUL.1992

İÇİNDEKİLER:

ÖZET

Çerçeve çözümlerini bilgisayar ile yapmak için çeşitli metodlar kullanılmaktadır. Bu metodlarda, bilinmeyen sayısı çerçevenin birleşim noktaları ve kat sayısına bağlıdır. Bilgisayarların hafıza sınırları içerisinde çalışması, büyük çerçevelerin analizleri için küçük kapasiteli bilgisayarların kullanımını güçleştirmektedir.

Bu tezde küçük bellekli bilgisayarlar ile çok katlı büyük çerçeveleri çözmek için yeni bir metod geliştirilmiştir.

Transfer matris kavramından faydalanılıp, kolon ve kat transfer matrisleri formüle edilmiştir. Kat transfer matrislerinin ardışık çarpımları ile tavan kat birleşim değerleri mesnet değerleri cinsinden ifade edilmiştir.

Bilinen tavan kat ve mesnet değerleri bu ifadede kullanılarak bilinmeyen tavan kat birleşim değerleri hesaplanmıştır. Bulunan değerler kolon ve kat transfer matrisleri ile çarpılarak çerçevenin çözümü tamamlanmıştır.

İşlemlerde kullanılan matrislerin boyutu diğer metodlarda olduğu gibi kat sayısı ile değil bir kattaki açıklık sayısı ile orantılıdır.

Erkan Altay

ABSTRACT

Various methods are used for the structural frame analysis by computers.In general,number of unknowns used in these methods is the function of the number of joints and joint restraints.The size of an orthogonal frame that can be analysed by a small memory capacity of such machines and due to the number of unknowns to be solved by using these methods.

This thesis aims to develop a method of analysis for planar orthogonal frames to lessen the memory need of the computer.

The development of the method is based on transfer matrix concept.Columb and story transfer matrices are derived.By chan matrix multiplication of story transfer matrices sequentially, state vector for the top floor is related to the state vector for the supports.By substituting the known conditions for the elements of the state vectors into the transfer matrix relation, a set of linear equations is obtained in terms of unknowns of the support state vector End forces and end rotations df all of the members of the frame are calculated by first solving these equations and then, by chain multiplication of story transfer matrices starting with the support state vector.

The computer memory needed for the calculations in this method is independent of the number of stories.

Erkan Altay

ÇOK KATLI YÜKSEK YAPILARA AİT ORTOGONAL KAT ÇERÇEVELERİNİN TRANSFER MATRİS METODUYLA ÇÖZÜMÜ

1.GİRİŞ:

Bu tez kapsamında,ortogonal bir yapı sistemine ait çok katlı çerçevelerin çözümünde,kat sayısının fazlalığından doğacak zaman ve hesap güçlüğünü en aza indirgemek amacıyla computer destekli metod ve programı üzerinde çalışma yer almaktadır.

Çalışmada,yapı sistemlerinin çözümünde programlamaya en yatkın yol olan matris metotlarının avantajları kullanılarak,diferansiyel ifadesi mümkün olan birçok yapı sistemlerini çözebilecek "Geçiş Matrisleri Metodu"nun çok katlı ortogonal kat çerçevelerinin çözümünde kullanılması amçlanmıştır.

Bu konu kapsamında önce yaygın olarak sürekli sistemlerin çözümünde kullanılan metodu,ana prensiplerini ortaya koyarak farklı bir modelleştirme ile kat çerçevelerine adaptesi yapılacaktır.

Geçiş matrisleri metodu kullanılarak çok katlı yüksek yapılara ait ortogonal çerçevelerin çözümünde diğer yollara nazaran,computer bellek kullanım sınırlarını zorlamadan en kısa zamanda,en az hata ve emek ile çözüme ulaşılması sağlanacaktır.

Ayrıca bu yolla,çözümün aşama aşama takip edilebiliyor olması ve kesit tesirleri ile birlikte deplasmanlarında çözüm neticesinde aynı anda elde edilebilmesi mümkün olacaktır.

Metodun uygulanmasında,ortogonal çok katlı yapı sistemlerini sürekli sistem halinde, çerçeveyi ardışık birer kolon grubu ve kirişler dizisi olarak gruplayıp modelleştirilmesi yapılacaktır.Oluşturulan yeni modele ait kat matrislerinin transfer matris mantığıyla ardışık çarpımlarını kullanarak çözüme ulaşılması amaç edinilmiştir.Bunun içinde her kataait kolonların sistematik gruplanması ve kirişleri için de her çubuğun ardışık çarpımlarının oluşturulması gerekecektir.

Bu amaçla çalışma kapsamında önce sürekli kirişlerin geçiş matrisleri metoduyla çözümü ele alınarak ana prensipler ortaya konulacak,bunlardan faydanılarak çerçeve sisteminde kullanılacak geçiş matrisleri ve ardışık çarpımları incelenecektir.Daha sonra çok katlı ortogonal çerçevelerin modelleştirilmesinde kullanılacak kolon gruplarının ve kirişler dizisinin oluşturumu ve bunların birer transfer matrisle ifadesi sağlanılacaktır.Geçiş matrisleri metoduyla modelleştirilmiş sistem üzerinde kat düzeyindeki deplasman ve kesit tesirlerini elde edeceğimiz çözüm sayısal olarak gerçeklenecektir.

Sistemde değişen parametreler göz önüne alınarak ardışık çarpımlar için gerekli transfer matrislerin oluşturumunu ve ardışık çarpımını içeren algoritma yardımıyla programlanması mümkün olabilecektir.

2.GEÇİŞ MATRİSLERİ METODU

Yük altındaki sürekli,çubuk,düzlemselve hacimsel sistemlerin diferansiyel ifadesi mümkün olabilen problemleri için geliştirilmiş bir metottur.

Mütemadi kirişler,elastik zemine oturan kirişler,dinamik etki altındaki sürekli sistemler ve 2.mertebe teorisi problemleri için oldukça elverişlidir.

Bu metodda küçük boyutlu matrisler ve az bilinmeyenli denklem takımıyla çalışılması ve hesaplamanın şematik olması büyük avantaj sağlamaktadır.

Sistemlerin çözümünde hiperstatiklik derecesi gözetilmemekte, gerilme ve yer değiştirmelerin aynı hesap içinde elde edilmesi mümkün olabilmektedir.

Aynı zamanda bu metod'un diferansiyel ifadesi mümkün olabilen bütün sistem problemlerinde kullanılabilir olması sebebiyle genis kapsamlı uygulanabilirlik alanı bulabilmektedir.Bu amaçla,aşağıda metod'un ana prensiplerinden yola çıkarak çok katlı ortagonal kat çerçeveleri için gerekli;oluşturulacak geçiş matrislerinin yardımıyla bu alanda avantajlı kullanılabilirliği gerçeklenecektir.

2.1 YAPILAN KABÜLLER:

a.Malzeme lineer elastik davranış gösterir;Hooke Kanunu geçerlidir.

b.Hesaplamalarda ikinci mertebe etkiler ihmal edilir. 1.Mertebe Teorisi geçerlidir.

c.Elamanlar doğru eksenli,sabit enkesitli düzlemsel çubuklardan oluşur.

d.Sisteme tesir eden yükler statik yüklerdir.

e.Süperpozisyon ilkesi geçerlidir.

f.Tez kapsamındaki çalışmada,kirişlerde normal kuvvet etkisi ihmal edilmiştir.

2.2 YÖNTEM ANA PRENSİPLERİ

Her düzlemsel çubuk eleman kendini belirleyici parametrelerle ifade edilir.Bu parametreler,üçü kuvvet ve diğer üçü deplasman büyüklüğü olmak üzere altı adettir.Bunlar,

- Kuvvet büyüklükleri:
  - Normal kuvvet (N)
  - Kesme kuvveti (Q)
  - Eğilme Momenti(M)
- Deplasman büyüklükleri:
  - Birim boy değişimi (u)
  - Çubuk eksenine dik deplasman (w)
  - Dönmeyi ifade eden (ø)'dir.

Düzlemsel çubuk için sol ve sağ olmak üzere başlangıç ve bitim noktalarında her birinde toplam 12 durum büyüklüğü mevcuttur.Herbir düğüm noktası durum büyüklüktlerini içeren durum vektöründe sınır şartlarına bağlı olarak altı parametreden üçü bilinir diğer üçü bilinmez.Farklı durumlarda sınır şartları aşağıdaki gibi değişiklik gösterir.

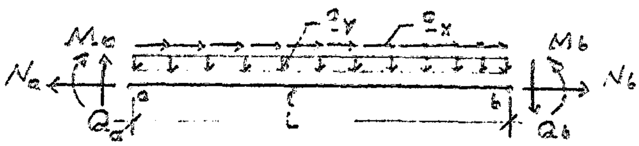
	Bilinenler	Bilinmeyenler
Tam ankastre	u=w=ø=0	N,Q,M
Serbest uç	N=Q=M=0	u,w.ø
Elastik mesnet	N=M=0.Q=Cw.W	u,w,ø
Basit mesnet	u=w=M=0	N,Q.ø

Sistemin tüm çubuklarında,sağındaki düğüm noktalarının durum büyüklükleri.solundaki durum büyüklükleri cinsinden ifade edilebilir.Bunun için kuvvet büyüklükleri denge koşulundan ve deplasman büyüklükleri kinematik uygunluk ve malzeme fleksibilite ilişkilerinden faydanılarak birbiri cinsinden belirlenebilir.Böylece yüklenmiş sistem,başlangıç ve son düğüm noktası arasında,mesnet.mafsal ve diğer ara şartsız çubuklara ayrılarak sağ durum büyüklükleri.sol durum büyüklükleri cinsinden ifade edilir.Bu işlem.sistemin sonuna kadar benzer şekilde uygulanarak son düğüm noktasındaki durum büyüklükleri,başlangıç düğüm noktasının bilinmeyen üç durum büyüklüğü bağıntısından oluşan denklem sisteminin çözümüyle elde edilir.

2.3.-İNTEGRAL ALMA YÖNTEMİYLE ÇUBUK ELEMAN GEÇİŞ MATRİSİNİN TEŞKİLİ

Denklem sistemini meydana getirecek düğüm noktası durum büyüklüklerinin diferansiyel ifadesi aşağıdaki şekilde oluşturulur.

İntegral alma yöntemine göre yüklü bir çubuk elemanda;



Denge denklemlerinden,

$$\frac{dN}{dx}=-q_x \quad , \quad \frac{dQ}{dx}=-q_y \; ; \quad \frac{dM}{dx}=Q$$

Geometrik uygunluk şartlarından,

$$\frac{dU}{dx}=\frac{N}{EF} \quad , \quad \frac{dw}{dx}=\phi \quad \frac{d\phi}{dx}=w$$

Malzeme kanunundan, $w=-\frac{M}{EI}$ bağıntıları yardımıyla

$$N=-\int q_x.dx+c_1 \quad , \quad Q=-\int q_y.dx+c_2 \quad , \quad M=\int (Q.dx+c_2)+c_3$$

$$ux=\int \frac{N}{EF}dx+c_4$$

$$=\int \frac{1}{EF}(-\int q_x.dx+c_1)dx+c_4$$

$ux=-\frac{1}{EF}\iint q_x.dx^2+c_1\int\frac{dx}{EF}+c_4$ bulunur. Benzer şekilde,

$$wx=\iint \frac{1}{EI}\iint q_y.dx^2-c_2\iint\frac{x}{EI}.dx^2-c_3\iint\frac{dx}{EI}+c_5.x+c_6$$

$$\phi x=\int \frac{1}{EI}\iint q_y.dx^2-c_2\int\frac{x}{EI}dx-c_3\int\frac{dx}{EI}+c_5$$

$$Nx=-\int q_x.dx+c_1$$

$$Qx=-\int q_y.dx+c_2$$

$Mx=-\iint q_y.dx^2+c_2.x+c_3$ bağıntıları elde edilir.

EI ve EF değerleri sabit, q_x ve q_y yükü üniform olarak alınırsa; integral sabitlerini bulabilmek için, x=0'da eldeki bağıntıların sınır değerlerinden, $c_1=Na$, $c_2=Qa$, $c_3=Ma$, $c_4=u_a$, $c_5=\phi_a$, $c_6=w_a$ olarak integral sabitleri elde edilir.

İntegral sabitleri bağıntılarında yerlerine konulup x=L için yazılırsa, b noktası durum büyüklükleri elde edilecektir.

$$u_b=u_a+N_a\cdot\frac{l}{EF}-q_x\cdot\frac{l^2}{2EF}$$

$$w_b=w_a+\phi_a\cdot l-Q_a\cdot\frac{l^3}{6EI}-M_a\cdot\frac{l^2}{2EI}+q_x\cdot\frac{l^4}{24EI}$$

$$\phi_b=\phi_a-Q_a\cdot\frac{l^2}{2EI}-M_a\cdot\frac{l}{EI}+q_y\cdot\frac{l^3}{6EI}$$

$$Nb=Na-q_x\cdot l$$

$$Qb=Qa-q_y\cdot l$$

$$Mb=Ma+Qa\cdot l-q_y\cdot\frac{l^2}{2}$$

Bu bağıntılar matrisyel olarak ifade edildiğinde,

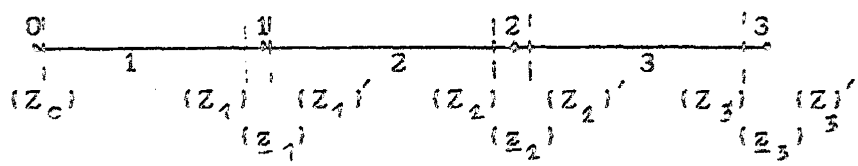
$$\begin{bmatrix} u_b \\ w_b \\ \phi_b \\ Nb \\ Qb \\ Mb \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & l/EF & 0 & 0 \\ 0 & 1 & l & 0 & -l^3/6EI & -l^2/2EI \\ 0 & 0 & 1 & 0 & -l^2/2EI & -l/EI \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & l & 1 \end{bmatrix} \cdot \begin{bmatrix} u_a \\ w_a \\ \phi_a \\ Na \\ Qa \\ Ma \end{bmatrix} + \begin{bmatrix} -q_x\cdot l^2/2EF \\ q_y\cdot l^4/24EI \\ q_y\cdot l^3/6EI \\ -q_x\cdot l \\ -q_y\cdot l \\ -q_y\cdot l^2/2 \end{bmatrix}$$

{Zb}=[u]i.{Za}+{Wb} formunda yazılabilir.

a ve b noktası durum büyüklükleri vektörü {Za} ve {Zb}, i çubuğu için yük vektörü {Wb}, eğilme +normal kuvvet etkisi altında çubuk elemanının kayma deformasyonları ihmal edilmiş haliyle çubuk geçiş matrisi [u]i belirlenmiş olur.

2.4. SÜREKLİ ÇUBUK SİSTEMLERDE GEÇİŞ MATRİSLERİ METODUNUN UYGULANMASI

Sürekli bir çubuk sisteminde; $\{Z\}$ durum vektörlerini, $[U]$ çubuk geçiş matrisini, $\{W\}$ yük vektörünü, $\{z\}$ ara şart durum vektörünü ifade ederse, son durum vektörünün ara şart bilinmeyenleri ve geçiş matrisleri cinsinden ifadesi şu şekilde yapılabilir:



$\{Z_1\}=[U_1].\{Z_0\}+\{w_1\}$

$\{Z_1\}'=\{Z_1\}+\{\underline{z}_1\}$

$\{Z_1\}'=[U_1].\{Z_0\}+\{w_1\}+\{\underline{z}_1\}$

$\{Z_2\}=[U_2].\{Z_1\}'+\{w_2\}$

$\{Z_2\}=[U_2].[U_1].\{Z_0\}+[U_2].\{w_1\}+[U_2].\{\underline{z}_1\}+\{w_2\}$

$\{Z_2\}'=\{Z_2\}+\{\underline{z}_2\}$

$\{Z_2\}'=[U_2].[U_1].\{Z_0\}+[U_2].\{w_1\}+[U_2].\{\underline{z}_1\}+\{w_2\}+\{\underline{z}_2\}$

$\{Z_3\}=[U_3].\{Z_2\}'+\{w_3\}$

$\{Z_3\}=[U_3].[U_2].[U_1].\{Z_0\}+[U_3].[U_2].\{w_1\}+[U_3].[U_2].\{\underline{z}_1\}+[U_3].\{w_2\}+[U_3].\{\underline{z}_2\}+\{w_3\}$ şeklinde daha fazla açıklıklar için bu ifade tarzı genişletilebilir. Burada son mesnet noktası sol durum büyüklükleri ile önceki ara şart noktalarının sağ durum vektörleri düzenlenerek.

$A_{31}=[U_3].[U_2].[U_1]$

$A_{32}=[U_3].[U_2]$

$A_{21}=[U_2].[U_1]$

$A_{33}=[U_3]$

$A_{11}=[U_1]$ şeklinde yazılarak son durum büyüklük vektörü ifadesi düzenlenirse;

$\{z_1\}'=[U_1].\{z_0\}+\{\underline{z}_1\}+\{w_1\}$

$\{z_2\}'=[U_2].[U_1].\{z_0\}+[U_2].\{\underline{z}_1\}+\{\underline{z}_2\}+[U_2].\{w_1\}+\{w_2\}$

$\{Z_3\}=[U_3].[U_2].[U_1].\{Z_0\}+[U_3].[U_2].\{\underline{z}_1\}+[U_3].\{\underline{z}_2\}+[U_3].[U_2].\{w_1\}+[U_3].\{w_2\}+\{w_3\}$

Yukarıdaki üç bağıntının matrisyel ifadesiyle,ara şart ve başlangıç bilinmeyenlerini içeren denklem sistemi elde edilmiş olacaktır.

$$\begin{bmatrix} z_1' \\ z_2' \\ Z_3 \end{bmatrix} = \begin{bmatrix} U_1 & I & 0 \\ U_2 \cdot U_1 & U_2 & I \\ U_3 \cdot U_2 \cdot U_1 & U_3 \cdot U_2 & U_3 \end{bmatrix} \cdot \begin{bmatrix} Z_0 \\ z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} I & 0 & 0 \\ U_2 & I & 0 \\ U_3 \cdot U_2 & U_3 & I \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Başlangıç durum vektörü ve ara şart bilinmeyenlerini içeren yukarıdaki üç denklem sisteminde,bilinmeyen sayısı kadar denklem elde edilebilmesi mümkündür.Bunun için $\{z_1'\}$, $\{z_2'\}$,$\{Z_3\}$,bilinmeyen vektörü içindeki sınır değerler çözüm için gerekli denklem takımını belirleyecektir.Çözüm için ayrıca başlangıç ve ara şart vektörlerininde tanımlanması gerekli olacaktır.

Durum vektörleri,nokta bilinmeyenlerini içeren bir katsayılar matrisi cinsinden ifade edilebilir.Buna göre;

$$\{Za\} = \begin{bmatrix} v & 0 & 0 & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & v \end{bmatrix} \cdot \begin{bmatrix} w \\ \varphi \\ Q \\ M \end{bmatrix}_a \text{'dir.}$$

Bu matriste v, nokta bilinmeyeni karşılığı veya ara şart karşılığı 1 veya 0 değerlerini alarak nokta tanımlaması yapılacaktır.

Benzer şekilde,sistemin sürekli çubuğuna ait tüm düğüm noktası değerlerini içerecek sistem durum vektörüde,sistem bilinmeyenlerini içeren sistem durum matrisi cinsinden yazılabilir.Böylece çözüm için gerekli olan $\{Zo\}$,$\{z_1\}$,$\{z_2\}$ ve diğer ara şart bilinmeyenlerini içeren durum matrisi oluşturulur.

$$\begin{bmatrix} z_1' \\ z_2' \\ Z_3 \end{bmatrix} = \begin{bmatrix} A_{11} & I & 0 \\ A_{21} & A_{22} & I \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \cdot \begin{bmatrix} vo & 0 & 0 \\ 0 & v_1 & 0 \\ 0 & 0 & v_2 \end{bmatrix} \cdot \begin{bmatrix} \hat{Z}o \\ \hat{z}_1 \\ \hat{z}_2 \end{bmatrix} + \begin{bmatrix} I & 0 & 0 \\ A_{22} & I & 0 \\ A_{32} & A_{33} & I \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Burada başlangıç durum vektörüne ait vo matrisi,başlangıç noktası bilinmeyenlerini içerecek,ara şart vektörüne ait v_1,v_2, matrisleride ilave denklem takımı elde etmek için ara şartları belirleyecektir.

İlave denklemler için;başlangıç noktası basit mesnet olması halinde vo:

$$\{Za\}=\begin{bmatrix} w \\ \emptyset \\ Q \\ M \end{bmatrix}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} wa \\ \emptyset a \\ Qa \\ Ma \end{bmatrix}=\begin{bmatrix} 0 \\ \emptyset a \\ Qa \\ 0 \end{bmatrix}=va.\{Za\}$$ olarak yazılır.

Ara şarttan faydanılarak yazılacak ilave denklemler için çökmenin sıfır olması ile ilave bilinmeyeni içeren durum matrisi şu şekilde yazılır.

$$\{Zb\}=\begin{bmatrix} w \\ \emptyset \\ Q \\ M \end{bmatrix}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} wb \\ \emptyset b \\ Qb \\ Mb \end{bmatrix}=vb.\{Zb\}$$

Ara mafsal şartı için ilave bilinmeyen ifadesi;Mc=0,øc=0

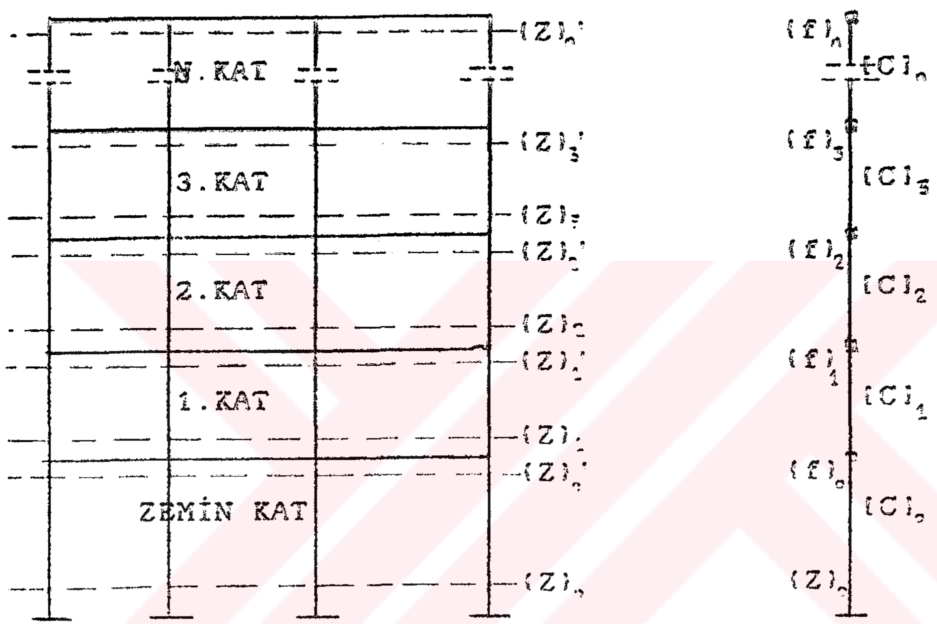
$$\{Zc\}=\begin{bmatrix} w \\ \emptyset \\ Q \\ M \end{bmatrix}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} wc \\ \emptyset c \\ Qc \\ Mc \end{bmatrix}=\begin{bmatrix} 0 \\ \emptyset c \\ 0 \\ 0 \end{bmatrix}=vc.\{Zc\}$$ olarak tanımlıdır.

$$\begin{bmatrix} M_1, Q_1, \emptyset_1, w_1 \\ \hline M_2, Q_2, \emptyset_2, w_2 \\ \hline M_3, Q_3, \emptyset_3, w_3 \end{bmatrix}=\left[\begin{array}{c:c:c} \underline{A}_{11}.\underline{Vo} & \underline{V}_1 & \underline{0} \\ \hdashline \underline{A}_{21}.\underline{Vo} & \underline{A}_{22}.\underline{V}_1 & \underline{V}_2 \\ \hdashline \underline{A}_{31}.\underline{Vo} & \underline{A}_{32}.\underline{V}_1 & \underline{A}_{33}.\underline{V}_2 \end{array}\right]\cdot\begin{bmatrix} \hat{Z}_0 \\ \hat{Z}_1 \\ \hat{Z}_2 \end{bmatrix}+\left[\begin{array}{c:c:c} \underline{I} & \underline{0} & \underline{0} \\ \hdashline \underline{A}_{22} & \underline{I} & \underline{0} \\ \hdashline \underline{A}_{32} & \underline{A}_{33} & \underline{I} \end{array}\right]\cdot\begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}$$

Yukarıdaki üç açıklıklı sürekli çubuk sistemindel2x12'lik matrisleri içinde yer alan ve başlangıç ve ara şart bilinmeyenleri kadar, $\{z_1\}'$, $\{z_2\}'$, $\{Z_3\}$ vektörlerinden oluşan çözüm vektörünün bilinen sınır değerlerinin satırındaki denklemlerden çözüme ulaşılır. Çözüm için $\{\hat{Z}o\}$, $\{\hat{z}_1\}$, $\{\hat{z}_2\}$ bilinmeyenlerini içeren denklem sistemi kullanılacak olup, aynı vektörlerinin bilinmeyen değerlerinden $\{z_1\}'$, $\{z_2\}'$, $\{Z_3\}$ bağıntıları yardımıyla tüm bilinmeyenler elde edilir.

Oluşturulacak sürekli sistemin çözüm geçiş matrisi ve yük matrisinin içinden $\{\hat{Z}o\}$, $\{\hat{z}_1\}'$, $\{\hat{z}_2\}'$ bilinmeyenlerini elde edebilecek denklem sistemini, $\{z_1\}'$, $\{z_2\}'$, $\{Z_3\}$ vektörü bilinen sınır şartları yardımıyla, otomatik tesbiti mümkün olacaktır. Bunun için v_1, v_2, v_3 tarifi yapılırken aynı zamanda oluşturulacak seçici matris bölüm 4'de tanımlanacaktır.

3.TRANSFER MATRİS METODUNUN ÇOK KATLI ORTOGONAL KAT ÇERÇEVELERİNDE UYGULANMASINA İLİŞKİN MODELLEŞTİRME



Kolon uç kuvvet ve deplasmanları içeren {Z}, kolon uçları durum vektörünü, herbir kolon çubuğunun geçiş matrislerinin açıklık sayısına göre genişletilerek oluşturulan yeni {C}, kolon geçiş matrisini, her kat düzeyindeki kiriş dizisini mütemadi çubuk sistem alarak, her kat'a ait çözüm vektörü olarak adlandırılacak {f},aynı zamanda her katın ait ve üst düzeyi kolon uçları parametreleri farkını ifade eder.

Transfer matris metodunun ardışık çarpımlar prensibinden yararlanılarak kat çerçevesini tek bir hat üzerindeki sürekli sisteme benzeterek yukarıdaki kat değerlerinin birbiri cinsinden ifadesi mümkün olabilecektir.

Bunun için gerekli olan,kat düzeyi kiriş sürekli sisteminin çözümü olan $\{f\}$ vektörü,her kata ait geçiş matrislerini ifade edecek $[C]$ matrisi ve bunların ardışık çarpımlarıyla ortaya çıkacak ve sonuçta elde edilecek katların alt ve üst düzeyindeki kolon uçları parametreleri; $\{Z\}$ vektörleri haricinde, en üst düzeye ait kolon uç kuvvetleri durum vektörü başlangıçta ayrı ayrı hesaplanacaktır.

Bununla birlikte dış yükler altındaki sistemin çözümü için aynı zamanda herhangi bir katın çözüm değerini veya tüm değerleri elde edecek ve sürekli bir algoritma dahilinde süre gelen hesap sistemi çerçevesinde çözüme ulaşabilecek hesaplamada metod gerekecektir.Bunun için de öncelikle ardışık çarpımların sistematik ifadesine gerek duyulmaktadır

Buna göre;

Her kat'ın alt ve üst düzeyi kolon uçları parametreleri farkı olan $\{f\}$durum vektörünün tanımından;son kat kolon üst düzeyi değerinin son kat kiriş sürekli sistemi$\{fn\}$ durum vektörüne eşit olduğu ortaya çıkmaktadır.

$\{Zn+1\}=0$

$\{Zn\}'-\{Zn+1\}=\{fn\} \implies \{Zn\}'=\{fn\}$ 3.1

-Dört katlı bir çerçevenin sistematik incelenmesi:

$\{Z_3\}'=\{f_3\}$ 3.1'den

$\{Z_3\} =[C_3].\{Z_3\}'$

$\{Z_2\}'=\{Z_3\}+\{f_2\}$

$\{Z_2\}'=[C_3].\{Z_3\}'+\{f_2\}$

$\{Z_2\} =[C_2].\{Z_2\}'$

$\{Z_2\} =[C_2].[C_3].\{Z_3\}'+[C_2].\{f_2\}$

$\{Z_1\}'=\{Z_2\}+\{f_1\}$

$\{Z_1\}'=[C_2].[C_3].\{Z_3\}'+[C_2].\{f_2\}+\{f_1\}$

$\{Z_1\} =[C_1].\{Z_1\}'$

$\{Z_1\} =[C_1].[C_2].[C_3].\{Z_3\}'+[C_1].[C_2].\{f_2\}+[C_1].\{f_1\}$

$\{Z_0\}'=\{Z_1\}+\{f_0\}$

$\{Z_0\}'=[C_1].[C_2].[C_3].\{Z_3\}'+[C_1].[C_2].\{f_2\}+[C_1].\{f_1\}+\{f_0\}$

$\{Z_0\} =[C_0].\{Z_0\}'$

$\{Z_0\} =[C_0].[C_1].[C_2].[C_3].\{Z_3\}'+[C_0].[C_1].[C_2].\{f_2\}+$

$+ [C_0].[C_1].\{f_1\}+[C_0].\{f_0\}$

$$[C_0].[C_1].[C_2].[C_3]=A_{03}$$
$$[C_0].[C_1].[C_2]=A_{02}$$
$$[C_0].[C_1]=A_{01}$$
$$[C_0]=A_{00}$$

$$\{Z_0\}=A_{03}.\{f_3\}+A_{02}.\{f_2\}+A_{01}.\{f_1\}+A_{00}.\{f_0\}$$

$$\{Z_0\}=\sum_{n=0}^{n=3} A_{0n}.\{f_n\}$$

$$\{Z_1\}=A_{13}.\{f_3\}+A_{12}.\{f_2\}+A_{11}.\{f_1\}$$
$$\{Z_2\}=A_{22}.\{f_2\}+A_{23}.\{f_3\}$$
$$\{Z_3\}=A_{33}.\{f_3\}$$

$$\{Z_m\}=\sum_{n=m}^{n=3} A_{mn}.\{f_n\}$$

$$[C_1].[C_2].[C_3]=A_{13}$$
$$[C_1].[C_2]=A_{12}$$
$$[C_2].[C_3]=A_{23}$$
$$[C_1]=A_{11}$$
$$[C_2]=A_{22}$$
$$[C_3]=A_{33}$$

Böylece kat kolon geçiş matrislerinin farklı çarpımlarının oluşan katsayılar ile kat düzeyi fark vektörlerinin çapımlarının oluşturduğu,kat adedi kadar denklem ortaya çıkmaktadır.

Ortaya çıkan denklemlerin oluşturduğu bir kare matris ile kat düzeyi fark vektörlerinin oluşturduğu genişletilmiş vektörün çarpımıyla ortaya çıkacak kolon vektör,kat kolon parametrelerini içerecektir.

$$\{Z\}=[A].\{f\}$$

$$\begin{bmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ 0 & A_{11} & A_{12} & A_{13} \\ 0 & 0 & A_{22} & A_{23} \\ 0 & 0 & 0 & A_{33} \end{bmatrix} . \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Genişletilmiş [A] matrisinin elemanı [Anm] matrisleri, kat kolon matrislerinin çarpımından oluştuğundan boyutu kat kolon geçiş matrisi ile aynıdır. Bununla beraber kat kolon matrisleri herbir kolon 6x6'lık çubuk geçiş matrislerinden oluştuğundan birkat düzeyindeki kolon adedi kadar genişletilmiştir.

Buna göre, beş açıklıklı bir çerçevede genişletilmiş kat kolon geçiş matrisi [C] 30x30'luk bir matris olacaktır. Böylece oluşacak [A] matrisi kat adedi çarpımı kadarlık bir kare matris olacaktırki, buda tüm sistemin aynı anda; [A] matrisini köşegen matrise dönüştürerek çözümünü imkansızlaştırmaktadır.

Tez amacında olduğu gibi kat adedinin artmasıyla oluşacak bu hesap güçlüğünü ortadan kaldırmak için sadece aynı kat düzeyindeki açıklık sayısına göre oluşacak matrisler ve işlemleriyle sonuca gidilebilmesi için gene Transfer Matris Metodu mantığıyla ardışık işlemler kullanılacaktır.

$\{Zm\}=\sum_{n=m}^{n-1} Amn.\{fn\}$ ifadesini zemin kat kolon alt ucu için yazılacak olursa;

$\{Zo\}=\sum Aon.\{fn\}$

$\{Zo\}=A_{oo}.\{fo\}+A_{o1}.\{f_1\}+A_{o2}.\{f_2\}+...+Aon.\{fn\}$ elde edilir.

$\{Zo\}$'ın hesabına müteakip ara kat değerleri için;

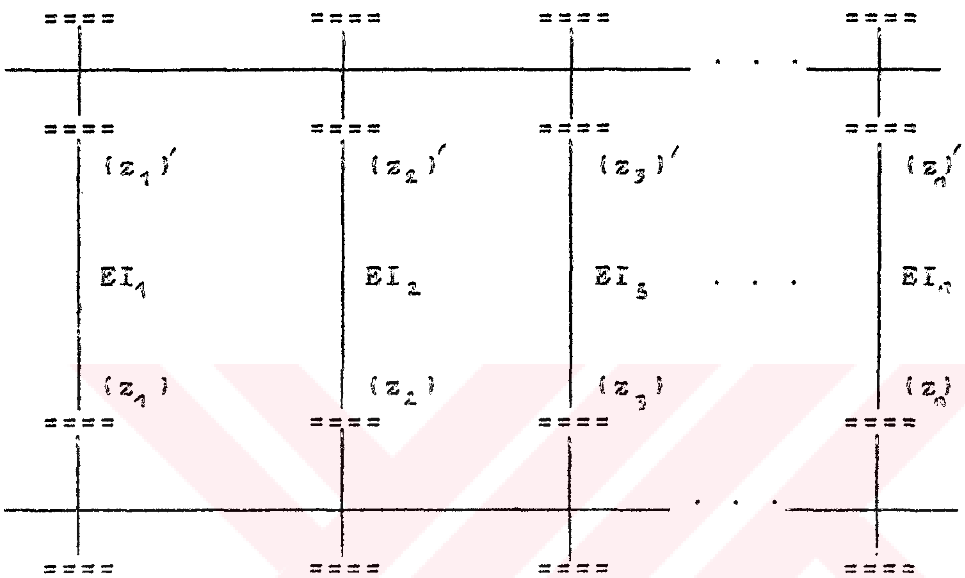
$\{Zo\}'=[Co]^{-1}.\{Zo\}$
$\{Z_1\}=\{Zo\}'-\{fo\}$
$\{Z_1\}'=[C_1]^{-1}.\{Z_1\}$
$\{Z_2\}=\{Z_1\}'-\{f_1\}$
$\{Z_2\}'=[C_2]^{-1}.\{Z_2\}$
$\{Z_3\}=\{Z_2\}'-\{f_2\}$
$\{Z_3\}'=[C_3]^{-1}.\{Z_3\}$

$\{Zn\}=\{Zn-1\}'-\{fn-1\}$
$\{Zn\}'=[Cn]^{-1}.\{Zn\}=\{fn\}$ ifade edilerek hesaplanır.

4.MODELE AİT GEÇİŞ MATRİSLERİNİN OLUŞTURUMU

4.1 KOLON GRUBU KAT TRANSFER MATRİSLERİNİN OLUŞTURUMU

4.1.-Yapılan Kabüller:



3.kısımda tanımlanmış olan,kat düzeyi kolon üst ucu tüm durum büyüklüklerini kat düzeyi kolon alt ucu tüm durum vektörlerine geçişini sağlayacak olan [C] kat düzeyi kolon geçiş matrisi;kattaki tüm çubukların herbirinin geçiş matrislerinin uygun birleşimlerinden faydalanılmasıyla bir genişletilmiş matris olarak elde edilecektir.

Bunun için bölüm 2.3'de tarifli ve bir çubuğun sağ durum büyüklüklerini,sol durum büyüklükleri cinsinden ifadesine yarayacak [U] çubuk geçiş matrisi kullanılacaktır.

$$[U]=\begin{bmatrix} 1 & 0 & 0 & L/EF & 0 & 0 \\ 0 & 1 & L & 0 & -L^3/6EI & -L^2/2EI \\ 0 & 0 & 1 & 0 & -L^2/2EI & -L/EI \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & L & 1 \end{bmatrix}$$

4.1.b-Bir Kat'a Ait Kolon Geçiş Matrislerinin Belirlenmesi:

Burada [U] geçiş matrisi, birtek çubuk eleman için sol durum büyüklük vektörünü sağ durum büyüklük vektörüne geçişini sağlamaktadır.

$$\overset{1}{\{Z\}} \xrightarrow{\{U\}} \overset{2}{\{Z\}'}$$

$\{Z\}'=[U].\{Z\}$'dir. Çubuk eleman burada kolon olacağından normal kuvvet etkiside birlikte dikkate alınmış haliyle durum büyüklükleri;

1 noktası durum büyüklükleri vektörü $\{Z\} = \begin{bmatrix} u \\ w \\ \varphi \\ N \\ Q \\ M \end{bmatrix}_1$

2 noktası durum büyüklükleri vektörü $\{Z\}' = \begin{bmatrix} u \\ w \\ \varphi \\ N \\ Q \\ M \end{bmatrix}_2'$dir.

Burada sağ düğüm noktası durum vektörü olan olan $\{Z\}'$, kattaki bir kolon çubuğunun üst ucuna, sol düğüm noktası durum vektörü $\{Z\}$, kolon alt ucuna eş kılınarak tek bir kolon çubuğunun geçiş matrisi olarak [U] alınması yeterli olacaktır.

Bundan yararlanılarak, [C] kat transfer matrisini Bölüm 3 de tanımlı haliyle yazabilmemiz için evvela kat düzeyi tüm kolon üst ucu parametrelerini, alt uc parametreleri cinsinden ifadesi oluşturulmalıdır.

Bunu sağlayacak genişletilmiş kat düzeyi kolon geçiş matrisi şu şekilde oluşturulacaktır:

Kat düzeyi kolon Üst ucu vektörleri: $\{z_1\}', \{z_2\}', \ldots, \{z_n\}'$,

" " " Alt " " : $\{z_1\}, \{z_2\}, \{z_3\}, \ldots \{z_n\}$dir.

Herbir çubuğun tüm durum vektörleri alt ve üst ucu olarak ayrı iki vektör halinde sınıflandırılabilir. Buna göre,

$$\{Z\}_{üst}=\begin{bmatrix}\{z_1\}' \\ \{z_2\}' \\ \{z_3\}' \\ . \\ . \\ . \\ \{z_n\}'\end{bmatrix} \qquad \{Z\}_{alt}=\begin{bmatrix}\{z_1\} \\ \{z_2\} \\ \{z_3\} \\ . \\ . \\ . \\ \{z_n\}\end{bmatrix}$$ şeklinde yazılabilir.

Kat düzeyi tüm üst parametrelerini içerecek olan {Z}üst vektörünü alt durum büyüklüklerinin tamamını içerecek olan {Z}alt vektörü cinsinden ifade edilirse;

{Z}üst=[K].{Z}alt'dır.

Bu ifadede görüldüğü gibi yeni bir [K] matrisi teşkil edilmelidir.[K],kat düzeyi geçiş matrisi olup,kat düzeyi kolon alt ucu parametrelerinin kolon üst ucu parametrelerine geçişi sağlayacaktır.[K] matrisini teşkil ederken her bir çubuğun ayrı ayrı geçiş matrislerinden faydalanılacağından, ilgili çubukların genişletilmiş matris içerisine köşegenlerde yerleştirilerek hazırlanacaktır.

$$\underset{(6nx1)}{\begin{bmatrix}\{z_1\}' \\ \{z_2\}' \\ \{z_3\}' \\ . \\ . \\ . \\ \{z_n\}'\end{bmatrix}} = \underset{(6nx6n)}{\begin{bmatrix}[U_1]_{6x6} & & & & & 0 \\ & [U_2]_{6x6} & & & & \\ & & [U_3]_{6x6} & & & \\ & & & . & & \\ & & & & . & \\ 0 & & & & & [U_n]_{6x6}\end{bmatrix}} , \underset{(6nx1)}{\begin{bmatrix}\{z_1\} \\ \{z_2\} \\ \{z_3\} \\ . \\ . \\ . \\ \{z_n\}\end{bmatrix}}$$

olarak bir köşegen matris elde edilmektedir.Böylece elde edilecek [K] matrisi kattaki kolon sayısıyla orantılı olarak boyutu artacaktır.Bu matrisi oluşturan herbir çubuk geçiş matrisi 6x6'lık bir matrisolduğu dikkate alındığında,kat kolon geçiş matrisi boyutu kattaki kolon sayısının 6 katı kadar olacaktır.

[K] matrisi teşkil edilirken kat yüksekliğinin ve elastikilik modülünün tüm kolonlar için aynı olduğu düşünül-düğünde her kat için değişken parametre olarak yalnız çubuk atalet momentleri belirlenecektir.

Bölüm 3'de yapılan modelleştirmede kullanılan kat kolon geçiş matrisi ile her bir çubuğun E,I,L parametrelerine göre teşkil edilmiş geçiş matrislerinden oluşacan [K] matrisi arasında tanım gereği,

$[C]=[K]^{-1}$ şeklinde tanımlı ilişki vardır.

Buna göre,modele ait kat kolon geçiş matrisinin teşkilinde,Bölüm 2.3'de bulunan her bir çubuğun geçiş matrisinin köşegen normda genişletilmiş halinden sonra inversi kullanılacaktır.

[K] matrisi daha sonraki aşamalarda ara kat değerlerinin bulunması amacıyla tekrar kullanılacaktır.

4.2. KAT DÜZEYİ ÇÖZÜMÜ İÇİN SÜREKLİ ÇUBUK SİSTEMİN TARNSFER MATRİSLERİ VE DURUM VEKTÖRLERİNİN BELİRLENMESİ

4.2.a-Yapılan Kabüller:

Bu bölümde kat düzeyinde bulunan kirişler dizisini bir sürekli sisteme benzeterek,yüke maruz sürekli sistemin çubuk uç durum vektörü bilinmeyenleri belirlenecektir.Bölüm 3'deki modelleştirmeye uygun olarak kat düzeyi parametreleri hesaplanıp,elde edilecek kat düzeyi durum vektörü bir çözüm vektör halinde modele ilişkin ardışık çarpımda yer alacaktır.

Çubuk sistemin ele alınışında,elde edilecek kat düzeyi çözüm vektörünün,model tanımı gereği kat düzeyinin bir alt katının kolon üst ucu parametreleri ile bir üst katın kolon alt ucu parametreleri farkını içerecek şekilde elde edilmesi gereklidir.

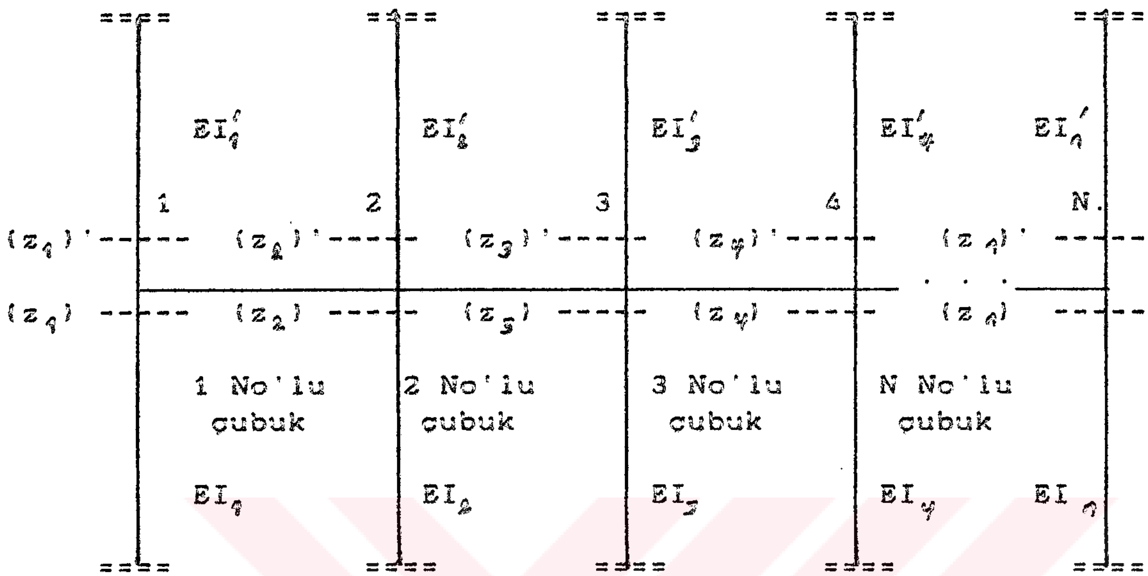
Bunun için çubuk sistemin alt kat kolonlarına mesnetli mütemadi kiriş olarak müstakilen çözümü yapılacak,mütemadi kirişin mesnetlendiği kolonlar ve üst kat kolonları tarafından zorlandığı kabülü ile mesnet şartları oluşturulacaktır. Nihayetinde kat çerçevesinin,mütemadi bir kiriş olarak çözümü yapılmasına imkan doğacaktır.Böylece sürekli sistemlerin çözümünde kullanışlı olan geçiş matrisleri metoduyla çözüme kolaylıkla ulaşılabilmiş olunacaktır.

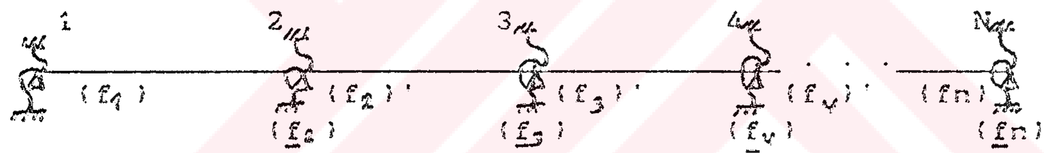
İlgili kat çerçevesinin mütemadi bir kiriş olarak benzetimini sağlayabilmek için,düğüm noktalarının durum vektörlerinin zorlanmış birer mesnet olarak tanımlanması gerekecektir.Mütemadi kirişin her bir çubuk elemanı için geçiş matrisi olarak Bölüm 2.3'de tanımlı [U] geçiş matrisi kullanılacaktır.

Yapılan modelleştirmede,mütemadi kirişin düğüm noktaları birer zorlanmış mesnet olarak nokta durum matrisleri oluşturulacak,bunun için mütemadi kirişin mesnetlendiği alt kolon üst ucu nokta durum matrisleri ile varsa üstteki kolonun alt ucu nokta durum matrislerinin farkından matris elemanları bulunacaktır.

Yapı sistemlerinin çözümünde karşılaştığımız kat çerçevelerinin mütemadi kirişe indirgenerek geçiş matrisleri metoduyla çözümünde kullanılacak çözüm matrisleri,açıklık sayısıyla orantılı olarak aşağıdaki oluşturulacak şekliyle kullanılacaktır.

Çözüm vektörünü bulacağımız kat çerçevesi;





şeklinde bir mütemadi kiriş olarak basitleştirilir.

Bölüm 2.4'de geçiş matrisleri metodunun sürekli çubuk sistemlerde uygulanışı anlatılmaktadır.

Buna göre,herbir çubuğun [U] geçiş matrisleri yazılacak, başlangıç ve son nokta durum vektörleri ile ara şartları ifade edecek $(\underline{f}_n)$ ara şart durum matrisleri nokta bilinmeyenlerini dikkate alınacak şekilde belirlenmelidir.

Ayrıca çubuk üzerinde bulunan yüklerden dolayı yük vektörleri hesaplanmalı ve yük katsayı çözüm matrisi çubuk eleman geçiş matrislerinin Bölüm 2.4'teki şekliyle ardışık çarpımları ile birlikte hazırlanmalıdır.

Elde edilecek olan çözüm katsayılar matrisindeki denklem takımlarından son nokta sınır şartları ile ara nokta sınır şartlarının belirlediği seçici matris yardımıyla otomatik

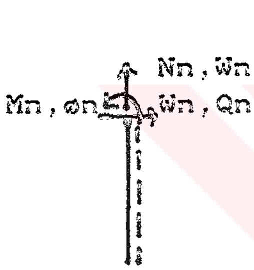
olarak bilinmeyen sayısı kadar denklem elde etmek mümkün olacaktır.

4.2.b-Düğüm Noktası Çözüm Vektörü İle Kolon Uç Vektörünün Arasındaki İlişkinin Belirlenmesi:

Bölüm 3'de yapılan modelleştirmede,{fn} kat düzeyi çözüm vektörü;çözümü yapılan mütemadi kirişin mesnetlendiği alt kolon üst ucu durum vektörü ile üst kolon alt ucu durum vektörünün farkı olarak ifadesi şu şekilde yapılır:

Kat düzeyi kirişler dizisini mesnetlerde zorlanmış birer mütemadi kiriş olarak modelleştirilmesinde,kolondan bir kesme kuvveti intikal etmediği ve iki kolon elastikliği oranında mesnette moment oluşacağı kabülleriyle mütemadi kirişin çözülmüş halini {fn},kat düzeyi çözüm vektörünü vermesi sağlanmış olur.

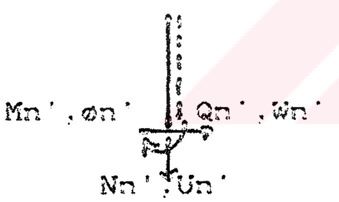
Buna göre,ilgili nokta alt kolonu üst ucu durum vektörü {Zn};



$$\{Z_n\} = \begin{bmatrix} U_n \\ W_n \\ \phi_n \\ N_n \\ Q_n \\ M_n \end{bmatrix}$$

Bilinenler	Bilinmeyenler
$U_n = 0$,	N_n
$W_n = 0$,	Q_n
$M_n = C_{\varphi_{alt}} . \phi_n$	ϕ_n

İlgili nokta üst kolonu alt ucu durum vektörü {Zn}':



$$\{Z_n\}' = \begin{bmatrix} 0 \\ 0 \\ \phi_n' \\ 0 \\ 0 \\ M_n' \end{bmatrix} \qquad -M_n' = C_{\varphi_{üst}} . \phi_n'$$

şeklinde ifadesiyle.durum vektörü her iki çubuk uç durum vektörlerinin süperpozisyonundan,

$$\{Z_n\} + \{Z_n\}' = \begin{bmatrix} 0 \\ 0 \\ \phi_n \\ N_n \\ Q_n \\ -C_{\varphi_{alt}} . \phi_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \phi_n' \\ 0 \\ 0 \\ -C_{\varphi_{üst}} . \phi_n' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \phi_n \\ N_n \\ Q_n \\ -(C_{\varphi_{alt}} + C_{\varphi_{üst}}) . \phi_n \end{bmatrix}$$

olarak elde edilir.

Bu vektör, dönmeye zorlanmış basit mesnet durum vektörü için düzenlenecek olursa, alt kolon normal kuvveti mesnet düşey tepkisine, alt kolon kesme kuvveti mesnet yatay tepkisine, kolon uçları momentleri toplamı mesnet momentine eşit değerler alacaktır. Bu içerikteki,

$$\{F\}=\begin{bmatrix} 0 \\ 0 \\ \emptyset n \\ Qn \\ Nn \\ -\emptyset n(C_{y_{\text{alt}}}+C_{y_{\text{üst}}}) \end{bmatrix}$$

nokta çözüm vektörü belirlenmiş olur.

Buna göre kat düzeyi çözüm vektörünü,

$$\{fn\}=\begin{bmatrix} \{Z_1\} \\ \{Z_2\} \\ \{Z_3\} \\ \cdot \\ \cdot \\ \cdot \\ \{Z_n\} \end{bmatrix}$$

oluşturan $\{Z_1\},\{Z_2\}\ldots\{Z_n\}$ kolon uçları çözüm vektörlerini elde edebilmek için, nokta çözüm vektörünü,

$$\{F\}=\begin{bmatrix} U \\ W \\ \emptyset \\ N \\ Q \\ M \end{bmatrix}$$

normunda mütemadi kiriş çözümüyle elde edilerek, $\{F\}^{*}$ ile $\{Z\}$ durum vektörleri aşağıdaki gibi eş kılınmalıdır.

$$\{Zn\}=\begin{bmatrix} 0 \\ 0 \\ \emptyset n \\ Nn \\ Qn \\ Mn \end{bmatrix}, \quad \{F\}^{*}=\begin{bmatrix} 0 \\ 0 \\ \emptyset n^{*} \\ Qn^{*} \\ Nn^{*} \\ Mn^{*} \end{bmatrix}$$

$$\begin{aligned} \emptyset n &= \emptyset n^{*} \\ Nn &= Qn^{*} \\ Qn &= Nn^{*} \\ -Mn^{*} &= C_{y_{\text{alt}}}\emptyset n + C_{y_{\text{üst}}}.\emptyset n \\ Mn &= C_{y_{\text{alt}}}.\emptyset n \\ -Mn &= Mn^{*}+C_{y_{\text{üst}}}.\emptyset n \end{aligned}$$

Sonuç olarak mütemadi kirişin düğüm noktası çözüm vektörü $\{fa\}$, elde edildikten sonra kat düzeyi çözüm vektörü $\{fn\}$'i oluşturan $\{Zn\}$ kolon uç vektörlerine geçebilmek için yukarıda anlatıldığı gibi iki vektörün eş kılınması gereklidir.

4.2.c.-Düğüm Noktası Çözüm Vektörlerinin Hesaplanması:

Şekil 4.2.2'deki gibi çözümü için mütemadi kiriş olarak modelleştirilen sürekli sistemin,bölüm 2.4'deki mütemadi kirişin genel çözümü matrisyel ifadesinden faydalanılarak çözümü yapılacaktır.

Mütemadi kirişe ait denklem sistemini içeren matrisyel ifade yük olmaması halinde şu şekildeydi:

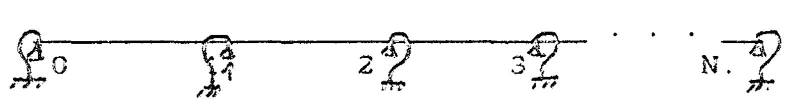
$$\begin{bmatrix} f_1' \\ f_2' \\ f_3' \\ \cdot \\ \cdot \\ \cdot \\ f_{n-1}' \\ f_n \end{bmatrix} = \begin{bmatrix} A_{11} & \underline{I} & 0 & 0 & \dots & 0 \\ A_{21} & A_{22} & \underline{I} & 0 & \dots & 0 \\ A_{31} & A_{32} & A_{33} & \underline{I} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \dots & \cdot \\ A_{n-1\,1} & A_{n-1\,2} & A_{n-1\,3} & . & A_{(n-1)(n-1)} & \underline{I} \\ A_{n1} & A_{n2} & A_{n3} & A_{n4} & \dots & A_{nn} \end{bmatrix} \cdot \begin{bmatrix} \underline{V}_0 & 0 & 0 & \dots & 0 \\ 0 & \underline{V}_1 & 0 & \dots & 0 \\ 0 & 0 & \underline{V}_2 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & 0 & 0 & \dots & \underline{V}_{n-1} \end{bmatrix} \begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \vdots \end{bmatrix}$$

$\{F\}=[A].[V].\{\hat{F}\}$ normuyla denklem takımı tanımlıdır.Burada [A] matrisi mütemadi kirişin çubuk geçiş matrislerinin ardışık çarpımları (Ann)'i içeren bir kare matris.[V] başlangıç düğüm noktasıyla son düğüm noktasının bir önceki ve son araşartlı noktaya kadarki düğüm noktaların durum katsayı matrislerini köşegen kabül eden bir genişletilmiş matris olarak tanımlandı.

Konu kapsamında ele alınacak sürekli sistemlerin çözümünde değişiklik gösteren husus,herbir çubuk elemanın geçiş matrislerinin ardışık çarpımlarıyla birlikte,düğüm noktalarına ait başlangıç,ara şart ve son nokta durum matrislerinin içeriği olacaktır.

Mütemadi kirişe ait modelde,kirişin ara kat veya son kat olması durumunda farklı iki şekilde başlangıç,ara şart ve son düğüm noktası durum matrislerinin belirlenmesi gerekecektir.

Kat düzeyinin son kat olması durumunda mesnetlerin yalnız ait kat kolonları tarafından dönmeye karşı tutulduğu göz önüne alınmalı ve buna göre başlangıç,ara şart ve son düğüm noktası durum matrislerinin ifadesi oluşturulmalıdır.



İlgili kat düzeyinin son kat olması durumunda başlangıç noktası durum vektörü;

$$
\{fo\}=\begin{bmatrix} u \\ w \\ \varnothing \\ N \\ Q \\ M \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ \varnothing o \\ No \\ Qo \\ -C_{\varphi_{\ddot{u}}}.\varnothing o \end{bmatrix}
$$

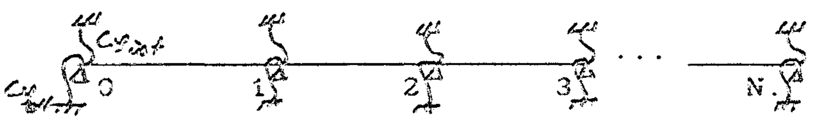
'dır. Bu ifade $\{fo\}=[Vo].\{\hat{fo}\}$ şeklinde yazılacak olursa,

$$
\{fo\}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -C_{\varphi_{\ddot{u}}} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} uo \\ wo \\ \varnothing o \\ No \\ Qo \\ Mo \end{bmatrix}
$$

yazılarak [Vo] başlangıç düğüm noktası durum matrisi elde edilmiş olur.

Modelde aradaki düğüm noktaları ile son düğüm noktası için ara şart durum vektörüde aynı mesnet şartlarına sahip olaçağından V durum matrisleride aynı içeriğe sahip olacaktır. İlgili kat düzeyinin ara kat olması durumunda, yalnızca kolonların her ikisi birden dönmeye karşı zorlandıklarından C_{φ} değerleriyerlerine $-(C_{\varphi_{alt}}+C_{\varphi_{\ddot{u}st}})$ olarak konulacaktır.

$\{fn\}$ ara kat düzeyi mütemadi kirişine ait nokta durum vektörü ise,



$$
\{fn\}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -(C_{\varphi_{\ddot{u}st}}+C_{\varphi_{alt}}) & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ w \\ \varnothing \\ N \\ Q \\ M \end{bmatrix}
$$

Modeldeki kat düzeyi mütemadi kirişin,çözümü için elde edilecek denklem sistemlerini oluşturan;

$$\{F\}=[A].\{V\}.\{\hat{F}\}+[B].\{W\}$$

matrisyel ifadesinde yer alan bütün alt matris ve değerler oluşturulmuş oldu.

Buna göre,

-[A] matrisinin teşkili için,U_1 ,U_2 ,U_3 ...Un herbir çubuk eleman geçiş matrisi olduğu halde herbirinin boyutları 6x6 olan matrislerin ardışık çarpımları bulunmalıdır.

$A_{11}=U_1$
$A_{21}=U_2.U_1$
$A_{22}=U_2$
$A_{31}=U_3.U_2.U_1$
$A_{32}=U_3.U_2$
$A_{33}=U_3$
$A_{n1}=U_n.U_{n-1}.U_{n-2}...U_2.U_1$
$A_{n2}=U_n.U_{n-1}.U_{n-2}...U_2$
$A_{nn-1}=U_n.U_{n-1}$
$A_{nn}=U_{nn}$

[A] matrisinde ayrıca I (6x6)'lik birim matriste yer almaktadır.

-{V} matrisi içeriğindeki ait matrisler mütemadi kirişin.

Vo,başlangıçtaki nokta durum matrisini,

V_1 ,1.nokta durum matrisini,

V_{n-1} ,son düğüm noktasından bir önceki son ara şart durum matrislerini köşegen kabül eden genişletilmiş matris olarak modelde tanımlıdır.

Mütemadi kirişin her düğüm noktası birer zorlanmış mesnet olması sebebiyle,durum matrislerinin yapısı aynı olup içeriği alt ve üst kolon elastikleri bakımından değişiklik gösterecektir.

Mütemadi kirişi çözerken herbir düğüm noktası olarak kabül edilen zorlanmış mesnetlere ait aşağıdaki formda,nokta durum matrisleri başlangıç noktasından son ara şart noktasına kadar(sondan bir önceki düğüm noktası) her birinin teşkil edilmesi gereklidir.

Modelde alt ve üst kolon tarafından zorlanmış nokta durum matrisleri,

$$[Vn]=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -(C_{y_{alt}}+C_{y_{üst}}) & 0 & 0 & 0 \end{bmatrix}$$

(6x6) şeklinde teşkil edilir.

Başlangıç noktasından başlamak üzere herbiri 6x6'lık ilk mesnetten başlayan, son mesnetten bir öncekinde son bulan toplam mesnet sayısından bir eksiği kadar [Vn]matrisleri oluşturularak, bunların köşegenlere yerleştirilmesiyle [V] matrisi elde edilir.

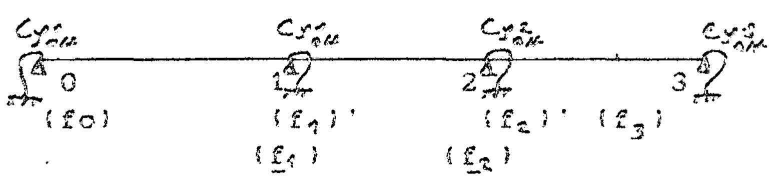
Yüklenmiş mütemadi kirişin çözümünde ilave olarak yük vektörü katsayı matrisi olan [B]'nin içeriğinde [A] matrinden farklı çarpımlar yer almamakta, yalnızca matrisin teşkil formu farklı olmaktadır.

Ayrıca yük vektörünün sayısal ifadesi farklı yükleme durumlarına göre Ek.1'de verilmiştir. Buna göre [B] matrisinin genel hali;

$$[B]=\begin{bmatrix} \underline{I} & 0 & 0 & \dots & 0 \\ A_{22} & \underline{I} & 0 & \dots & 0 \\ A_{32} & A_{33} & \underline{I} & \dots & 0 \\ . & . & . & & . \\ A_{n2} & A_{n3} & & \dots & \underline{I} \end{bmatrix}$$

'dir.

Örnek olarak üç açıklıklı bir kat çerçevesine ait mütemadi kirişin çözüm denklemlerini içeren matrisyel ifade şu şekilde oluşturulur. Buna göre daha fazla açıklık için genelleştirme yapılabilecektir.



$$\begin{bmatrix}\{f_1\}\\ \{f_2\}\\ \{f_3\}\end{bmatrix} = \begin{bmatrix} A_{11}.V_0 & V_1 & 0 \\ A_{21}.V_0 & A_{22}.V_1 & V_2 \\ A_{31}.V_0 & A_{32}.V_1 & A_{33}.V_2\end{bmatrix} . \begin{bmatrix}\{\hat{f}_0\}\\ \{\hat{f}_1\}\\ \{\hat{f}_2\}\end{bmatrix} + \begin{bmatrix} I & 0 & 0\\ A_{22} & I & 0 \\ A_{32} & A_{33} & I\end{bmatrix} . \begin{bmatrix}W_1\\ W_2\\ W_3\end{bmatrix}$$

Denklem sisteminde (F) vektörü 1.,2.ve son düğüm noktası sınır şartlarından (F) bilinmeyenleri kadar denklem elde edilmektedir.

$$\begin{bmatrix} u_1' \\ w_1' \\ \phi_1' \\ N_1' \\ Q_1' \\ M_1' \\ \\ u_2' \\ w_2' \\ \phi_2' \\ N_2' \\ Q_2' \\ M_2' \\ \\ u_3 \\ w_3 \\ \phi_3 \\ N_3 \\ Q_3 \\ M_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \phi_1' \\ N_1' \\ Q_1' \\ -\Sigma C y_1' . \phi_1 \\ \\ 0 \\ 0 \\ \phi_2' \\ N_2' \\ Q_2' \\ -\Sigma C y_2' . \phi_2 \\ \\ 0 \\ 0 \\ \phi_3 \\ N_3 \\ Q_3 \\ -\Sigma C y_3 . \phi_3 \end{bmatrix} \quad \begin{array}{l} 1.\text{Denklem} \\ 2. \quad " \\ 10. \quad " \\ 11. \quad " \\ 12. \quad " \\ 3. \quad " \\ \\ 4.\text{Denklem} \\ 5. \quad " \\ 13. \quad " \\ 14. \quad " \\ 15. \quad " \\ 6. \quad " \\ \\ 7.\text{Denklem} \\ 8. \quad " \\ 16. \quad " \\ 17. \quad " \\ 18. \quad " \\ 9. \quad " \end{array}$$

Yukarıdaki ilk 9 denklemden (F) vektörü içeriğindeki.

$\{\hat{f}_0\}$ vektörü bilinmeyenleri ϕ_0.No.Qo ve $C_{y_0}.\phi_0=M_0$

$\{\underline{f}_1\}$ " " ϕ_1 ,N_1 ,Q_1 ve $C_{y_1}.\phi_1=M_1$

$\{\underline{f}_2\}$ " " ϕ_2 ,N_2 ,Q_2 ve $C_{y_2}.\phi_2=M_2'$dan

oluşan 9 bilinmeyen bulunur.

Birinci düğüm noktası bilinmeyenleri aynı zamanda ilk 9 denklemden elde edilmiştir.Bilinen 9 durum bilinmeyenleri diğer denklemler yardımıyla,

1.Nokta bilinmeyenleri ϕ_1' ,N_1' ,Q_1' 10.,11.,12.Denklemlerden

2. " " ϕ_2' ,N_2' ,Q_2' 13.,14.,15. "

3. " " ϕ_3' ,N_3' ,Q_3' 16.,17.,18. "

elde edilecektir.

Böylece nokta çözüm vektörleri olan $\{fo\}$, $\{f_1\}'$, $\{f_2\}'$, $\{f_3\}$ vektörlerinin içerikleri bulunmuş olur.Mesnet noktası sol durum büyüklükleri,sürekli çubuk sistemlerde geçiş matrisleri metodunun uygulanması tanımı gereği,sağ düğüm noktasından ara şartı farkı kadar olacaktır.

$$\{f_1\}'=\{f_1\}+\{\underline{f}_1\} \Longrightarrow \{f_1\}=\{f_1\}'-\{\underline{f}_1\}$$

Bunagöre,bulunan $\{f_1\}'$, $\{\underline{f}_1\}$, $\{f_2\}'$, $\{\underline{f}_2\}$vektörlerinden faydalanılarak $\{f_1\}$, $\{f_2\}$ vektörleri elde edilir.

$$\{f_2\}=\{f_2\}'-\{\underline{f}_2\}$$

5. KOLON GRUBU UÇ DURUM VEKTÖRLERİ İLE KAT DÜZEYİ ÇÖZÜM DURUM VEKTÖRÜNÜN SÜPERPOZİSYONU

Çok katlı çerçeve sisteminin tez kapsamında yapılan modelleştirmesinde, n kat adedi için zemin kat kolon alt ucu toplam ifadesi Bölüm 3'de formülize edildiği gibi;

$$\{Zo\}=\sum_{n=o}^{n-1} Aon.\{fn\}$$

$\{Zo\}=Aoo.\{fo\}+Ao_1.\{f_1\}+Ao_2.\{f_2\}+...+Ao_n.\{fn\}$ şeklinde ifade edilmişti.

Buna göre, herbir kat kolon geçiş matrislerinin ardışık çarpımları olan Amn matrisleri, kat düzeyi kolonları üst parametrelerini kolonların alt ucu parametrelerine geçişini sağlayan [Cn] matrisleri yardımıyla elde edilecektir.

$$[C]=[K]^{-1}=\begin{bmatrix} [U_1] & 0 & 0 & 0 & \dots & 0 \\ & [U_2] & 0 & 0 & \dots & 0 \\ & & [U_3] & 0 & \dots & 0 \\ & & & . & \dots & 0 \\ & & & & \dots. & 0 \\ & & & & . & 0 \\ 0 & & & & & [U_n] \end{bmatrix}$$

N. kat düzeyi kolon üst ucu durum vektörü $\{Zn\}'$,

Alt ucu durum vektörü $\{Zn\}$,

Üst ucu durum vektörünü alt ucu durum vektörüne geçişini sağlayan geçiş matrisi $[Cn]$, olduğuna göre n. kat alt ve üst paremetreleri'nin birbiri cinsinden ifadesi: $\{Zn\}=[Cn].\{Zn\}'$ şeklinde yapılacaktır.

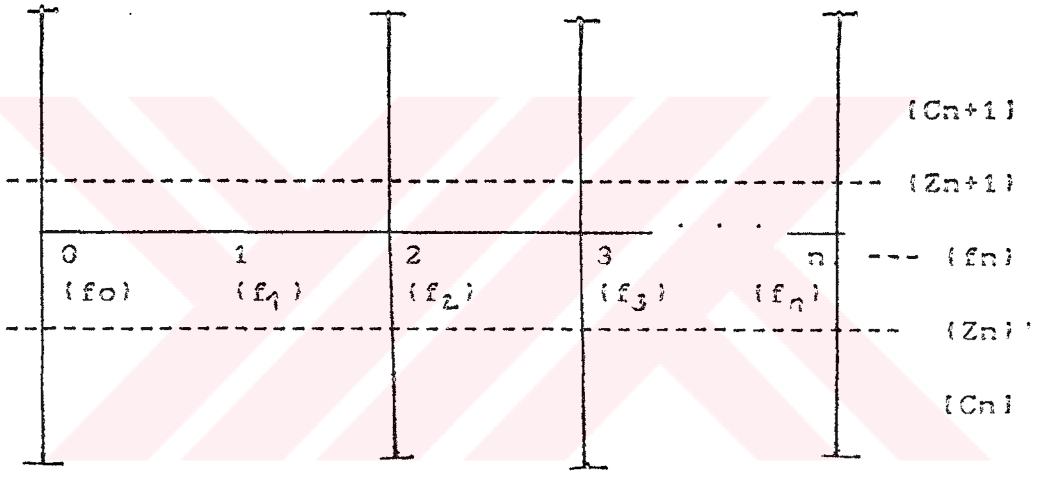
Toplam ifadenin açılımında yer alan Amn matrislerinin çarpımı durumunda bulunan n.kat çözüm vektörü $\{fn\}$, kat düzeyi'nin altındaki kolonun üst ucu parametreleri'nin üstündeki kolonun alt ucu parametreleri değerleri farkını ifade etmektedir. Buna göre denklem 3.1'de ,

$$\{Zn\}'-\{Zn+1\}=\{fn\}$$

vektörüne eşit kılınacaktır.

Kolon üst ucu parametresi olan $\{Zn\}'$ vektörünün içeriğinde, kat düzeyinde bulunan her bir düğüm noktası çözüm değerleri yer alacaktır. Kat düzeyi çözüm vektörü $\{f\}$'nün parametreleri sürekli çubuk sisteme benzetilen modelin soldan sağa doğru yer alan mesnet değerlerini içerecektir.

$$\{Zn\}'=\{fn\}=\begin{bmatrix} fo \\ f_1 \\ f_2 \\ . \\ . \\ . \\ f_n \end{bmatrix}$$



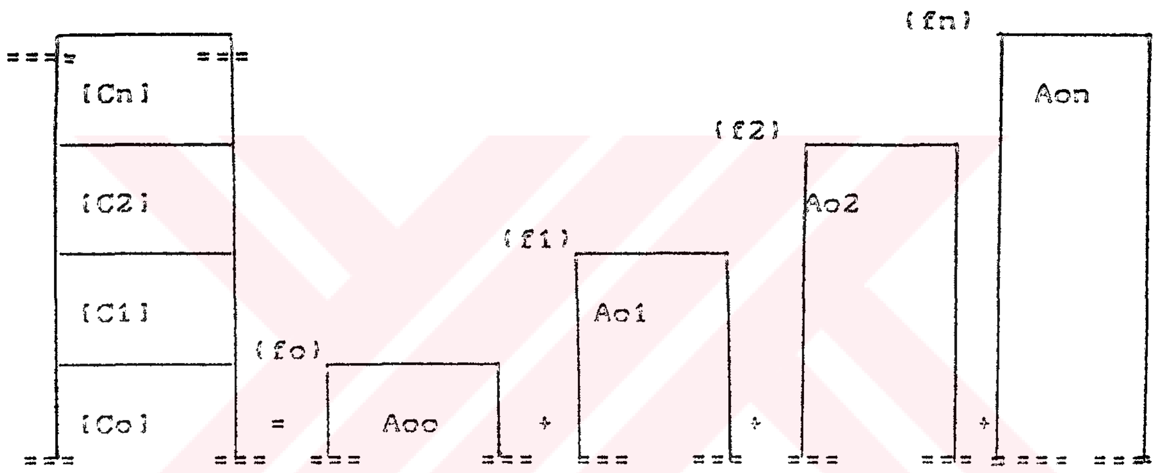
$\{Zn+1\}=[Cn+1].\{Zn+1\}'$
$\{Zn\}' =\{Zn+1\}+\{fn\}$
$\{Zn\} =[Cn].\{Zn\}'$

(n+1).kat'ın son kat olması durumunda $\{Zn+1\}'=\{fn+1\}$ dir. Aynı zamanda $\{fn+1\}$ vektörü son katın, sürekli sisteme benzetimiyle elde edilen son kat düzeyi çözüm vektörü olarak elde edilecektir. Daha sonra kat düzeyi kolon geçiş matrisi yardımıyla son kat kolon üst parametrelerini içeren kolon ucu durum vektöründen kolon alt ucu durum vektörüne geçiş sağlanabilecektir.

Kat düzeyinin üst ve alt kat çözüm vektörlerinin farkı olarak toplam dizide yer aldığından sürkli sistemi mesnetleri zorlanmış bir mütemadi kiriş benzetimiyle çözerek mesnet değerleri elde ediimektedir.

Kat düzeyi mesnet değerlerini sırasıyla içeren çözüm vektörü,alt kat kolonları üst durum vektörlerine geçiş için üst kat kolonları alt ucu durum vektörlerine eklenecektir. Aynı işlem ardışık çarpımların ayrı ayrı toplamlarıyla en alt kat kolon alt uçları durum vektörlerini elde edilene kadar toplamlar ayrı ayrı hesaplanacaktır.

Bu açıklama tarzı ile aynı zamanda Bölüm 3'de ele alınan modelleştirmenin sistem çözüm mantığına uygunluğunu göstermektedir.Söyleki;



Toplam sistem 1.Kat düzeyi 2.Kat d. 3.Kat d. n.Kat d.

1.Kat Düzeyi: $\{Zo\}=Aoo.\{fo\}$
2.Kat Düzeyi: $\{Zo\}=Ao1.\{f1\}$
3.Kat Düzeyi: $\{Zo\}=Ao2.\{f2\}$
. " " . . .
. " " . . .
. " " . . .
n.Kat Düzeyi: $\{Zo\}=Aon.\{fn\}$

------------- -------------

Toplam Sistem: $\{Zo\}=Aoo.\{fo\}+Ao1.\{f1\}+Ao2.\{f2\}+\ldots+Aon.\{fn\}$

$$\{Zo\}=\sum_{n=o}^{n-1} Aon.\{fn\}$$

6.SİSTEM ÇÖZÜMÜ İÇİN İZLENİLECEK ALGORİTMA

1.-Kat Kolon Grubu Geçiş Matrislerinin Hesaplanması:

1.a.-Çubuk Geçiş Matrislerinin Hesaplanması

Kolon grubu içinde yer almakta olan herbir kolon çubuk elemanının geçiş matrisi;[Ui]

$$[Ui]=\begin{bmatrix} 1 & 0 & 0 & l/EF & 0 & 0 \\ 0 & 1 & 1 & 0 & -l^3/6EI & -l^2/2EI \\ 0 & 0 & 1 & 0 & -l^2/2EI & -l/EI \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix};$$

Bir kattaki kolon sayısı:n

Hesaplanacak olan n tane [Ui] matrisleridir.

		Verilenler:	Hesaplanacaklar:
1.Kolon Özellikleri	:	I1,h1.E	[U1]
2.Kolon "	:	I2,h2,E	[U2]
3.Kolon "	:	I3,h3,E	[U3]
. " "	:	.,. ,.	.
N. " "	:	In,hn,E	[Un]

1.b.-Genişletilmiş Kat Kolon Geçiş Matrisinin Oluşturulması

Bir kattaki kolon sayısı:n

Genişletilmiş Kat Kolon Geçiş Matrisinin Boyutu:(6.nX6.n)

$$[Kn]=\begin{bmatrix} [U1]_{6\times6} & 0 & 0 & 0 & 0 \\ 0 & [U2]_{6\times6} & 0 & 0 & 0 \\ 0 & 0 & [U3]_{6\times6} & 0 & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & [Un]_{6\times6} \end{bmatrix}_{(6.n\times6.n)}$$

[Cn] kat kolon grubu geçiş matrisi:$[Cn]=[Kn]^{-1}$ hesaplanılacaktır.Bu işlem kat adedi kadar tekrarlanarak kat kolon grubu [Co],[C1],[C2],....,[Cn] geçiş matrisleri elde edilir.

2.Kat Kolon Grubu Geçiş Matrislerinin Ardışık Çarpımlarının hesaplanması:

Kat Adedi:n
İlgili Kat Numarası:m
Kat Kolon Grubu Geçiş Matrislerinin Ardışık Çarpım Matrisleri:Amn;

Aoo=[Co]
Ao1=[Co].[C1]
Ao2=[Co].[C1].[C2]
. = . . .
. = . . .
Aon=[Co].[C1].[C2] ... [Cn]

A11=[C1]
A12=[C1].[C2]
A13=[C1].[C2].[C3]
. = . . .
. = . . .
A1n=[C1].[C2].[C3] ... [Cn]

Amm =[Cm]
Am(m+1)=[Cm].[Cm+1]
Am(m+2)=[Cm].[Cm+1].[Cm+2]
. = . . .
. = . . .
Am(n-1)=[Cm].[Cm+1].[Cm+2] ... [Cn-2].[Cn-1]
Amn =[Cm].[Cm+1].[Cm+2] ... [Cn-1].[Cn]

Sistem çözümü toplam ifadesi olan;

$(Zm)=\sum_{n=0}^{n-1}$ Amn.(fn)'nin açılımında sistemin tamamının çözümü isteniliyorsa m=0 için yazılarak zemin kat kolon alt ucu çözüm vektörü (Zo) elde edilecektir ki,burada yalnızca Aon kat düzeyi kolon geçiş matrisleri ardışık çarpımları gerekli olacaktır.

Sürkli sisteme ait kat düzeyi çözüm vektörleri {fn}'nin hesaplanmasından sonra kat kolon grubu geçiş matrisleri ardışık çarpımları da kullanılarak zemin kat kolon grubu alt ucu çözüm vektörü elde edilmektedir. Bundan sonra ara kat değerlerinin bulunması için kat düzeyi çözüm vektörleri farkları ve kolon grubu alt ucu parametrelerini üst ucu parametrelerine geçişini sağlayan kolon grubu geçiş matrislerinden faydalanılacaktır.

En alt kat kolon grubu alt ucu çözüm vektörü:{Zo}
En alt kat kolon grubu üst ucu çözüm vektörü:{Zo}'
En alt kat kolon grubu üst ucu parametrelerini
Alt ucu parametrelerine geçişini sağlayan matris:[Co]
En alt kat kolon grubu alt ucu parametrelerini
üst ucu parametrelerine geçişini sağlayan matris:[Ko]

Bilinenler	Hesaplanacaklar	Aralarındaki Bağıntı
{Zo},[Ko]	{Zo}'	{Zo}'=[Ko].{Zo}
{Zo}',{fo}	{Z1}	{Z1} ={Zo}'-{fo}
{Z1},[K1]	{Z1}'	{Z1}'=[K1].{Z1}
{Z1}',{f1}	{Z2}	{Z2} ={Z1}'-{f1}
{Z2},[K2]	{Z2}'	{Z2}'=[K2].{Z2}
. , .	.	. = . . .
. , .	.	. = . . .
{Zn-1},[Kn-1]	{Zn-1}'	{Zn-1}'=[Kn-1].{Zn-1}
{Zn-1},{fn-1}	{Zn}	{Zn} ={Zn-1}-{fn-1}
{Zn},[Kn]	{Zn}'	{Zn}'=[Kn].{Zn}
{Zn}',{fn}	{Zn+1}	{Zn+1} ={Zn}'-{fn}

Yukarıdaki ardışık işlemler neticesinde tüm sisteme ait ara çözüm değerleri elde edilebilmektedir. Bu işlemlerde faydalanılan ara kat ve son kat sürekli sistemlerinin çözüm vektörleri {fn}, mütemadi kirişin alt kat ve üst kat parametreleri farkı olarak çözümü gerekmektedir.

3.Kat Düzeyi Mütemadi Kirişlerin Çözüm Vektörlerinin Bulunması:

Kat çerçevesinin modelleştirilmiş hali olan sürekli sistemin çözümü için gerekli denklem sistemlerini içeren matrisyel ifade;

$$\{F\}=[A].\{V\}.\{\hat{F}\}+[B].\{W\}$$ şeklindedir.

[A]'nın teşkili:

$$[A]=\begin{bmatrix} A11 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A21 & A22 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A31 & A32 & A33 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ A41 & A42 & A43 & A44 & I & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ An1 & An2 & An3 & An4 & An5 & . & . & . & I & 0 \end{bmatrix}$$

Sürekli çubuk sistemi içinde yer alan her bir açıklıktaki çubuk elemanın geçiş matrisi;[Ui]

Bir kattaki açıklık sayısı:n

Hesaplanacak olan n adet [Ui] matrisidir.

	Verilenler:	Hesaplanacaklar:
1.Açıklıktaki çubuk Özellikleri:	I1,L1,E	[U1]
2. " " "	I2,L2,E	[U2]
3. " " "	I3,L3,E	[U3]
. . . .	. ,. ,.	.
. . . .	. ,. ,.	.
N. " " "	In,Ln,E	[Un]

Kat düzeyi sürekli sistemin çubuk geçiş matrislerinin ardışık çarpımları:Amn;

Açıklık sayısı:n

A11=[U1]

A21=[U2].[U1]

A22=[U2]

A31=[U3].[U2].[U1]

A32=[U3].[U2]

A33=[U3]

An1=[Un].[Un-1].[Un-2] ... [U2].[U1]

An2=[Un].[Un-1].[Un-2] ... [U2]

Ann=[Un]

[V] Matrisi teşkili:

Bir Kattaki Açıklık Sayısı:n

Bir Kattaki Düğüm Noktası Durum Vektörü:n+1

[V] Matrisi Boyutları:6(n+1)X6(n+1)

Modeldeki sürekli sisteme ait bir düğüm noktası,zorlanmış mesnet durum matrisi:

$$[Vn]=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -(C_{y_{alt}}+C_{y_{üst}}) & 0 & 0 & 0 \end{bmatrix}_{6x6}$$

$C_{y_{üst}}$=4.E.Iüst/hüst

$C_{y_{alt}}$=4.E.Ialt/halt

Hesaplanacak olan (n+1) adet [Vn] matrisidir.

	Verilenler:	Hesaplanacaklar:
1.Düğüm Noktası Özellikleri:	Iüst,hüst,E	
	Ialt,halt,E	[V1]
2.Düğüm Noktası Özellikleri:	Iüst,hüst,E	
	Ialt,hüst,E	[V2]
. " " "	: . , . ,.	.
N. " " "	:Iüst,hüst,E	
	Ialt,halt,E	[Vn]

6(n+1)x6(n+1) boyutundaki kat durum matrisi,genişletilmiş bir matris olup her bir düğüm noktası durum matrislerinin sırası ile köşegenlere yerleştirimi yapılarak hazırlanmalıdır.

$$[V]=\begin{bmatrix} [Vo] & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & [V1] & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & [V2] & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & [V3] & 0 & \ldots & 0 \\ . & . & . & . & . & \ldots & 0 \\ . & . & . & . & . & \ldots & . \\ 0 & 0 & 0 & 0 & 0 & \ldots & [Vn] \end{bmatrix}_{6(n+1)\times 6(n+1)}$$

[B] matrisinin teşkili:

$$
[B]=\begin{bmatrix}
I & 0 & 0 & 0 & 0\ ...0 & 0 \\
A22 & I & 0 & 0 & 0\ ...0 & 0 \\
A32 & A33 & I & 0 & 0\ ...0 & 0 \\
A42 & A43 & A44 & I & 0\ ...0 & 0 \\
. & . & . & . & .\ .... & . \\
. & . & . & . & .\ .... & . \\
An2 & An3 & An4 & An5 & 0\ ...Ann & I
\end{bmatrix}
$$

Amn ardışık çarpımları,sürekli çubuk sistem içinde yer alan çubuk eleman geçiş matrisleri yardımıyla her kat için ayrı ayrı hesaplanarak kat adedi kadar [B] matrisleri oluşturulacaktır.

$$
[I]=\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}_{6x6}
$$

Daha önceden hesaplanmış olan sürekli çubuk sisteme ait geçiş matrislerinin ardışık çarpımları aynen kullanılacaktır.

[A22]=[U2]

[A32]=[U3].[U2]
[A33]=[U3]

[A42]=[U4].[U3].[U2]
[A43]=[U4].[U3]
[A44]=[U4]

[An2]=[Un].[Un-1] ... [U3].[U2]
[An3]=[Un].[Un-1] ... [U3]
. = ...
[Ann]=[Un]

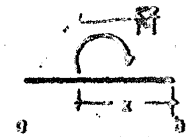
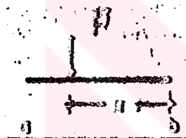
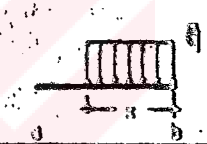
[B] matrisinin çarpımı durumunda bulunan {W} vektörü,kat düzeyi sürekli çubuk sisteminin maruz kaldığı yükleri ifade etmektedir.{W} vektörleri sırası ile her bir açıklıktaki yük grubunun veya yükün şekline göre Ek1'deki değerleri alacaktır.Bun göre her bir açıklıktaki yüklemenin değerleri ayrı ayrı her kat için hesaplanacaktır.

Bölüm 4.2.c'de çözüm şekline açıklık getirilen denklem sistemini içeren,

$\{F\}=[A].[V].\{\hat{F}\}+[B].\{W\}$ matrisyel ifade oluşturulmuş olur.

Sayfa 27'deki gibi başlangıç sınır şartları yardımıyla bilinmeyen sayısı kadar denklem elde etmek mümkün olacaktır. Hesaplanan başlangıç düğüm noktası ve ara düğüm noktaları sağı çözüm durum vektörü,son düğüm noktası solu çözüm vektörlerinin sırasıyla ele alınmasıyla sistemin toplamı ile ilgili formülasyonunda kullanılacak kat düzeyi çözüm vektörleri elde edilmiş olacaktır.Bu değerler sayfa 34'deki gibihesap düzeni içerisinde yer alarak ara kat değerlerinin tümünü elde etmek mümkün olacaktır.

Geçiş matrisleri metodu : Yük terimleri   EJ = sabit

Z					
U	0	0	0	0	0
W	$-\frac{Mx^2}{2EI}$	$-\frac{Px^3}{6EI}$	$\frac{qx^4}{24EI}$	$\frac{ql^4}{120EI}$	$\frac{ql^4}{30EI}$
φ	$-\frac{Mx}{EI}$	$\frac{Px^2}{2EI}$	$\frac{qx^3}{6EI}$	$\frac{ql^3}{24EI}$	$\frac{ql^3}{8EI}$
N	0	0	0	0	0
Q	0	$-P$	$-qx$	$-\frac{ql}{2}$	$-\frac{ql}{2}$
M	M	$-Px$	$-\frac{qx^2}{2}$	$-\frac{ql^2}{6}$	$-\frac{ql^2}{3}$

KAYNAKLAR

1- Ders Notları: Prof. İRDESEL GÜĞÜŞ

2- Matrizenmethoden der statik und dynamik
Teiel1: Statik, Michael Lawo, Georg Theieraup

ÖZGEÇMİŞ

1969 yılı Arapgir-Malatya doğumludur. 1985 yılında girdiği Yıldız Üniversitesi Mühendislik Fakültesi İnşaat Mühendisliği Bölümü mezunudur. 1989 yılında aynı üniversitenin Fen Bilimleri Enstitüsü İnşaat Bilim Dalı'nda yüksek lisans öğrenimine başlamıştır. Halen müteahhitlik hizmetleri yapmaktadır.

DEGER GIRISI   BILGI GORME   CIKIS

SABITLER
KOLON BOYUTLARI
KAT YUKSEKLIKLERI
KIRIS BOYUTLARI
KIRIS ACIKLIKLARI
YUKLEMELER
HESAPLA

DEGER GIRI

KOLON SAYISI = 0
KAT SAYISI = 0
E = 0

SABITLE
KOLON B
KAT YUK
KIRIS B
KIRIS A
YUKLEME
HESAPLA

DEGER GIRISI BILGI GORME CIKIS

SABITILER
KO
KA | CERCEVE DUZLEMINDEKI KOLON BOYUTU
KI | CERCEVE DUZLEMINE DIK KOLON BOYUTU
KI
YUKLEMELER
HESAPLA

CERCEVE DUZLEMINDEKI KOLON BOYUTU

	1.KLN	2.KLN	3.KLN	4.KLN
0.KOLON	.00	.00	.00	.00
1.KOLON	.00	.00	.00	.00
2.KOLON	.00	.00	.00	.00
3.KOLON	.00	.00	.00	.00

DEGER GIRISI   BILGI GORME   CIKIS

SABITLER
KOLON BOYUTLARI
KAT YUKSEKLIKLERI
KIRIS BOYUTLARI
KIRIS ACIKLIKLARI
YUKLEMELER
HESAPLA

KOLON YUKSEKLIKLER

0.KOLON	0.00
1.KOLON	0.00
2.KOLON	0.00
3.KOLON	0.00

CERCEVE DUZLEMINDEKI KIRIS YUKSEKLIGI

	1.KLN	2.KLN	3.KLN
0.KOLON	.00	.00	.00
1.KOLON	.00	.00	.00
2.KOLON	.00	.00	.00
3.KOLON	.00	.00	.00

DEGER GİRİSİ   BİLGİ GORME   CIKIS

SABİTLER
KOLON BOYUTLARİ
KAT YUKSEKLİKLERİ
KIR
KIR CERCEVE DUZLEMİNDEKİ KİRİS YUKSEKLIGİ
YUK CERCEVE DUZLEMİNE DIK KIRIS GENISLIGİ
HES

DEGER GİRİSİ   BİLGİ GORME   CIKIS

SABİTLER
KOLON BOYUTLARİ
KAT YUKSEKLİKLERİ
KİRİS BOYUTLARİ
KİRİS ACIKLIKLARİ
YUKLEMELER
HESAPLA

KIRIS BOYLARI

1.KOLON	0.00
2.KOLON	0.00
3.KOLON	0.00

DEGER GIRISI   BILGI GORME   CIKIS

KOLON KESIT ALANLARI MATRISI
KOLON ATALET MOMENTLERI MATRISI
KOLON CUBUK GECIS MATRISLERI
KIRIS KESIT ALANLARI MATRISI
KIRIS ATALET MOMENTLERI MATRISI
KIRIS CUBUK GECIS MATRISLERI
KIRIS CUBUK YUK VEKTORLERI

```
{$F+,N+}
{$M 8192,0,655300}

USES
  DOS,
  OPINLINE,
  OPSTRING,
  OPROOT,
  OPCRT,
  OPCOLOR,
  {$IFDEF UseMouse}
  OPMOUSE,
  {$ENDIF}
  OPFRAME,
  OPCMD,
  OPWINDOW,
  OPMENU,
  OPLARRAY,
  OPENTRY,
  MMENU;

  {$IFDEF UseMouse}
CONST
  MOUSECHAR : CHAR = #04;
  {$ENDIF}

{COLOR SET USED BY MENU SYSTEM}
CONST
  MENUCOLORS : COLORSET = (
    TEXTCOLOR          : YELLOWONBLUE;        TEXTMONO          : LTGRAYONBLACK;
    CTRLCOLOR          : YELLOWONBLUE;        CTRLMONO          : WHITEONBLACK;
    FRAMECOLOR         : CYANONBLUE;          FRAMEMONO         : LTGRAYONBLACK;
    HEADERCOLOR        : WHITEONCYAN;         HEADERMONO        : BLACKONLTGRAY;
    SHADOWCOLOR        : DKGRAYONBLACK;       SHADOWMONO        : WHITEONBLACK;
    HIGHLIGHTCOLOR     : WHITEONRED;          HIGHLIGHTMONO     : BLACKONLTGRAY;
    PROMPTCOLOR        : BLACKONCYAN;         PROMPTMONO        : LTGRAYONBLACK;
    SELPROMPTCOLOR     : BLACKONCYAN;         SELPROMPTMONO     : LTGRAYONBLACK;
    PROPROMPTCOLOR     : BLACKONCYAN;         PROPROMPTMONO     : LTGRAYONBLACK;
    FIELDCOLOR         : YELLOWONBLUE;        FIELDMONO         : LTGRAYONBLACK;
    SELFIELDCOLOR      : BLUEONCYAN;          SELFIELDMONO      : WHITEONBLACK;
    PROFIELDCOLOR      : LTGRAYONBLUE;        PROFIELDMONO      : LTGRAYONBLACK;
    SCROLLBARCOLOR     : CYANONBLUE;          SCROLLBARMONO     : LTGRAYONBLACK;
    SLIDERCOLOR        : CYANONBLUE;          SLIDERMONO        : WHITEONBLACK;
    HOTSPOTCOLOR       : BLACKONCYAN;         HOTSPOTMONO       : BLACKONLTGRAY;
    BLOCKCOLOR         : YELLOWONCYAN;        BLOCKMONO         : WHITEONBLACK;
    MARKERCOLOR        : WHITEONMAGENTA;      MARKERMONO        : BLACKONLTGRAY;
    DELIMCOLOR         : BLUEONCYAN;          DELIMMONO         : WHITEONBLACK;
    SELDELIMCOLOR      : BLUEONCYAN;          SELDELIMMONO      : WHITEONBLACK;
    PRODELIMCOLOR      : BLUEONCYAN;          PRODELIMMONO      : WHITEONBLACK;
    SELITEMCOLOR       : YELLOWONCYAN;        SELITEMMONO       : BLACKONLTGRAY;
    PROITEMCOLOR       : LTGRAYONBLUE;        PROITEMMONO       : LTGRAYONBLACK;
    HIGHITEMCOLOR      : WHITEONBLUE;         HIGHITEMMONO      : WHITEONBLACK;
    ALTITEMCOLOR       : WHITEONBLUE;         ALTITEMMONO       : WHITEONBLACK;
    ALTSELITEMCOLOR    : WHITEONCYAN;         ALTSELITEMMONO    : BLACKONLTGRAY;
    FLEXAHELPCOLOR     : WHITEONBLUE;         FLEXAHELPMONO     : WHITEONBLACK;
    FLEXBHELPCOLOR     : WHITEONBLUE;         FLEXBHELPMONO     : WHITEONBLACK;
    FLEXCHELPCOLOR     : LTCYANONBLUE;        FLEXCHELPMONO     : BLACKONLTGRAY;
    UNSELXREFCOLOR     : YELLOWONBLUE;        UNSELXREFMONO     : LTBLUEONBLACK;
    SELXREFCOLOR       : WHITEONMAGENTA;      SELXREFMONO       : BLACKONLTGRAY;
    MOUSECOLOR         : WHITEONRED;          MOUSEMONO         : BLACKONLTGRAY
  );


  NST
```

```
  MaxKolon =  8;
  ExitProg : Boolean = FALSE;

 Type
  UIType       = Array[1..6, 1..6] of Double;
  UIMatrix     = Array[0..MaxKat, 1..MaxKolon] of UIType;
  UIMatrix_    = Array[0..MaxKat, 1..MaxKolon-1] of UIType;

 Var
  UI           : ^UIMatrix;
  UI_          : ^UIMatrix_;
  V            : ^UIMatrix_;
  DummyUI      : UIType;
  Status       : Word;
  Mnu          : Menu;        {Menu system}
  E            : Double;
  H            : Array[0..MaxKat] of Double;
  A, B, I, F   : Array[0..MaxKat, 1..MaxKolon] of Double;
  A_, B_,
  II, F_       : Array[0..MaxKat, 1..MaxKolon-1] of Double;
  L_           : Array[1..MaxKolon-1] of Double;
  ES           : EntryScreen;
  Kat,
  Kolon        : Word;
  I_, J_       : Word;
  KatKolGM     : Array[0..MaxKat] of VirtualArray;
  KatDurM      : Array[0..MaxKat] of VirtualArray;
  KirisGKM     : VirtualArray;
  BMatris,
  AMatris      : VirtualArray;


 {$F+}
 procedure ErrorHandler(UnitCode : Byte; var ErrCode : Word; Msg : string);
   {-Report errors}
 begin
   RingBell;
 end;

 procedure DisplayHelp(UnitCode : Byte; IdPtr : Pointer; HelpIndex : Word);
   {-Display context sensitive help}
 begin
 end;

 procedure CustomizeItemString(var Name : string; Key : LongInt;
                               Selected, Highlighted : Boolean;
                               WPtr : RawWindowPtr);
   {-Customize menu item strings}
 begin
 end;
 {$F-}

 procedure Mult(A, B: UIType; Var C:UIType);
  Var
   i, j, k : Word;
  Begin
   For i:=1 to 6 do
    For j:=1 to 6 do
     Begin
      C[i,j]:=0;
      For k:=1 to 6 do
       C[i,j]:=C[i,j]+A[i,k]*B[k,j];
     End;
  End;
```

```
PROCEDURE MASSAGE(S: STRING);
VAR
 CW : COMMANDWINDOW;
 BEGIN
  CW.INIT(10, 9, 60, 13, ENTRYCOMMANDS, ULNONE);
  CW.WOPTIONSON(WBORDERED);
  CW.DRAW;
  FASTCENTER(S, 2, 7);
  CW.GETNEXTCOMMAND;
  CW.DONE;
 END;


{ KOLONLAR ICIN }

PROCEDURE KESITHESAPLA;
 BEGIN
  FOR I_:=0 TO KAT-1 DO
   FOR J_:= 1 TO KOLON DO
    F[I_, J_] := A[I_, J_] * B[I_, J_];
 END;

PROCEDURE ATALETHESAPLA;
 BEGIN
  FOR I_:=0 TO KAT-1 DO
   FOR J_:= 1 TO KOLON DO
    I[I_, J_] := A[I_, J_] * B[I_, J_] * B[I_, J_] * B[I_, J_] / 12;
 END;

PROCEDURE UIHESAPLA;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KAT VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
   FOR I_:=0 TO KAT-1 DO
    FOR J_:= 1 TO KOLON DO
     BEGIN

      UI^[I_, J_][1,1] := 1;  { 1.SATIR }
      UI^[I_, J_][1,2] := 0;
      UI^[I_, J_][1,3] := 0;
      UI^[I_, J_][1,4] := H[I_]/(E*F[I_,J_]);
      UI^[I_, J_][1,5] := 0;
      UI^[I_, J_][1,6] := 0;

      UI^[I_, J_][2,1] := 0;  { 2.SATIR }
      UI^[I_, J_][2,2] := 1;
      UI^[I_, J_][2,3] := H[I_];
      UI^[I_, J_][2,4] := 0;
      UI^[I_, J_][2,5] := -H[I_]*H[I_]*H[I_]/(6*E*I[I_, J_]);
      UI^[I_, J_][2,6] := -H[I_]*H[I_]/(2*E*I[I_, J_]);;

      UI^[I_, J_][3,1] := 0;  { 3.SATIR }
      UI^[I_, J_][3,2] := 0;
      UI^[I_, J_][3,3] := 1;
      UI^[I_, J_][3,4] := 0;
      UI^[I_, J_][3,5] := -H[I_]*H[I_]/(2*E*I[I_, J_]);
      UI^[I_, J_][3,6] := -H[I_]/(E*I[I_, J_]);;

      UI^[I_, J_][4,1] := 0;  { 4.SATIR }
      UI^[I_, J_][4,2] := 0;
      UI^[I_, J_][4,3] := 0;
      UI^[I_, J_][4,4] := 1;
      UI^[I_, J_][4,5] := 0;
```

```
 IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
  BEGIN
   MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
   EXIT;
  END;
 ES.INIT(2, 2, 78, 24);
 ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
 ES.WFRAME.ADDHEADER('CERCEVE DUZLEMINDEKI KOLON BOYUTU', HETC);
 FOR I_ := 0 TO KAT-1 DO
  FOR  J_ := 1 TO KOLON DO
   ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, B[I_,J_]);
 ES.DRAW;
 FOR I_ := 0 TO KAT-1 DO
  FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
 FOR J_ := 1 TO KOLON DO
  FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
 FASTTEXT(CHARSTR('-',6*KOLON), 4, 11);
 ES.PROCESSSELF;
 ES.DONE;
 ATALETHESAPLA;
 KESITHESAPLA;
END;

PROCEDURE CERCEVE60KU;
BEGIN
 IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
  BEGIN
   MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
   EXIT;
  END;
 ES.INIT(2, 2, 78, 24);
 ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
 ES.WFRAME.ADDHEADER('CERCEVE DUZLEMINE DIK KOLON BOYUTU', HETC);
 FOR I_ := 0 TO KAT-1 DO
  FOR  J_ := 1 TO KOLON DO
   ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, A[I_,J_]);
 ES.DRAW;
 FOR I_ := 0 TO KAT-1 DO
  FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
 FOR J_ := 1 TO KOLON DO
  FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
 FASTTEXT(CHARSTR('-',6*KOLON), 4, 11);
 ES.PROCESSSELF;
 ES.DONE;
 ATALETHESAPLA;
 KESITHESAPLA;
END;

PROCEDURE KATYUKSEKLIKLERIOKU;
BEGIN
 IF (KAT<1) OR (KAT>MAXKAT) THEN
  BEGIN
   MASSAGE('KOLON SAYISI HATALI !');
   EXIT;
  END;
 ES.INIT(48, 2, 65, 24);
 ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
 ES.WFRAME.ADDHEADER('KOLON YUKSEKLIKLERI', HETC);
 FOR I_ := 0 TO KAT-1 DO
  ES.ADDDBLFIELD('', 2, 2, '9.99', 2+I_, 12, 0, 0, 6.00, 0, HE[I_]);
 ES.DRAW;
 FOR I_ := 0 TO KAT-1 DO
  ES.WFASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 2+I_, 5);
 ES.PROCESSSELF;
 ES.DONE;
```

```
      UI^[I_, J_][4,6] := 0;

      UI^[I_, J_][5,1] := 0;  { 5.SATIR }
      UI^[I_, J_][5,2] := 0;
      UI^[I_, J_][5,3] := 0;
      UI^[I_, J_][5,4] := 0;
      UI^[I_, J_][5,5] := 1;
      UI^[I_, J_][5,6] := 0;

      UI^[I_, J_][6,1] := 0;  { 6.SATIR }
      UI^[I_, J_][6,2] := 0;
      UI^[I_, J_][6,3] := 0;
      UI^[I_, J_][6,4] := 0;
      UI^[I_, J_][6,5] := H[I_];
      UI^[I_, J_][6,6] := 1;

    END;
  END;

  PROCEDURE UIGOSTER;
   VAR
    DONE : BOOLEAN;
    I__, J__ : WORD;
    BUFFERUI : ^UITYPE;
   BEGIN
    IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
     BEGIN
      MASSAGE('KAT VEYA KOLON SAYISI HATALI !');
      EXIT;
     END;
    UIHESAPLA;
    NEW(BUFFERUI);
    DONE := FALSE;
    I__ := 0;
    J__ := 1;
    ES.INIT(2, 2, 78, 24);
    ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
    ES.WFRAME.ADDHEADER(' KOLON CUBUKLARI GECIS MATRISI ', HEIC);
    FOR I_ := 1 TO 6 DO
     FOR  J_ := 1 TO 6 DO
      ES.ADDDBLFIELD('', 2, 2, '.9999', 4+I_, 5+6*J_, 0, 0, 0.9999, 0, BUFFERUI^[I_,J_]);
    ES.DRAW;
    FOR I_ := 1 TO 6 DO
     FASTTEXT(LEFTPAD(LONG2STR(I_)+' |', 8), 5+I_, 3);
    FOR J_ := 1 TO 6 DO
     FASTTEXT(LONG2STR(J_), 4, 7+6*J_);
    FASTTEXT(CHARSTR('-',6*6), 5, 11);
    REPEAT
     ES.WFASTTEXT('KAT NO       : '+LEFTPAD(LONG2STR(I__), 2), 13, 7);
     ES.WFASTTEXT('KOLON NO   : '+LEFTPAD(LONG2STR(J__), 2), 14, 7);
     MOVE(UI^[I__,J__], BUFFERUI^, SIZEOF(BUFFERUI^));
     ES.DRAW;
     ES.GETNEXTCOMMAND;
     CASE ES.GETLASTCOMMAND OF
      CCQUIT: DONE := TRUE;
      CCLEFT: IF I__>0 THEN DEC(I__);
      CCUP: IF J__>1 THEN DEC(J__);
      CCRIGHT: IF I__<MAXKAT-1 THEN INC(I__);
      CCDOWN: IF J__<MAXKOLON THEN INC(J__);
     END;
    UNTIL DONE;
    ES.DONE;
    DISPOSE(BUFFERUI);
  END;

  PROCEDURE CERCEVEZORU
```

```
 END;

PROCEDURE KOLONKESITGOSTER;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('KOLON KESIT ALANLARI MATRISI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR  J_ := 1 TO KOLON DO
    ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, F[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*KOLON), 4, 11);

  REPEAT
   ES.PROCESSSELF;
  UNTIL ES.GETLASTCOMMAND=CCDONE;

  ES.DONE;
 END;

PROCEDURE ATALETGOSTER;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('ATALET MOMENTLERI MATRISI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR  J_ := 1 TO KOLON DO
    ES.ADDDBLFIELD('', 2, 2, '.9999', 4+I_, 5+6*J_, 0, 0, 0.99, 0, I[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*KOLON), 4, 11);

  REPEAT
   ES.PROCESSSELF;
  UNTIL ES.GETLASTCOMMAND=CCDONE;

  ES.DONE;
 END;


  {KIRISLER ICIN }

  PROCEDURE KESITHESAPLA_;
  BEGIN
  FOR I_:=0 TO KAT-1 DO
  FOR J_:= 1 TO KOLON-1 DO
  F_[I_, J_] := A_[I_, J_] * B_[I_, J_];
 END;
```

```
PROCEDURE ATALETHESAPLA_;
 BEGIN
  FOR I_:=0 TO KAT-1 DO
   FOR J_:= 1 TO KOLON-1 DO
    II[I_, J_] := A_[I_, J_] * B_[I_, J_] * B_[I_, J_] * B_[I_, J_] / 12;
 END;

PROCEDURE UIHESAPLA_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
   FOR I_:=0 TO KAT-1 DO
    FOR J_:= 1 TO KOLON-1 DO
     BEGIN
      UI_^[I_, J_][1,1] := 1;  { 1.SATIR }
      UI_^[I_, J_][1,2] := 0;
      UI_^[I_, J_][1,3] := 0;
      UI_^[I_, J_][1,4] := L_[J_]/(E*F_[I_,J_]);
      UI_^[I_, J_][1,5] := 0;
      UI_^[I_, J_][1,6] := 0;

      UI_^[I_, J_][2,1] := 0;  { 2.SATIR }
      UI_^[I_, J_][2,2] := 1;
      UI_^[I_, J_][2,3] := L_[J_];
      UI_^[I_, J_][2,4] := 0;
      UI_^[I_, J_][2,5] := -L_[J_]*L_[J_]*L_[J_]/(6*E*II[I_, J_]);
      UI_^[I_, J_][2,6] := -L_[J_]*L_[J_]/(2*E*II[I_, J_]);;

      UI_^[I_, J_][3,1] := 0;  { 3.SATIR }
      UI_^[I_, J_][3,2] := 0;
      UI_^[I_, J_][3,3] := 1;
      UI_^[I_, J_][3,4] := 0;
      UI_^[I_, J_][3,5] := -L_[J_]*L_[J_]/(2*E*II[I_, J_]);
      UI_^[I_, J_][3,6] := -L_[J_]/(E*II[I_, J_]);;

      UI_^[I_, J_][4,1] := 0;  { 4.SATIR }
      UI_^[I_, J_][4,2] := 0;
      UI_^[I_, J_][4,3] := 0;
      UI_^[I_, J_][4,4] := 1;
      UI_^[I_, J_][4,5] := 0;
      UI_^[I_, J_][4,6] := 0;

      UI_^[I_, J_][5,1] := 0;  { 5.SATIR }
      UI_^[I_, J_][5,2] := 0;
      UI_^[I_, J_][5,3] := 0;
      UI_^[I_, J_][5,4] := 0;
      UI_^[I_, J_][5,5] := 1;
      UI_^[I_, J_][5,6] := 0;

      UI_^[I_, J_][6,1] := 0;  { 6.SATIR }
      UI_^[I_, J_][6,2] := 0;
      UI_^[I_, J_][6,3] := 0;
      UI_^[I_, J_][6,4] := 0;
      UI_^[I_, J_][6,5] := L_[J_];
      UI_^[I_, J_][6,6] := 1;

     END;
 END;

PROCEDURE UIGOSTER_;
 VAR
  DONE : BOOLEAN;
```

```
Procedure AtaletHesapla_;
 Begin
  For i_:=0 to Kat-1 do
   For j_:= 1 to Kolon-1 do
    II[i_, j_] := A_[i_, j_] * B_[i_, j_] * B_[i_, j_] * B_[i_, j_] / 12;
 End;

Procedure UIHesapla_;
 Begin
  If (Kolon<1) or (Kolon>MaxKolon) or (Kat<1) or (Kat>MaxKat) then
   Begin
    Massage('Kolon veya Kolon Sayisi Hatali !');
    Exit;
   End;
   For i_:=0 to Kat-1 do
    For j_:= 1 to Kolon-1 do
     Begin
      UI_^[i_, j_][1,1] := 1;  { 1.SATIR }
      UI_^[i_, j_][1,2] := 0;
      UI_^[i_, j_][1,3] := 0;
      UI_^[i_, j_][1,4] := L_[j_]/(E*F_[i_,j_]);
      UI_^[i_, j_][1,5] := 0;
      UI_^[i_, j_][1,6] := 0;

      UI_^[i_, j_][2,1] := 0;  { 2.SATIR }
      UI_^[i_, j_][2,2] := 1;
      UI_^[i_, j_][2,3] := L_[j_];
      UI_^[i_, j_][2,4] := 0;
      UI_^[i_, j_][2,5] := -L_[j_]*L_[j_]*L_[j_]/(6*E*II[i_, j_]);
      UI_^[i_, j_][2,6] := -L_[j_]*L_[j_]/(2*E*II[i_, j_]);;

      UI_^[i_, j_][3,1] := 0;  { 3.SATIR }
      UI_^[i_, j_][3,2] := 0;
      UI_^[i_, j_][3,3] := 1;
      UI_^[i_, j_][3,4] := 0;
      UI_^[i_, j_][3,5] := -L_[j_]*L_[j_]/(2*E*II[i_, j_]);
      UI_^[i_, j_][3,6] := -L_[j_]/(E*II[i_, j_]);;

      UI_^[i_, j_][4,1] := 0;  { 4.SATIR }
      UI_^[i_, j_][4,2] := 0;
      UI_^[i_, j_][4,3] := 0;
      UI_^[i_, j_][4,4] := 1;
      UI_^[i_, j_][4,5] := 0;
      UI_^[i_, j_][4,6] := 0;

      UI_^[i_, j_][5,1] := 0;  { 5.SATIR }
      UI_^[i_, j_][5,2] := 0;
      UI_^[i_, j_][5,3] := 0;
      UI_^[i_, j_][5,4] := 0;
      UI_^[i_, j_][5,5] := 1;
      UI_^[i_, j_][5,6] := 0;

      UI_^[i_, j_][6,1] := 0;  { 6.SATIR }
      UI_^[i_, j_][6,2] := 0;
      UI_^[i_, j_][6,3] := 0;
      UI_^[i_, j_][6,4] := 0;
      UI_^[i_, j_][6,5] := L_[j_];
      UI_^[i_, j_][6,6] := 1;

     End;
End;

Procedure UIGoster_;
 Var
  Done : Boolean;
```

```
  BUFFERUI : ^UIType;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  UIHESAPLA_;
  NEW(BUFFERUI);
  DONE := FALSE;
  I__ := 0;
  J__ := 1;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER(' KIRIS CUBUKLARI GECIS MATRISI ', HETC);
  FOR I_ := 1 TO 6 DO
   FOR  J_ := 1 TO 6 DO
    ES.ADDDBLFIELD('', 2, 2, '.9999', 4+I_, 5+6*J_, 0, 0, 0.9999, 0, BUFFERUI^[I_,J_]);
  ES.DRAW;
  FOR I_ := 1 TO 6 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+' |', 8), 5+I_, 3);
  FOR J_ := 1 TO 6 DO
   FASTTEXT(LONG2STR(J_), 4, 7+6*J_);
  FASTTEXT(CHARSTR('-',6*6), 5, 11);
  REPEAT
   ES.WFASTTEXT('KOLON NO      : '+LEFTPAD(LONG2STR(I__), 2), 13, 7);
   ES.WFASTTEXT('KIRIS NO    : '+LEFTPAD(LONG2STR(J__), 2), 14, 7);
   MOVE(UI_^[I__,J__], BUFFERUI^, SIZEOF(BUFFERUI^));
   ES.DRAW;
   ES.GETNEXTCOMMAND;
   CASE ES.GETLASTCOMMAND OF
    CCQUIT: DONE := TRUE;
    CCLEFT: IF I__>0 THEN DEC(I__);
    CCUP: IF J__>1 THEN DEC(J__);
    CCRIGHT: IF I__<MAXKAT-1 THEN INC(I__);
    CCDOWN: IF J__<MAXKOLON-1 THEN INC(J__);
   END;
  UNTIL DONE;
  ES.DONE;
  DISPOSE(BUFFERUI);
END;

PROCEDURE CERCEVEZOKU_;
 BEGIN
  IF (KOLON<2) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('CERCEVE DUZLEMINDEKI KIRIS YUKSEKLIGI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR  J_ := 1 TO KOLON-1 DO
    ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, B_[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON-1 DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);
  ES.PROCESSSELF;
  ES.DONE;
  HALETHESAPLA_;
  ESITHESAPLA_;
  ...
```

```
    BufferUI : ^UIType;
  BEGIN
   IF (Kolon<1) OR (Kolon>MaxKolon) OR (Kat<1) OR (Kat>MaxKat) THEN
    BEGIN
     Massage('Kolon veya Kolon Sayisi Hatali !');
     Exit;
    END;
   UIHesapla_;
   New(BufferUI);
   Done := False;
   I__ := 0;
   J__ := 1;
   ES.Init(2, 2, 78, 24);
   ES.wOptionsOn(wBordered+wUserContents);
   ES.wFrame.AddHeader(' KIRIS CUBUKLARI GECIS MATRISI ', heTC);
   For I_ := 1 to 6 do
    For J_ := 1 to 6 do
     ES.AddDblField('', 2, 2, '.9999', 4+I_, 5+6*J_, 0, 0, 0.9999, 0, BufferUI^[I_,J_]);
   ES.Draw;
   For I_ := 1 to 6 do
    FastText(LeftPad(Long2Str(I_)+' |', 8), 5+I_, 3);
   For J_ := 1 to 6 do
    FastText(Long2Str(J_), 4, 7+6*J_);
   FastText(CharStr('-',6*6), 5, 11);
   Repeat
    ES.wFastText('Kolon No      : '+LeftPad(Long2Str(I__), 2), 15, 7);
    ES.wFastText('Kiris No    : '+LeftPad(Long2Str(J__), 2), 14, 7);
    Move(UI_^[I__,J__], BufferUI^, SizeOf(BufferUI^));
    ES.Draw;
    ES.GetNextCommand;
    Case ES.GetLastCommand of
     ccQuit: Done := True;
     ccLeft: If I__>0 then Dec(I__);
     ccUp: If J__>1 then Dec(J__);
     ccRight: If I__<MaxKat-1 then Inc(I__);
     ccDown: If J__<MaxKolon-1 then Inc(J__);
    END;
   Until Done;
   ES.Done;
   Dispose(BufferUI);
END;

Procedure CerceveZoku_;
 BEGIN
  If (Kolon<2) or (Kolon>MaxKolon) or (Kat<1) or (Kat>MaxKat) then
   BEGIN
    Massage('Kolon veya Kolon Sayisi Hatali !');
    Exit;
   END;
  ES.Init(2, 2, 78, 24);
  ES.wOptionsOn(wBordered+wUserContents);
  ES.wFrame.AddHeader('CERCEVE DUZLEMINDEKI KIRIS YUKSEKLIGI', heTC);
  For I_ := 0 to Kat-1 do
   For J_ := 1 to Kolon-1 do
    ES.AddDblField('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, B_[I_,J_]);
  ES.Draw;
  For I_ := 0 to Kat-1 do
   FastText(LeftPad(Long2Str(I_)+'.Kolon |', 8), 5+I_, 3);
  For J_ := 1 to Kolon-1 do
   FastText(Long2Str(J_)+'.Kin', 3, 5+6*J_);
  FastText(CharStr('-',6*(Kolon-1)), 4, 11);
  ES.ProcessSelf;
  ES.Done;
  AtaletHesapla_;
  EsitHesapla_;
  ...
```

```
PROCEDURE CERCEVE8OKU_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('CERCEVE DUZLEMINE DIK KIRIS GENISLIGI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR J_ := 1 TO KOLON-1 DO
    ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, A_[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON-1 DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);
  ES.PROCESSSELF;
  ES.DONE;
  ATALETHESAPLA_;
  KESITHESAPLA_;
 END;

PROCEDURE KIRISBOYLARIOKU_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) THEN
   BEGIN
    MASSAGE('KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(48, 2, 65, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('KIRIS BOYLARI', HETC);
  FOR I_ := 1 TO KOLON-1 DO
   ES.ADDDBLFIELD('', 2, 2, '9.99', 2+I_, 12, 0, 0, 9.99, 0, L_[I_]);
  ES.DRAW;
  FOR I_ := 1 TO KOLON-1 DO
   ES.WFASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 2+I_, 3);
  ES.PROCESSSELF;
  ES.DONE;
 END;

PROCEDURE KIRISKESITGOSTER_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('KIRIS KESIT ALANLARI MATRISI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR J_ := 1 TO KOLON-1 DO
    ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, F_[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON-1 DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);

  REPEAT
```

```
PROCEDURE CERCEVE6OKU_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('CERCEVE DUZLEMINE DIK KIRIS GENISLIGI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR J_ := 1 TO KOLON-1 DO
    ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, A_[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON-1 DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);
  ES.PROCESSSELF;
  ES.DONE;
  ATALETHESAPLA_;
  KESITHESAPLA_;
 END;

PROCEDURE KIRISBOYLARIOKU_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) THEN
   BEGIN
    MASSAGE('KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(48, 2, 65, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('KIRIS BOYLARI', HETC);
  FOR I_ := 1 TO KOLON-1 DO
   ES.ADDDBLFIELD('', 2, 2, '9.99', 2+I_, 12, 0, 0, 9.99, 0, L_[I_]);
  ES.DRAW;
  FOR I_ := 1 TO KOLON-1 DO
   ES.WFASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 2+I_, 3);
  ES.PROCESSSELF;
  ES.DONE;
 END;

PROCEDURE KIRISKESITGOSTER_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('KIRIS KESIT ALANLARI MATRISI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR J_ := 1 TO KOLON-1 DO
    ES.ADDDBLFIELD('', 2, 2, '.99', 4+I_, 5+6*J_, 0, 0, 0.99, 0, F_[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON-1 DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);
```

```
   ES.PROCESSSELF;
  UNTIL ES.GETLASTCOMMAND=CCDONE;

  ES.DONE;
 END;

PROCEDURE ATALETGOSTER_;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  ES.INIT(2, 2, 78, 24);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
  ES.WFRAME.ADDHEADER('KIRIS ATALET MOMENTLERI MATRISI', HETC);
  FOR I_ := 0 TO KAT-1 DO
   FOR  J_ := 1 TO KOLON-1 DO
    ES.ADDDBLFIELD('', 2, 2, '.9999', 4+I_, 5+6*J_, 0, 0, 0.99, 0, II[I_,J_]);
  ES.DRAW;
  FOR I_ := 0 TO KAT-1 DO
   FASTTEXT(LEFTPAD(LONG2STR(I_)+',KOLON |', 8), 5+I_, 3);
  FOR J_ := 1 TO KOLON-1 DO
   FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
  FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);
  REPEAT
   ES.PROCESSSELF;
  UNTIL ES.GETLASTCOMMAND=CCDONE;
  ES.DONE;
 END;



PROCEDURE KATKOLONGMATRISIHESAPLA;
 VAR
  I, J, K, L : WORD;
  R : REAL;
 BEGIN
  FOR I:=0 TO KAT-1 DO
   BEGIN
    KATKOLGM[I].INIT( 6*KOLON, 6*KOLON,
                      SIZEOF(DOUBLE),
                      'KKGM'+LONG2STR(I)+'.MTX',
                      10240,
                      LRANGECHECK+LKEEPDISKCURRENT);
    FOR J:=1 TO KOLON DO
     BEGIN
      FOR K:=1 TO 6 DO
       FOR L:=1 TO 6 DO
        BEGIN
         R := UI^[I, J][K, L];
         KATKOLGM[I].SETA((J-1)*6+K-1, (J-1)*6+L-1, R);
         FASTWRITE(PAD(LONG2STR(I)+','+LONG2STR(J)+','+LONG2STR(K)+','+LONG2STR(L), 80), 1, 1
);
        END;
     END;
    KATKOLGM[I].STOREA('KKGM'+LONG2STR(I)+'.MTX');
    KATKOLGM[I].CLOSEA;
    KATKOLGM[I].DONE;
   END;
 END;

PROCEDURE SABITLERIOKU;
 BEGIN
  ES.INIT(20, 8, 60, 15);
  ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
```

```
  ES.PROCESSSELF;
 UNTIL ES.GETLASTCOMMAND=CCDONE;

 ES.DONE;
END;

PROCEDURE ATALETGOSTER_;
BEGIN
 IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
  BEGIN
   MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
   EXIT;
  END;
 ES.INIT(2, 2, 78, 24);
 ES.WOPTIONSON(WBORDERED+WUSERCONTENTS);
 ES.WFRAME.ADDHEADER('KIRIS ATALET MOMENTLERI MATRISI', HETC);
 FOR I_ := 0 TO KAT-1 DO
  FOR  J_ := 1 TO KOLON-1 DO
   ES.ADDDBLFIELD('', 2, 2, '.9999', 4+I_, 5+6*J_, 0, 0, 0.99, 0, II[I_,J_]);
 ES.DRAW;
 FOR I_ := 0 TO KAT-1 DO
  FASTTEXT(LEFTPAD(LONG2STR(I_)+'.KOLON |', 8), 5+I_, 3);
 FOR J_ := 1 TO KOLON-1 DO
  FASTTEXT(LONG2STR(J_)+'.KLN', 3, 5+6*J_);
 FASTTEXT(CHARSTR('-',6*(KOLON-1)), 4, 11);
 REPEAT
  ES.PROCESSSELF;
 UNTIL ES.GETLASTCOMMAND=CCDONE;
 ES.DONE;
END;


PROCEDURE KATKOLONGMATRISHESAPLA;
 VAR
  I, J, K, L : WORD;
  R : REAL;
BEGIN
 FOR I:=0 TO KAT-1 DO
  BEGIN
   KATKOLGM[I].INIT( 6*KOLON, 6*KOLON,
                     SIZEOF(DOUBLE),
                     'KKGM'+LONG2STR(I)+'.MTX',
                     10240,
                     LRANGECHECK+LKEEPDISKCURRENT);
   FOR J:=1 TO KOLON DO
    BEGIN
     FOR K:=1 TO 6 DO
      FOR L:=1 TO 6 DO
       BEGIN
        R := UI^[I, J][K, L];
        KATKOLGM[I].SETA((J-1)*6+K-1, (J-1)*6+L-1, R);
        FASTWRITE(PAD(LONG2STR(I)+','+LONG2STR(J)+','+LONG2STR(K)+','+LONG2STR(L), 80), 1, 1
);
       END;
    END;
   KATKOLGM[I].STOREA('KKGM'+LONG2STR(I)+'.MTX');
   KATKOLGM[I].CLOSEA;
   KATKOLGM[I].DONE;
  END;
ND;

CEDURE SABITLERIOKU;
GIN
S.INIT(20, 8, 60, 15);
```

```
    ES.AddW        LD(' Kolon        = ', 5, 4, '99', 5, 21, 0, 0, MaxKolon, Kolon);
    ES.AddWordField('   Kat Sayisi = ', 4, 4, '99', 4, 21, 0, 0, MaxKat, Kat);
    ES.AddDblField ('            E = ', 5, 4, '!!!!!!!!!!!', 5, 21, 0, 0, 9E300, 0, E);
    ES.ProcessSelf;
    ES.Done;
  END;


Procedure InitVariables;
 BEGIN
   E := 2E5;
   Kolon := 4;
   Kat := 4;
   FillChar(A, SizeOf(A), 0);
   FillChar(B, SizeOf(B), 0);
   FillChar(F, SizeOf(F), 0);
   FillChar(I, SizeOf(I), 0);
   FillChar(H, SizeOf(H), 0);
   FillChar(A_, SizeOf(A_), 0);
   FillChar(B_, SizeOf(B_), 0);
   FillChar(F_, SizeOf(F_), 0);
   FillChar(II, SizeOf(II), 0);
   FillChar(L_, SizeOf(L_), 0);
   For I_:=0 to MaxKat do H[I_]:=1;
   New(UI);
   FillChar(UI^, SizeOf(UI^), 0);
   New(UI_);
   FillChar(UI_^, SizeOf(UI_^), 0);
   New(V);
   FillChar(V^, SizeOf(V^), 0);
   Randomize;
   For I_:=0 to MaxKat-1 do
    For J_:= 1 to MaxKolon do
     BEGIN
      A[I_, J_] := Random(100)/101;
      IF A[I_, J_] < 0.05 THEN A[I_, J_] := 0.44;
      B[I_, J_] := Random(100)/101;
      IF B[I_, J_] < 0.05 THEN B[I_, J_] := 0.44;
     END;
   For I_:=0 to MaxKat-1 do
    For J_:= 1 to MaxKolon-1 do
     BEGIN
      A_[I_, J_] := Random(100)/101;
      IF A_[I_, J_] < 0.05 THEN A_[I_, J_] := 0.77;
      B_[I_, J_] := Random(100)/101;
      IF B_[I_, J_] < 0.05 THEN B_[I_, J_] := 0.77;
     END;
 END;

Procedure A_N_N_Hesapla;
 Var
  I__, I_, J_, I, J, K : Word;
 BEGIN
  For I:=0 to Kat-1 do
   BEGIN
    For J:=1 to Kolon-1 do
     For K:=J to Kolon-1 do
      BEGIN
       KirisGKM.Init( 6, 6,
                      SizeOf(Double),
                      'KG'+HexB(I)+HexB(J)+HexB(K)+'.MTX',
                      10240,
                      LRangeCheck+LKeepDiskCurrent);
       IF J=K THEN
        BEGIN
```

```
        FOR I_:=1 TO 6 DO
         FOR J_:=1 TO 6 DO
          KirisGKM.SetA(I_-1, J_-1, UI_^[I, K][I_, J_]);
       END ELSE
       BEGIN
        FillChar(DummyUI, SizeOf(DummyUI), 0);
        FOR I__:=1 TO 6 DO DummyUI[I__,I__]:=1;
        FOR I__:=K TO J DO
         Mult(DummyUI, UI_^[I,I__], DummyUI);
        FOR I_:=1 TO 6 DO
         FOR J_:=1 TO 6 DO
          KirisGKM.SetA(I_-1, J_-1, DummyUI[I_, J_]);
       END;
      KirisGKM.StoreA('KG'+HexB(I)+HexB(J)+HexB(K)+'.MIX');
      KirisGKM.Done;
     END;
   END;
 END;

Procedure AHesapla;
 Var
  I__, I_, J_, I, J, K : Word;
  R : Real;
 BEGIN
  FOR I:=0 TO KAT-1 DO
   BEGIN
    AMatris.Init( 6*(Kolon-1), 6*(Kolon-1),
                  SizeOf(Double),
                  'A'+HexB(I)+'.MIX',
                  10240,
                  LRangeCheck+LKeepDiskCurrent);
{    FOR J:=1 TO KOLON-1 DO
     FOR K:=J TO KOLON-1 DO
      BEGIN
       KirisGKM.Init( 6, 6,
                      SizeOf(Double),
                      'KG'+HexB(I)+HexB(J)+HexB(K)+'.MIX',
                      10240,
                      LRangeCheck+LKeepDiskCurrent);
       KirisGKM.LoadA( 'KG'+HexB(I)+HexB(J)+HexB(K)+'.MIX',
                       10240,
                       LRangeCheck+LKeepDiskCurrent);
       FOR I_:=1 TO 6 DO
        FOR J_:=1 TO 6 DO
         BEGIN
          FastWrite(Pad(Long2Str(I)+','+Long2Str(J)+','+Long2Str(K)+','+Long2Str(I_)+','+Long2St
r(J_), 80), 1, 1, 7);ReadLn;
          KirisGKM.RetA(I_, J_, R);
          FastWrite(Pad(Form('#######.##', R), 80), 1, 1, 7);ReadLn;
          AMatris.SetA((J-1)*6+I_-1, (K-1)*6+J_-1, R);
         END;
       KirisGKM.Done;
      END;}
    FOR J:=0 TO (KOLON-1)*6-1 DO
     FOR K:=0 TO (KOLON-1)*6-1 DO
      AMatris.SetA(J, K, R);
    AMatris.StoreA('A'+HexB(I)+'.MIX');
    AMatris.CloseA;
    AMatris.Done;
   END;
 END;

Procedure BHesapla;
 Var
  I__, I_, J_, I, J, K : Word;
  R : Real;
```

```
 BEGIN
  FOR I:=0 TO KAT-1 DO
   BEGIN
    BMATRIS.INIT( 6*(KOLON-1), 6*(KOLON-1),
                  SIZEOF(DOUBLE),
                  'B'+HEXB(I)+'.MTX',
                  10240,
                  LRANGECHECK+LKEEPDISKCURRENT);
{   FOR J:=2 TO KOLON-1 DO
    FOR K:=J TO KOLON-1 DO
     BEGIN
      KIRISGKM.INIT( 6, 6,
                     SIZEOF(DOUBLE),
                     'KG'+HEXB(I)+HEXB(J)+HEXB(K)+'.MTX',
                     10240,
                     LRANGECHECK+LKEEPDISKCURRENT);
      KIRISGKM.LOADA( 'KG'+HEXB(I)+HEXB(J)+HEXB(K)+'.MTX',
                      10240,
                      LRANGECHECK+LKEEPDISKCURRENT);
      FOR I_:=1 TO 6 DO
       FOR J_:=1 TO 6 DO
        BEGIN
         FASTWRITE(PAD(LONG2STR(I)+','+LONG2STR(J)+','+LONG2STR(K)+','+LONG2STR(I_)+','+LONG
R(J_), 80), 1, 1, 7);READLN;
         KIRISGKM.RETA(I_, J_, R);
         FASTWRITE(PAD(FORM('#######.##', R), 80), 1, 1, 7);READLN;
         AMATRIS.SETA((J-1)*6+I_-1, (K-1)*6+J_-1, R);
        END;
      KIRISGKM.DONE;
     END;}
   FOR J:=0 TO (KOLON-1)*6-1 DO
    FOR K:=0 TO (KOLON-1)*6-1 DO
     BMATRIS.SETA(J, K, R);
   BMATRIS.STOREA('B'+HEXB(I)+'.MTX');
   BMATRIS.CLOSEA;
   BMATRIS.DONE;
  END;
 END;

PROCEDURE VHESAPLA;
 BEGIN
  IF (KOLON<1) OR (KOLON>MAXKOLON) OR (KAT<1) OR (KAT>MAXKAT) THEN
   BEGIN
    MASSAGE('KOLON VEYA KOLON SAYISI HATALI !');
    EXIT;
   END;
  FOR I_:=0 TO KAT-1 DO
   FOR J_:= 1 TO KOLON-1 DO
    BEGIN
     FASTWRITE('V HESAPLANIYOR ...('+LONG2STR(I_)+','+LONG2STR(J_)+')', 1, 1, 7);
     V^[I_, J_][1,1] := 0;  { 1.SATIR }
     V^[I_, J_][1,2] := 0;
     V^[I_, J_][1,3] := 0;
     V^[I_, J_][1,4] := 0;
     V^[I_, J_][1,5] := 0;
     V^[I_, J_][1,6] := 0;

     V^[I_, J_][2,1] := 0;  { 2.SATIR }
     V^[I_, J_][2,2] := 0;
     V^[I_, J_][2,3] := 0;
     V^[I_, J_][2,4] := 0;
     V^[I_, J_][2,5] := 0;
     V^[I_, J_][2,6] := 0;

     V^[I_, J_][3,1] := 0;  { 3.SATIR }
     V^[I_, J_][3,2] := 0;
```

```
        V^[I_, J_][3,3] := 1;
        V^[I_, J_][3,4] := 0;
        V^[I_, J_][3,5] := 0;
        V^[I_, J_][3,6] := 0;

        V^[I_, J_][4,1] := 0;   { 4.SATIR }
        V^[I_, J_][4,2] := 0;
        V^[I_, J_][4,3] := 0;
        V^[I_, J_][4,4] := 1;
        V^[I_, J_][4,5] := 0;
        V^[I_, J_][4,6] := 0;

        V^[I_, J_][5,1] := 0;   { 5.SATIR }
        V^[I_, J_][5,2] := 0;
        V^[I_, J_][5,3] := 0;
        V^[I_, J_][5,4] := 0;
        V^[I_, J_][5,5] := 1;
        V^[I_, J_][5,6] := 0;

        V^[I_, J_][6,1] := 0;   { 6.SATIR }
        V^[I_, J_][6,2] := 0;
        V^[I_, J_][6,3] := -(4*E)*((I[I_+1,J_]/H[I_+1])+(I[I_,J_]/H[I_]));
        V^[I_, J_][6,4] := 0;
        V^[I_, J_][6,5] := 0;
        V^[I_, J_][6,6] := 0;

      END;
  END;

  PROCEDURE KATDURUMMATRISHESAPLA;
   VAR
    I, J, K, L : WORD;
    R : REAL;
   BEGIN
    FASTWRITE(PAD('',80), 1, 1, 7);
    FOR I:=0 TO KAT-1 DO
     BEGIN
      KATDURM[I].INIT( 6*KOLON, 6*KOLON,
                       SIZEOF(DOUBLE),
                       'KD'+LONG2STR(I)+'.MIX',
                       10240,
                       LRANGECHECK+LKEEPDISKCURRENT);
      FOR J:=1 TO KOLON DO
       BEGIN
        FOR K:=1 TO 6 DO
         FOR L:=1 TO 6 DO
          BEGIN
           R := V^[I, J][K, L];
           KATDURM[I].SETA((J-1)*6+K-1, (J-1)*6+L-1, R);
           FASTWRITE(PAD(LONG2STR(I)+','+LONG2STR(J)+','+LONG2STR(K)+','+LONG2STR(L), 80), 1, 1,
7);
          END;
       END;
      KATDURM[I].STOREA('KD'+LONG2STR(I)+'.MIX');
      KATDURM[I].CLOSEA;
      KATDURM[I].DONE;
     END;
   END;


  PROCEDURE HESAPLA;
   BEGIN
    UIHESAPLA;
    KATKOLONMATRISHESAPLA;
    FASTWRITE(PAD('A M M HESAPLANIYOR [illegible]', 80), 1, 1, 7);
```

```
      A;
   FASTWRITE(PAD('B HESAPLANIYOR....',80), 1, 1, 7);
   BHESAPLA;
   FASTWRITE(PAD('V HESAPLANIYOR....',80), 1, 1, 7);
   VHESAPLA;
   FASTWRITE(PAD('KAT DURUM MATRISLERI HESAPLANIYOR....',80), 1, 1, 7);
   KATDURUMMATRISHESAPLA;
  END;

PROCEDURE YUKLEMEOKU;
 VAR
  S : STRING;
 BEGIN
  FOR I_:=0 TO KAT-1 DO
   FOR J_:=1 TO KOLON DO
    BEGIN
     FASTWRITE(PAD('KAT : '+PAD(LONG2STR(I_),3)+'  '+'KOLON : '+PAD(LONG2STR(J_),3)+'  YUKLEME
', 80), 1, 1, 7);
    END;
 END;



BEGIN
  CLRSCR;
{  FILLWORD(PTR(VIDEOSEGMENT, 0)^, 25*80, $07B1);}

  STATUS := INITMENU(MNU, MENUCOLORS);
  IF STATUS <> 0 THEN BEGIN
    WRITELN('ERROR INITIALIZING MENU: ', STATUS);
    HALT(1);
  END;

  INITVARIABLES;

  {SET UP USER HOOKS}
  MNU.SETERRORPROC(ERRORHANDLER);
  MNU.SETCUSTOMSTRINGPROC(CUSTOMIZEITEMSTRING);
  MENUCOMMANDS.SETHELPPROC(DISPLAYHELP);

  {$IFDEF USEMOUSE}
  IF MOUSEINSTALLED THEN
    WITH MENUCOLORS DO BEGIN
      {ACTIVATE MOUSE CURSOR}
      SOFTMOUSECURSOR($0000, (COLORMONO(MOUSECOLOR, MOUSEMONO) SHL 8)+
                              BYTE(MOUSECHAR));
      SHOWMOUSE;
      {ENABLE MOUSE SUPPORT}
      MENUCOMMANDS.CPOPTIONSON(CPENABLEMOUSE);
    END;
  {$ENDIF}

  MNU.DRAW;
  REPEAT
   MNU.PROCESS;
   IF MNU.GETLASTCOMMAND=CCSELECT THEN
    CASE MNU.MENUCHOICE OF
     MICIKIS2:       EXITPROG := TRUE;
     MISABITLER3:    SABITLERIOKU;
     MICERCEVE7:     CERCEVE7OKU;
     MICERCEVE6:     CERCEVE6OKU;
     MIKAT8:         KATYUKSEKLIKLERIOKU;
     MIHESAPLA5:     HESAPLA;
```

```
      M       NIO:       AtaletGoster;
      MiKolon11:       UIGoster;

      MiCerceve13:   cerceve7Oku_;
      MiCerceve14:   cerceve6Oku_;
      MiKiris15:       KirisBoylariOku_;
      MiKiris16:       KirisKesitGoster_;
      MiKiris17:       AtaletGoster_;
      MiKiris18:       UIGoster_;
    END;

  UNTIL ExitProg;
  Mnu.Done;

  {$IFDEF UseMouse}
  HideMouse;
  {$ENDIF}
  ClrScr;
END.
```