



GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ

TASARIM VE İMALAT MÜHENDİSLİĞİ

STIFNESS-DIRECT METHOD İLE ÇEŞİTLİ ELEMAN TİPLERİ İÇİN LİNEER ANALİZİN BİLGİSAYAR YARDIMIYLA HESAPLANMASI (QBASIC DİLİNDE PAKET PROGRAM)



HAZIRLAYAN

YUSUF MANSUROĞLU

YÖNETEN

YRD. DOÇ .DR. HASAN KURTARAN

SONLU ELEMANLAR ANALİZİ II

Günümüzde her alanda gittikçe artan bilgisayar kullanımı ve buna paralel olarak onların verimli kullanımı için yapılan çalışmaların artması ile , kimi zaman kullanıcılarını çileden çıkarsa da , bilgisayarlar insanoğluna sunduğu sayısız kolaylık ve sürat , istenilen her bilgiye ulaşma imkanı gibi fonksiyonlarla hayatımızın vazgeçilmezlerinden olmuştur.

Bu nedenle hiçbir mühendislik dalının bilgisayarlardan bihaber olması , ve uzak durması düşünülemez. Mühendisliğin hemen hemen her uygulamasında ortaya çıkan hesaplamalar; mühendislerin günlerce sayılarla boğuşmasına , insan canın söz konusu olduğu yerlerde hata olasılığı en aza indirmek için elde edilen değerlerin defalarca gözden geçirilmesine sebep olmakta ; dolayısıyla bize bahşedilen zamanın zaten kısa oluşu işleri büsbütün çıkmaza sokmaktaydı.

Tam bu aşamada mühendislerin imdadına yetişen bilgisayarlarda, programlama dilleriyle yapılan paket programlar sayesinde günlerle ifade edilen bu uzun hesaplamalara dakikalarla ifade edilen zaman zarflarında ulaşma olanağını sağlamıştı.

Mühendislerin ihtiyacına cevap verebilecek bir programın hazırlanması , mühendislik bilgisine sahip bir programcıya gerek duyulmasına sebep olmaktaydı. Bu ihtiyacı karşılamak yine bizzat mühendislerin kendilerine düşmüş ve çeşitli programlama dillerini öğrenmelerine ve mühendis namzetlerine öğrenmelerine vesile olmuştu.

Bu amaçla hazırlanan bu çalışma stiffness-directmethod yardımıyla projelendirilmesini sağlamakta ve bu husustaki açığı giderecek düzeyde olmasa da öncü olması ümidiyle.

PROGRAMIN ÇALIŞMA PRENSİBİ

Program çalışma mantığı STIFNESS-DIRECT METHOD'a dayanmaktadır. Programın çalıştırılması ile ilk başta kullanıcıdan eleman sayısı ve node sayısı istenmektedir. Bunun amacı arka planda oluşturulacak olan Elementary ve Global Stifness Matrisler için (node)x(node) boyutunda matrisler oluşturmaktır.

Her Bir Eleman İçin Stifness Matrisler Hesaplanır:

Çeşitli eleman tipleri için Yay Sabiti (stiffness): k (lb/in, N/m, N/mm) değerlerini hesaplamakta ve bu değerler ile öncelikle her bir eleman için stifness matris hesaplanır. Aslında her bir eleman için 2x2 boyutunda Elementary Stifness matrislerin oluşturulması yeterlidir. Ancak daha sonra her bir elemanın için hesaplanan Elementary Stifness matrislerin Global Stifness matriste monte edilmesi zor olacaktır.

$$\begin{array}{ccc}
 \begin{array}{cc} u_1 & u_2 \\ \hat{e} k_1 & -k_1 \hat{u} \\ \hat{e} -k_1 & k_1 \hat{u} \end{array} &
 \begin{array}{cc} u_2 & u_3 \\ \hat{e} k_2 & -k_2 \hat{u} \\ \hat{e} -k_2 & k_2 \hat{u} \end{array} &
 \begin{array}{cc} u_3 & u_4 \\ \hat{e} k_3 & -k_3 \hat{u} \\ \hat{e} -k_3 & k_3 \hat{u} \end{array}
 \end{array}$$

O nedenle her bir eleman için hesaplanan matrisler global matrisin boyutuna uygun olacak şekilde genişletilir. Yay sabitleri değerleri uygun node'lara yerleştirilir ve diğer node'lara "0" (sıfır) değerleri yerleştirilir.

$$\begin{array}{ccc}
 \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \\ \hat{e} k_1 & -k_1 & 0 & 0 \\ \hat{e} -k_1 & k_1 & 0 & 0 \\ \hat{e} 0 & 0 & 0 & 0 \\ \hat{e} 0 & 0 & 0 & 0 \end{array} &
 \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \\ \hat{e} 0 & 0 & 0 & 0 \\ \hat{e} 0 & k_2 & -k_2 & 0 \\ \hat{e} 0 & -k_2 & k_2 & 0 \\ \hat{e} 0 & 0 & 0 & 0 \end{array} &
 \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \\ \hat{e} 0 & 0 & 0 & 0 \\ \hat{e} 0 & 0 & 0 & 0 \\ \hat{e} 0 & 0 & k_3 & -k_3 \\ \hat{e} 0 & 0 & -k_3 & k_3 \end{array}
 \end{array}$$

Elementary Stifness matrisleriler toplanarak Global Stifness Hesaplanır:

Bundan sonra işlemiz matrislerin toplanmasına gelmektedir. Bunun için birkaç FOR NEXT döngüsü yeterli olacaktır.

```

FOR i = 1 TO node
FOR j = 1 TO node
FOR n = 1 TO element

```

```

IF i = nx(n) AND j = nx(n) THEN K(i, j) = kk(n): GOTO 1
IF i = ny(n) AND j = ny(n) THEN K(i, j) = kk(n): GOTO 1
IF i = nx(n) AND j = ny(n) THEN K(i, j) = -kk(n): GOTO 1
IF i = ny(n) AND j = nx(n) THEN K(i, j) = -kk(n): GOTO 1

```

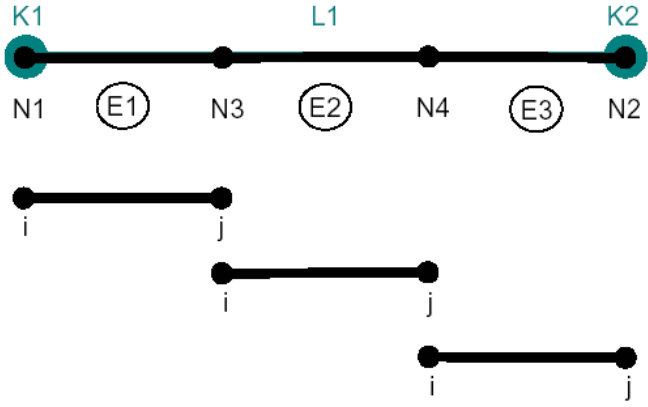
K(i, j) = 0

1 A(i, j) = A(i, j) + K(i, j) 'Global stifness matrix

NEXT n: NEXT j: NEXT i

$$K = \begin{bmatrix}
 \begin{array}{cc|cc}
 u_1 & u_2 & u_3 & u_4 \\
 \hline
 k_1 & -k_1 & 0 & 0 \\
 \hline
 -k_1 & k_1 + k_2 & -k_2 & 0 \\
 \hline
 0 & -k_2 & k_2 + k_3 & -k_3 \\
 \hline
 0 & 0 & -k_3 & k_3 \\
 \hline
 \end{array}
 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 \\ 0 & 0 & -k_3 & k_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{bmatrix}$$



Yerdeğiştirme Vektörü Bulunur:

Son olarak Global Stiffness Matris oluşturulduktan sonra program kullanıcıya hangi node'larda yerdeğiştirmelerin olduğu haahilerinde olmadığı bilgisini istemektedir. Bu sayede $u =$ (element nodal) yer deęiştirme vektörü elde edilir.



$k =$ (element) stiffness matris
 $u =$ (element nodal) yer deęiştirme vektörü
 $f =$ (element nodal) kuvvet vektörü

Kuvvet Vektörü Bulunur:

Program son olarak kullanıcıdan hangi node'larda kuvvetlerin uygulandığı bilgisi istenir ve böylece son vektörümüz $f =$ (element nodal) kuvvet vektörü elde edilir.



Sınır Şartlar Uygulanır:

Yer deęiştirmelerin "0" (sıfır) olduğu satır ve sütunlar silinerek yeni bir matris elde edilir.

$$\begin{bmatrix} \textcircled{u_1} & u_2 & u_3 & \textcircled{u_4} & \textcircled{u_5} \\ -200 & 200 & 0 & 0 & 0 \\ -200 & 300 & -100 & 0 & 0 \\ 0 & -100 & 750 & -200 & -450 \\ 0 & 0 & 200 & 200 & 0 \\ 0 & 0 & 450 & 0 & 450 \end{bmatrix} \times \begin{bmatrix} 0 \\ u_2 \\ u_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ P \\ F_4 \\ F_5 \end{bmatrix}$$

Gauss Jordan ile Matrisin Tersini Bulunuyor:

$\mathbf{K}\mathbf{x}=\mathbf{F}$ denklem sisteminde \mathbf{u} = (element nodal) yer deęiřtirme vektörünün elde edilmesi için \mathbf{k} = (element) stiffness matris'in tersinin \mathbf{f} = (element nodal) kuvvet vektörü ile çarpılması gerekir.

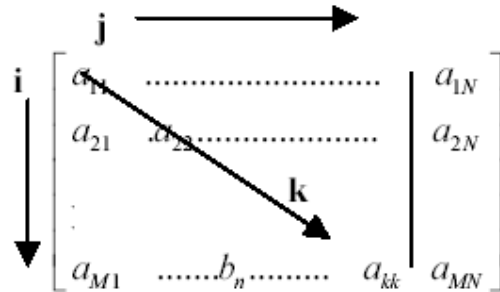
$$\mathbf{U}=\mathbf{K}^{-1}\mathbf{x}\mathbf{F}$$

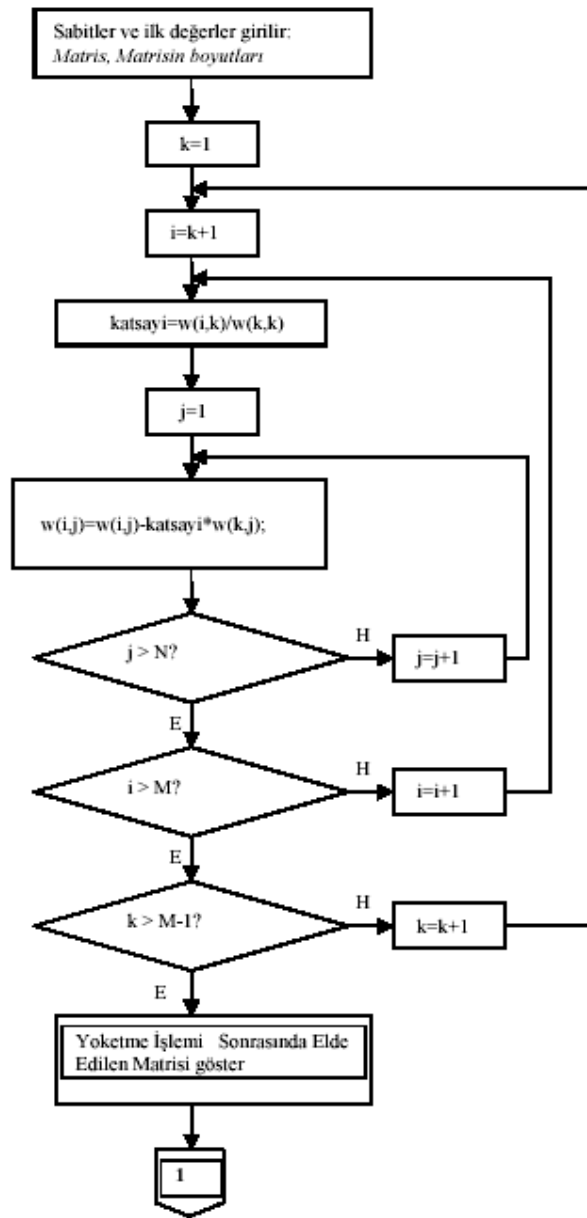
Sınır Şartlar uygulandıktan sonra elde edilen bu yeni matrisin Gauss-Jordan yöntemi ile ters elde edilir. Bu yöntemde de, denklemler sırayla uygun bir şekilde birbirlerinden çıkarılarak, köşegenin altındaki katsayılar adım adım yok edilir. Matris köşegeninin altında kalan üçgen tamamen 0 yapılır.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

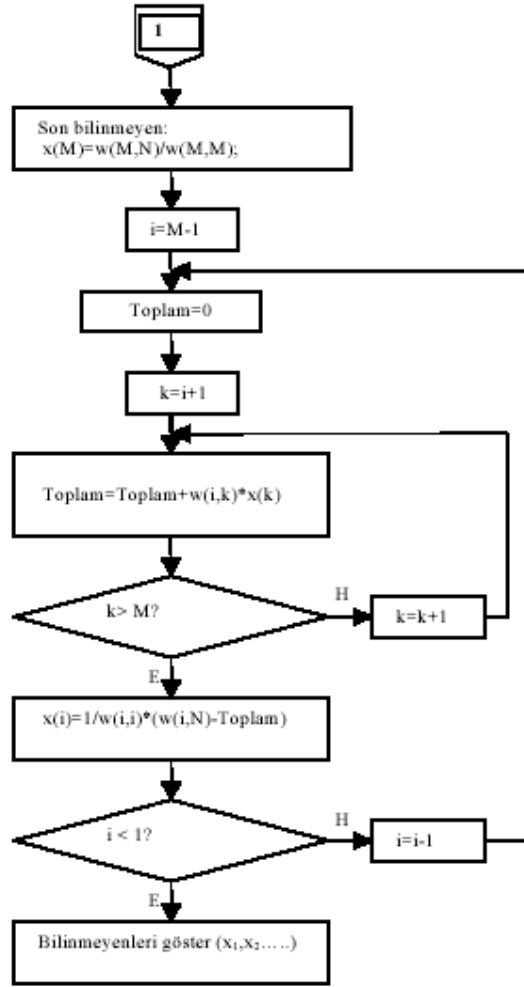
daha sonra en alt denklemden başlayıp (örneğin burada $a_{33} x_3=b_3 \rightarrow x_3= b_3/a_{33}$) bulunan bilinmeyenler geriye doğru yerine koyularak tüm bilinmeyen değerler bulunur. Denklemleri katsayılarla çarpıp bölerken, satır sütun işlemlerini önce katsayı matrisinde sonra da eşitlik vektöründe tekrar yapmamak için eşitlik vektörü de katsayı matrisinin yanına eklenerek bir defada hesaplama yapılır. Bu matris sağa genişlemiş matris olarak adlandırılır. Satır-sütun işlemleri bu matris üzerinde yapılır.

Gauss-Jordan yönteminde yok etme işlemi, pivot elemanlar seçilerek yapılır. Örneğin a_{11} elemanı pivot seçilerek a_{21} , a_{31} , a_{41} elemanları yok edilir, daha sonra a_{22} elemanı pivot seçilerek a_{32} ve a_{42} elemanları, benzer şekilde a_{33} pivot seçilerek a_{43} elemanı yok edilir. İşlem aşamalarının birinde a_{ii} sıfır olsaydı, geriye doğru yerine koyma işlemi sırasında sıfıra bölme hatası oluşacağı için işlem sonuçsuz kalacaktı. Bu nedenle bir satırdaki pivot sıfırsa, kendinden sonraki satırda aynı sıradaki elemana bakılır. O da sıfırsa bir sonraki satırdaki elemana bakılır. Uygun bir pivot bulununca satırlar yer deęiřtirilir.



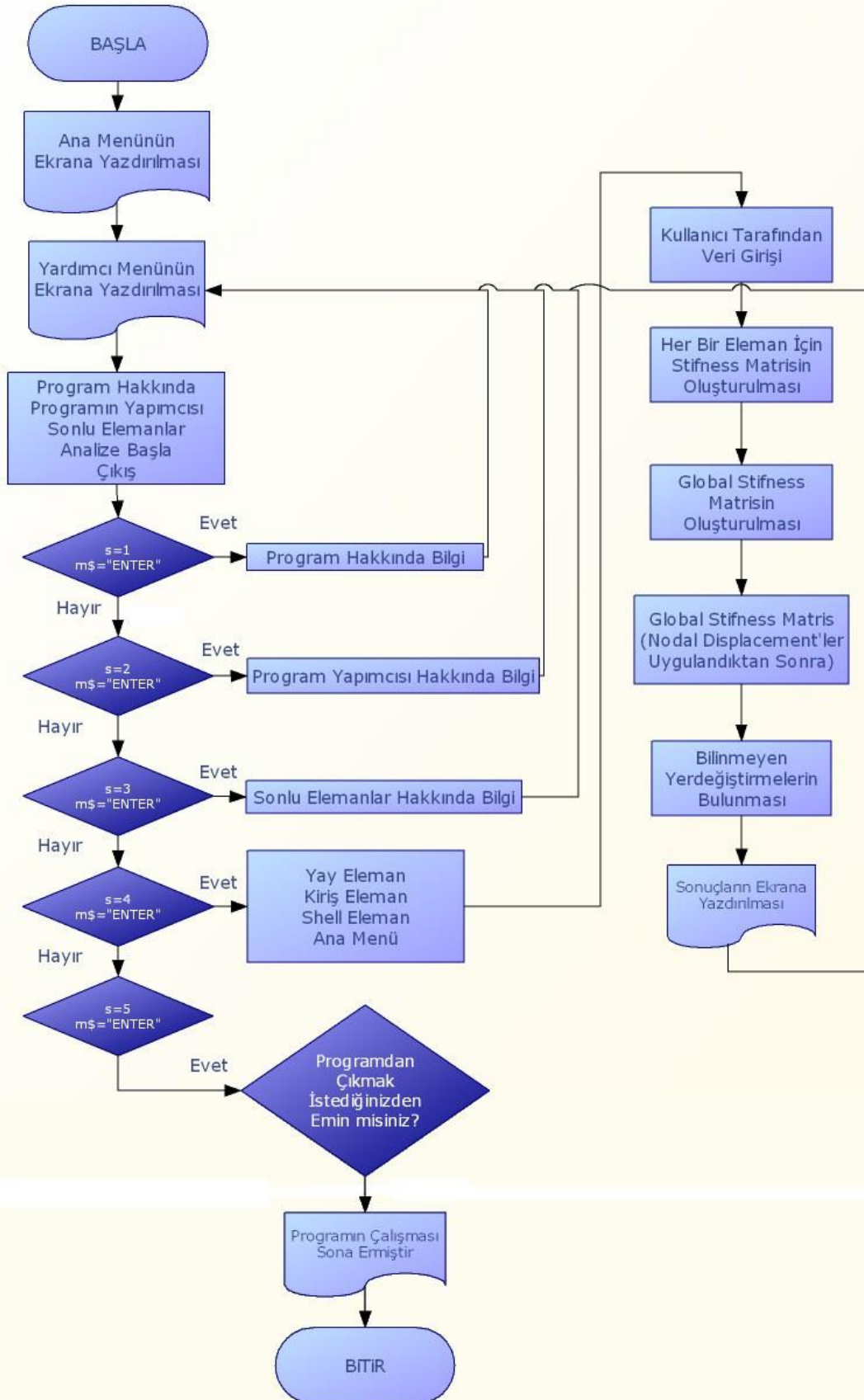


Gauss-Jordan Yönteminin Algoritması (Yok etme yordamı)



(Devamı) Gauss-Jordan Yönt. Algoritması (Bilinmeyenlerin geriye doğru çözümü)

ANALİZCİ v1.0 BASIC PROGRAMI AKIŞ ŞEMASI



Q BASIC DİLİNDEKİ PROGRAM

```
DECLARE SUB MENU2 ()
DECLARE SUB MENU ()
DECLARE SUB BACKGROUND ()

CLS
'TOPLAM SATIR UZUNLUGU (x) 80 BiRiM

COLOR 1
FOR Y = 4 TO 20
LOCATE Y, 23: PRINT STRING$(34, 219)
NEXT Y

Y = 5
COLOR 15, 1
LOCATE Y, 25: PRINT STRING$(30, 4)
LOCATE Y + 2, 29: PRINT "A N A L i Z C i v1 . 0"
LOCATE Y + 4, 32: PRINT "Yusuf MANSUROGLU"
LOCATE Y + 5, 35: PRINT "ARALIK 2005"
LOCATE Y + 7, 25: PRINT STRING$(30, 4)

COLOR 3
LOCATE Y + 9, 25: PRINT STRING$(30, "-")
LOCATE Y + 10, 28: PRINT "SONLU ELEMANLAR PROGRAMI"
LOCATE Y + 11, 25: PRINT STRING$(30, "-")

SLEEP 1
COLOR 30, 1
LOCATE 19, 46: PRINT "ENTER >>"

FOR j = 1 TO 32: YOK$ = INKEY$: NEXT
DO: LOOP UNTIL INKEY$ <> ""

W: CALL BACKGROUND

CLS
COLOR 7, 6: LOCATE 25, 50: PRINT "YUSUF MANSUROGLU 2005"

COLOR 1
LOCATE 1, 1: PRINT STRING$(80, 219)

COLOR 15, 1
LOCATE 1, 29: PRINT "A N A L İ Z C İ 1.0"

m$(1) = "PROGRAM HAKKINDA":
m$(2) = "PROGRAM YAPIMCISI":
m$(3) = "SONLU ELEMANLAR":
m$(4) = "ANALİZE BASLA":
m$(5) = "CIKIS"

A$(1) = "YAY ELEMAN":
```

```
A$(2) = "KİRİŞ ELEMAN":  
A$(3) = "SOLİD ELEMAN":  
A$(4) = "ANA MENU"
```

```
COLOR 0: FOR SM = 8 TO 15: LOCATE SM, 18: PRINT STRING$(50, 219): NEXT'GÖLGE  
EFEKTİ  
COLOR 3: FOR s = 8 TO 14: LOCATE s, 16: PRINT STRING$(50, 219): NEXT'YEŞİL ZEMİN
```

```
'MENÜ GÖLGE EFEKTLİ  
COLOR 4, 3  
LOCATE 9, 30: PRINT "PROGRAM HAKKINDA":  
LOCATE 10, 30: PRINT "PROGRAM YAPIMCISI"  
LOCATE 11, 30: PRINT "SONLU ELEMANLAR"  
LOCATE 12, 30: PRINT "ANALİZE BASLA"  
LOCATE 13, 30: PRINT "CIKIS"
```

```
'ANA MENÜ YAZISI  
COLOR 2: LOCATE 7, 16: PRINT STRING$(50, 219)  
COLOR 15, 2: LOCATE 7, 36: PRINT "ANA MENU"  
COLOR 3: LOCATE 8, 16: PRINT STRING$(50, 219)
```

```
'ÇEVREDEKİ KIRMIZI KUTUCUK  
COLOR 4, 3:  
FOR R = 8 TO 13: LOCATE R, 16: PRINT CHR$(179):  
LOCATE R, 65: PRINT CHR$(179): NEXT  
FOR n = 17 TO 64: LOCATE 14, n: PRINT CHR$(196): NEXT  
FOR i = 17 TO 64: LOCATE 8, i: PRINT CHR$(196): NEXT  
LOCATE 8, 16: PRINT CHR$(218): LOCATE 8, 65: PRINT CHR$(191)  
LOCATE 14, 16: PRINT CHR$(192): LOCATE 14, 65: PRINT CHR$(217)
```

```
s = 9:  
COLOR 4: LOCATE s, 17: PRINT STRING$(48, 219)'KIRMIZI ŞERİT LİGT  
COLOR 11, 4: LOCATE s, 30: PRINT m$(s - 8)
```

```
FOR i = 1 TO 32: YOKET$ = E$: NEXT
```

```
DO: E$ = INKEY$  
IF E$ = CHR$(0) + CHR$(72) THEN s = s - 1  
IF s < 9 THEN s = 13
```

```
IF E$ = CHR$(0) + CHR$(80) THEN s = s + 1  
IF s > 13 THEN s = 9
```

```
IF E$ <> "" THEN CALL MENU: COLOR 4: LOCATE s, 17: PRINT STRING$(48, 219):  
COLOR 11, 4: LOCATE s, 30: PRINT m$(s - 8)
```

```
IF E$ = CHR$(13) THEN GOTO 100  
LOOP
```

```
100
IF s = 10 THEN GOTO 200
IF s = 11 THEN GOTO 300
IF s = 12 THEN GOTO 400
IF s = 13 THEN GOTO 500
```

```
CLS : '***** PROGRAM HAKKINDA *****
```

```
CLS
```

```
COLOR 1
```

```
LOCATE 1, 1: PRINT STRING$(80, 219)
```

```
COLOR 15, 1
```

```
LOCATE 1, 29: PRINT "A N A L İ Z C İ 1.0"
```

```
COLOR 11, 4: PRINT : PRINT " PROGRAM HAKKINDA": PRINT
```

```
COLOR 7, 4
```

```
LOCATE 6
```

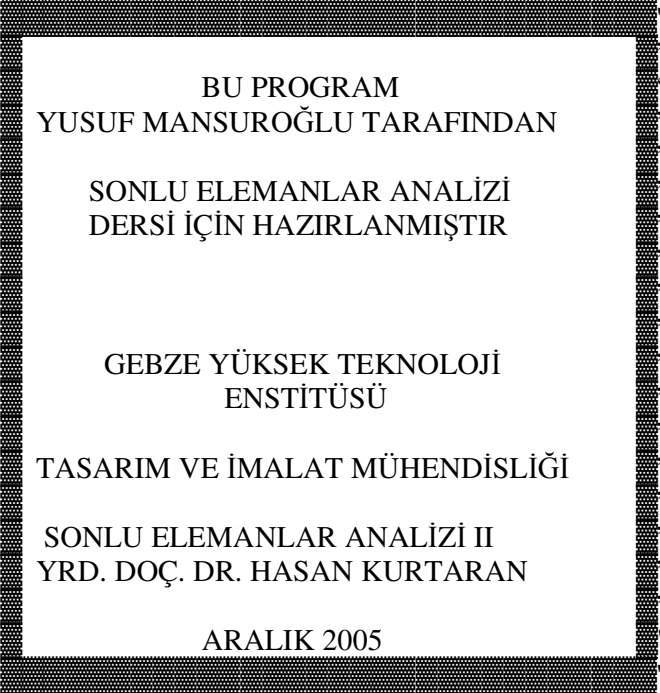
```
PRINT " Günümüzde her alanda gittikçe artan bilgisayar kullanımı"
PRINT " ve buna paralel olarak onların verimli kullanımı için yapılan"
PRINT " çalışmaların artması ile , kimi zaman kullanıcılarını çileden"
PRINT " çıkarsa da , bilgisayarlar insanoğluna sunduğu sayısız"
PRINT " kolaylık ve sürat , istenilen her bilgiye ulaşma imkanı"
PRINT " gibi fonksiyonlarla hayatımızın vazgeçilmezlerinden olmuştur."
PRINT
PRINT " Bu nedenle hiç bir mühendislik dalının bilgisayarlardan"
PRINT " bihaber olması , ve uzak durması düşünülemez"
PRINT " Mühendisliğin hemen hemen her uygulamasında ortaya çıkan"
PRINT " hesaplamalar; mühendislerin günlerce sayılarla boğuşmasına"
PRINT " insan canın söz konusu olduğu yerlerde hata olasılığı en aza"
PRINT " indirmek için elde edilen değerlerin defalarca gözden"
PRINT " geçirilmesine sebep olmakta ; dolayısıyla bize bahşedilen"
PRINT " zamanın zaten kısa oluşu işleri büsbütün çıkmaza sokmaktaydı."
PRINT " "
PRINT " Tam bu aşamada mühendislerin imdadına yetişen bilgisayarlarda"
PRINT " programlama dilleriyle yapılan paket programlar sayesinde günlerle"
PRINT " ifade edilen bu uzun hesaplamalara dakikalarla ifade edilen"
PRINT " zaman zarflarında ulaşma olanağını sağlamıştır."
PRINT
PRINT " Mühendislerin ihtiyacına cevap verebilecek bir programın hazırlanması"
PRINT " mühendislik bilgisine sahip bir programcıya gerek duyulmasına"
PRINT " sebep olmaktaydı. Bu ihtiyacı karşılamak yine bizzat mühendislerin"
PRINT " kendilerine düşmüş ve çeşitli programlama dillerini öğrenmelerine"
PRINT " ve mühendis namzetlerine öğretmelerine vesile olmuştur."
PRINT " "
PRINT " Bu amaçla hazırlanan bu çalışma Direct Stiffness Method"
PRINT " yöntemini kullanarak çeşitli eleman tipleri için lineer analizleri"
PRINT " bilgisayar yardımıyla hesaplanmasını sağlamaktadır."
```

```
LOCATE 37, 37: COLOR 9, 4: PRINT "TAMAM"
```

```
DO: LOOP UNTIL INKEY$ = CHR$(13)
GOTO W
```

```
200 : '***** PROGRAM YAPIMCISI *****
CLS
COLOR 1
FOR Y = 1 TO 50
LOCATE Y, 1: PRINT STRING$(80, 219)
NEXT Y
```

```
COLOR 15, 1
LOCATE 15, 21: PRINT , "
LOCATE 16, 21: PRINT , "
LOCATE 17, 21: PRINT , "
LOCATE 18, 21: PRINT , "
LOCATE 19, 21: PRINT , "
LOCATE 20, 21: PRINT , "
LOCATE 21, 21: PRINT , "
LOCATE 22, 21: PRINT , "
LOCATE 23, 21: PRINT , "
LOCATE 24, 21: PRINT , "
LOCATE 25, 21: PRINT , "
LOCATE 26, 21: PRINT , "
LOCATE 27, 21: PRINT , "
LOCATE 28, 21: PRINT , "
LOCATE 29, 21: PRINT , "
LOCATE 30, 21: PRINT , "
LOCATE 31, 21: PRINT , "
LOCATE 32, 21: PRINT , "
LOCATE 33, 21: PRINT , "
LOCATE 34, 21: PRINT , "
```



BU PROGRAM
YUSUF MANSUROĞLU TARAFINDAN
SONLU ELEMENLAR ANALİZİ
DERSİ İÇİN HAZIRLANMIŞTIR
GEBZE YÜKSEK TEKNOLOJİ
ENSTİTÜSÜ
TASARIM VE İMALAT MÜHENDİSLİĞİ
SONLU ELEMENLAR ANALİZİ II
YRD. DOÇ. DR. HASAN KURTARAN
ARALIK 2005

```
LOCATE 36, 43: COLOR 9, 1: PRINT "TAMAM"
DO: LOOP UNTIL INKEY$ = CHR$(13)
GOTO W
```

```
300 : '***** SONLU ELEMENLAR YÖNTEMİ *****
CLS
COLOR 1
LOCATE 1, 1: PRINT STRING$(80, 219)
```

```
COLOR 15, 1
LOCATE 1, 29: PRINT "A N A L İ Z C İ 1.0"
COLOR 11, 4: PRINT : PRINT " SONLU ELEMENLAR YÖNTEMİ": PRINT
```

```
COLOR 7, 4
LOCATE 6
PRINT " Sonlu elemanlar yöntemi fizik ve mühendislikte karşılaşılan"
PRINT " bir çok problemin çözümünde kullanılan en yaygın ve etkin"
```

```
PRINT " sayısal yöntemlerden biridir"
PRINT " Sonlu elemanlar metodu matematikçilerden ziyade daha çok"
PRINT " mühendisler tarafından geliştirilmiştir."
PRINT
PRINT " Metot ilk olarak gerilme analizi problemlerine uygulanmıştır"
PRINT " Tüm bu uygulamalarda bir büyüklük alanının hesaplanması istenmektedir"
PRINT " Gerilme analizinde bu değer deplasman alanı veya gerilme alanı"
PRINT " ısı analizinde sıcaklık alanı veya ısı akışı"
PRINT " akışkan problemlerinde ise akım fonksiyonu veya hız potansiyel fonksiyonudur"
PRINT
PRINT " Hesaplanan büyüklük, alanın almış olduğu en büyük değer veya"
PRINT " en büyük gradyen pratikte özel bir önem içerir "
PRINT " Sonlu elemanlar metodunda yapı, davranışı daha önce belirlenmiş olan"
PRINT " bir çok elemana bölünür"
PRINT " Elemanlar nod adı verilen noktalarda tekrar birleştirilirler"
PRINT " Bu şekilde cebri bir denklem takımı elde edilir. "
PRINT " Gerilme analizinde bu denklemler nodlardaki denge denklemleridir"
PRINT " incelenen probleme bağlı olarak bu şekilde yüzlerce hatta"
PRINT " binlerce denklem elde edilir"
PRINT " Bu denklem takımının çözümü ise bilgisayar kullanımını zorunlu kılmaktadır"
```

```
LOCATE 29, 37: COLOR 9, 4: PRINT "TAMAM"
DO: LOOP UNTIL INKEY$ = CHR$(13)
GOTO W
```

```
400 : GOTO 11
500 '*** ÇIKIŞ ***
```

```
COLOR 0, 0: CLS : COLOR 12: LOCATE 25, 64: PRINT " "
COLOR 1: LOCATE 1, 1: PRINT STRING$(80, 219)
COLOR 15, 1: LOCATE 1, 30: PRINT "A N A L İ Z C İ 1.0"
COLOR 5: FOR i = 8 TO 10: LOCATE i, 20: PRINT STRING$(40, 219): NEXT: COLOR 11, 5:
LOCATE 9, 23: PRINT "GERCEKTEN ÇIKMAK İSTİYOR MUSUNUZ ?"
```

```
FOR i = 1 TO 32: YOK$ = INKEY$: NEXT
DO: W$ = INKEY$
IF UCASE$(W$) = "E" THEN GOTO W51
IF UCASE$(W$) = "H" THEN GOTO W
LOOP
```

```
W51: '***** ÇIKIŞ *****
COLOR 0, 0: CLS : COLOR 12: LOCATE 25, 64: PRINT " ": END
```

```
11 '***** YAY ELEMAN İÇİN HESAPLAMALAR *****
```

```
COLOR 0, 0
CLS
```

```
COLOR 1: LOCATE 1, 1: PRINT STRING$(80, 219): COLOR 15, 1
LOCATE 1, 29: PRINT "A N A L İ Z C İ 1.0"
```

```
COLOR 5: FOR K = 10 TO 16: LOCATE K, 30: PRINT STRING$(19, 219): NEXT
COLOR 6: LOCATE 9, 30: PRINT STRING$(19, 219)
COLOR 14, 6: LOCATE 9, 36: PRINT "ELEMAN TİPİ"
```

```
COLOR 10, 5
LOCATE 11, 35: PRINT "YAY ELEMAN"
LOCATE 12, 35: PRINT "KİRİŞ ELEMAN"
LOCATE 13, 35: PRINT "SHELL ELEMAN"
LOCATE 14, 35: PRINT "ANA MENÜ"
```

```
YON = 11: LOCATE 11, 33: PRINT CHR$(16)
DO: E$ = INKEY$
IF E$ = CHR$(0) + CHR$(72) THEN YON = YON - 1
    IF YON < 11 THEN YON = 14
```

```
IF E$ = CHR$(0) + CHR$(80) THEN YON = YON + 1
    IF YON > 14 THEN YON = 11
```

```
IF E$ <> "" THEN FOR i = 11 TO 14: LOCATE i, 33: PRINT " ": NEXT: LOCATE YON, 33:
PRINT CHR$(16)
```

```
IF E$ = CHR$(13) THEN GOTO W312K
IF E$ = CHR$(27) THEN SOUND 100, 1: GOTO W3
LOOP
```

```
W312K: '** ELEMAN ADRESLEME **
IF YON = 12 THEN GOTO W311
IF YON = 13 THEN CALL MENU2: GOTO 11
IF YON = 14 THEN GOTO W
```

```
W311: 'Yay eleman
```

```
COLOR 0, 0: CLS : COLOR 12: LOCATE 25, 64: PRINT " "
```

```
LOCATE 5, 1
INPUT "ELEMAN SAYISINI GİRİNİZ =", element
INPUT "NODE SAYISINI GİRİNİZ =", node
```

```
DIM K(node, node)
DIM A(node, node)
DIM P(node)
DIM U(node, 1)
```

```
'elemanların k değerleri
```

```
'-----
```

```
PRINT
```

```
FOR i = 1 TO element
PRINT i; ". ELEMEN İÇİN k DEĞERİ ="; : INPUT kk(i)
NEXT i
GOTO 21
```

X311: 'Kiriş eleman

'elemanların Elastisite Modül değerleri

'-----

PRINT

FOR i = 1 TO element

PRINT i; ". ELEMEN İÇİN Elastisite Modülü="; : INPUT elastic(i)

NEXT i

'elemanların Alan değerleri

'-----

PRINT

FOR i = 1 TO element

PRINT i; ". ELEMEN İÇİN Kesit Alan DEĞERİ ="; : INPUT square(i)

NEXT i

'elemanların Uzunluk değerleri

'-----

PRINT

FOR i = 1 TO element

PRINT i; ". ELEMEN İÇİN Uzunluk DEĞERİ ="; : INPUT lenght(i)

NEXT i

'elemanların k değerleri (Kiriş Eleman İçin)

'-----

PRINT

FOR i = 1 TO element

kk(i) = (elastic(i) * square(i) / lenght(i))

PRINT i; ". ELEMEN İÇİN k DEĞERİ ="; : PRINT kk(i)

NEXT i

21

'connectivity table

'-----

FOR i = 1 TO element

PRINT

PRINT i; ". ELEMEN İÇİN İLK NODE DEĞERİ ="; : INPUT nx(i)

PRINT i; ". ELEMEN İÇİN İKİNCİ NODE DEĞERİ ="; : INPUT ny(i)

NEXT i

'Elementary and Global Stifness Matrix

```

'-----
FOR i = 1 TO node
FOR j = 1 TO node
FOR n = 1 TO element

IF i = nx(n) AND j = nx(n) THEN K(i, j) = kk(n): GOTO 1
IF i = ny(n) AND j = ny(n) THEN K(i, j) = kk(n): GOTO 1
IF i = nx(n) AND j = ny(n) THEN K(i, j) = -kk(n): GOTO 1
IF i = ny(n) AND j = nx(n) THEN K(i, j) = -kk(n): GOTO 1

K(i, j) = 0
1 A(i, j) = A(i, j) + K(i, j) 'Global stiffness matrix
NEXT n: NEXT j: NEXT i

PRINT " Global Stifness Matrix"; node; "x"; node
PRINT " -----"
FOR i = 1 TO node
FOR j = 1 TO node
PRINT A(i, j),
NEXT j
PRINT
NEXT i

'yer deęiřtirme matrisi
'-----

CLS

FOR i = 1 TO node
FOR j = 1 TO 1

COLOR 0, 0
CLS

COLOR 1: LOCATE 1, 1: PRINT STRING$(80, 219):
COLOR 15, 1: LOCATE 1, 30: PRINT "A N A L İ Z C İ 1.0"
COLOR 15, 1: LOCATE 5, 26: PRINT i; ". NODE İÇİN YER DEęİŐTİRME";

COLOR 2
FOR R = 9 TO 11
LOCATE R, 30: PRINT STRING$(20, 219)
NEXT
COLOR 1: LOCATE 8, 30: PRINT STRING$(20, 219)
COLOR 14, 1: LOCATE 8, 33: PRINT "YER DEİŐTİRME"

COLOR 8, 2: LOCATE 10, 42: PRINT "VAR"
COLOR 11, 2: LOCATE 10, 34: PRINT "YOK"

FOR W = 1 TO 32: YOK$ = INKE$: NEXT W

```


DO: E\$ = INKEY\$

```
IF E$ = CHR$(0) + CHR$(75) THEN COLOR 8, 2: LOCATE 10, 42: PRINT "VAR": COLOR 11,
2: LOCATE 10, 34: PRINT "YOK": U(i, j) = 0
IF E$ = CHR$(0) + CHR$(77) THEN COLOR 11, 2: LOCATE 10, 42: PRINT "VAR": COLOR 8,
2: LOCATE 10, 34: PRINT "YOK": U(i, j) = 1
IF E$ = CHR$(13) THEN GOTO W3 'ENTER tuşuna basılırsa
IF E$ = CHR$(27) THEN SOUND 100, 1: GOTO W4 'ESCAPE tuşuna basılırsa
LOOP
W3:
```

NEXT j: NEXT i

W4:

```
FOR i = 1 TO node
FOR j = 1 TO 1
```

```
COLOR 0, 0
CLS
```

```
COLOR 1: LOCATE 1, 1: PRINT STRING$(80, 219):
COLOR 15, 1: LOCATE 1, 30: PRINT "A N A L İ Z C İ 1.0"
COLOR 15, 1: LOCATE 5, 26: PRINT i; ". NODE İÇİN KUVVET DEİERİ";
```

```
COLOR 2
FOR R = 9 TO 11
LOCATE R, 30: PRINT STRING$(20, 219)
NEXT
COLOR 1: LOCATE 8, 30: PRINT STRING$(20, 219)
COLOR 14, 1: LOCATE 8, 33: PRINT "KUVVET DEİERİ"
```

```
COLOR 8, 2: LOCATE 10, 42: PRINT "VAR"
COLOR 11, 2: LOCATE 10, 34: PRINT "YOK"
```

```
FOR W = 1 TO 32: YOK$ = INKE$: NEXT W
DO: E$ = INKEY$
```

```
IF E$ = CHR$(0) + CHR$(75) THEN COLOR 8, 2: LOCATE 10, 42: PRINT "VAR": COLOR 11,
2: LOCATE 10, 34: PRINT "YOK": F(i, j) = 0
IF E$ = CHR$(0) + CHR$(77) THEN COLOR 11, 2: LOCATE 10, 42: PRINT "VAR": COLOR 8,
2: LOCATE 10, 34: PRINT "YOK": F(i, j) = 1
IF E$ = CHR$(13) AND F(i, j) = 1 THEN COLOR 15, 1: LOCATE 14, 2: PRINT i; ". NODE İÇİN
KUVVET DEİERİİRİ GİRİNİZ"; : INPUT F(i, j)
IF E$ = CHR$(13) THEN GOTO W5 'ENTER tuşuna basılırsa
IF E$ = CHR$(27) THEN SOUND 100, 1: GOTO W6 'ESCAPE tuşuna basılırsa
LOOP
```

W5:

NEXT j: NEXT i

W6: '***** HESAPLAMALAR *****

'Nodal Yerdeğiřtirmeler Uygulanıyor

IF U(node, 1) = 0 AND U(node - 1, 1) = 0 THEN nodex = node - 2: GOTO 55
IF U(node, 1) = 0 THEN nodex = node - 1 ELSE nodex = node

55

m = 0

K = 1

FOR i = 1 TO nodex

n = 0

m = m + 1

FOR j = 1 TO nodex

IF U(i, K) = 0 AND U(j, K) = 0 THEN m = m - 1: GOTO 10

IF U(i, K) = 0 THEN GOTO 10

IF U(j, K) = 0 THEN GOTO 10

n = n + 1

Ax(m, n) = A(i, j)

Ux(m, K) = U(i, K)

Fx(m, K) = F(i, K)

10 IF m < 0 THEN m = 1

NEXT j

NEXT i

'Gauss Jordan Metodu ile Global Stifness Matrix'in
'ters matrisi hesaplanıyor

'-----

530

n = m

550 '

560

570 UFL = 5.9E-39

```
580 G = 4: G = G / 3: G = G - 1
590 EPS = ABS(((G + G) - 1) + G)
600 G = 1
```

```
630 FOR j = 1 TO n
P(j) = 0
640 FOR i = 1 TO n
t = Ax(i, j)
X(i, j) = t
t = ABS(t)
```

```
650 IF t > P(j) THEN P(j) = t
660 NEXT i: NEXT j
```

```
680 FOR K = 1 TO n:
690 q = 0: j = K:
700 FOR i = K TO n
710 t = ABS(X(i, K)): IF t > q THEN q = t: j = i
720 NEXT i
730 IF q = 0 THEN q = EPS * P(K) + UFL: X(K, K) = q
740 IF P(K) > 0 THEN q = q / P(K): IF q > G THEN G = q
750 IF G <= 8 * K THEN GOTO 790
760 PRINT "Artış Faktörü g = "; G; " aşıyor "; 8 * K; " ; deneme"
770 PRINT "A'nın Kolonu "; K; " g'yi azaltmak için 1. kolona taşıyor."
```

```
780 STOP
790 P(K) = j
```

```
800 IF j = K THEN GOTO 830
810 FOR L = 1 TO n: q = X(j, L): X(j, L) = X(K, L)
820 X(K, L) = q: NEXT L
830 q = X(K, K): X(K, K) = 1
840 FOR j = 1 TO n: X(K, j) = X(K, j) / q: NEXT j
850 FOR i = 1 TO n: IF i = K THEN GOTO 890
860 q = X(i, K): X(i, K) = 0
870 FOR j = 1 TO n
880 X(i, j) = X(i, j) - X(K, j) * q: NEXT j
890 NEXT i: NEXT K
900 '
910 FOR K = n - 1 TO 1 STEP -1'
920 j = P(K): IF j = K THEN GOTO 950
930 FOR i = 1 TO n: q = X(i, K): X(i, K) = X(i, j)
940 X(i, j) = q: NEXT i
950 NEXT K
```

```

CLS
COLOR 1
LOCATE 1, 1: PRINT STRING$(80, 219)

COLOR 15, 1
LOCATE 1, 29: PRINT "A N A L İ Z C İ 1.0"
COLOR 15, 10: PRINT : PRINT " SONUÇLAR": PRINT

```

```

PRINT
PRINT "Connectivity Table"
PRINT "-----"
PRINT "ELEMEN - NODE İ (1) - NODE j(2)"
PRINT "-----"
FOR i = 1 TO element
PRINT i, nx(i), ny(i)
NEXT i

```

```

PRINT
PRINT "Global Stifness Matris"
PRINT "-----"
FOR j = 1 TO node
FOR i = 1 TO node
PRINT A(i, j),
NEXT i
PRINT
NEXT j

```

```

PRINT
PRINT "Global Stifness Matris"
PRINT "Nodal Displacement Uygulandıktan Sonra"
PRINT "-----"
FOR j = 1 TO m
FOR i = 1 TO m
PRINT Ax(i, j),
NEXT i
PRINT
NEXT j

```

```

'Ters Global Stifness x Nodal Displacement
'----- u Nodal Displacement -----
FOR i = 1 TO node
FOR j = 1 TO node
K = 1
q(i, K) = q(i, K) + X(i, j) * Fx(j, K)
NEXT j
NEXT i

```

```
PRINT
PRINT "Yer Değiştirme Değerleri"
PRINT "-----"
FOR i = 1 TO m
FOR j = 1 TO 1
PRINT q(i, j)
NEXT j
NEXT i
```

```
LOCATE 45, 37: COLOR 14, 10: PRINT "TAMAM"
DO: LOOP UNTIL INKEY$ = CHR$(13)
GOTO W
```

```
1000 END
```

```
SUB BACKGROUND
COLOR 6, 6: FOR R = 2 TO 24: LOCATE R, 1: PRINT STRING$(80, 219): NEXT
END SUB
```

```
SUB MENU
COLOR 3: FOR s = 9 TO 13: LOCATE s, 17: PRINT STRING$(48, 219): NEXT
COLOR 4, 3
LOCATE 9, 30: PRINT "PROGRAM HAKKINDA"
LOCATE 10, 30: PRINT "PROGRAM YAPIMCISI"
LOCATE 11, 30: PRINT "SONLU ELEMANLAR"
LOCATE 12, 30: PRINT "ANALİZE BASLA"
LOCATE 13, 30: PRINT "CIKIS"
END SUB
```

```
SUB MENU2
COLOR 0, 0
CLS
COLOR 15, 0
```

```
LOCATE 18, 21: PRINT , "
LOCATE 19, 21: PRINT , "
LOCATE 20, 21: PRINT , "
LOCATE 21, 21: PRINT , "
LOCATE 22, 21: PRINT , "
LOCATE 23, 21: PRINT , "
```

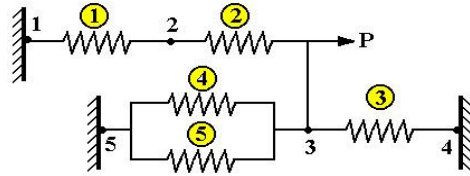


BU VERSİYONDA
SEÇTİĞİNİZ BÖLÜM AKTİF DEĞİLDİR

```
LOCATE 25, 43: PRINT "TAMAM"
DO: LOOP UNTIL INKEY$ = CHR$(13)
END SUB
```

YAY ELEMAN İÇİN ÖRNEK PROBLEM

SORU:



Verilen:

$$k_1 = 200 \text{ N/mm} \quad k_2 = 100 \text{ N/mm} \quad k_3 = 200 \text{ N/mm} \quad k_4 = 300 \text{ N/mm} \quad k_5 = 150 \text{ N/mm}$$

$$P = 5000 \text{ N}$$

İstenilen:

- 1- Connectivity table
- 2- Sınır Koşulları (Boundary conditions)
- 3- Eleman Direngenlik matrisleri (Elementary stiffness matrix) Üzerlerinde değişkenleri belirterek
- 4- Sistemin global direngenlik matrisi (Global stiffness matrix) Üzerlerinde değişkenleri belirterek
- 5- Sınır koşullarını uygulayınız
- 6- Bilinmeyen yerdeğiştirme terimlerini bulunuz (nodal displacements)

Çözüm:

Displacements:

$$u_1 = 0$$

$$u_2 \neq 0$$

$$u_3 \neq 0$$

$$u_4 = 0$$

$$u_5 = 0$$

Forces:

$$F_1 \neq 0$$

$$F_2 = 0$$

$$F_3 = 0$$

$$F_4 \neq 0$$

$$F_5 \neq 0$$

Connectivity Table:

Elemen t	Node i (1)	Node j (2)
1	1	2
2	2	3
3	3	4
4	5	3
5	5	3

Elementary Stiffness Matrices:

$$k_1 = \begin{matrix} & u_1 & u_2 \\ \begin{matrix} \hat{e} \\ \hat{e} \end{matrix} & \begin{matrix} 200 & -200 \\ -200 & 200 \end{matrix} & \begin{matrix} \hat{u} \\ \hat{u} \end{matrix} \end{matrix} \text{ N/mm}$$

$$k_2 = \begin{matrix} & u_2 & u_3 \\ \begin{matrix} \hat{e} \\ \hat{e} \end{matrix} & \begin{matrix} 100 & -100 \\ -100 & 100 \end{matrix} & \begin{matrix} \hat{u} \\ \hat{u} \end{matrix} \end{matrix} \text{ N/mm}$$

$$k_3 = \begin{matrix} & u_3 & u_4 \\ \begin{matrix} \hat{e} \\ \hat{e} \end{matrix} & \begin{matrix} 200 & -200 \\ -200 & 200 \end{matrix} & \begin{matrix} \hat{u} \\ \hat{u} \end{matrix} \end{matrix} \text{ N/mm}$$

$$k_4 = \begin{bmatrix} \epsilon & 300 & -300 \\ \hat{\epsilon} & -300 & 300 \end{bmatrix} \frac{N}{mm}$$

$$k_5 = \begin{bmatrix} \epsilon & 150 & -150 \\ \hat{\epsilon} & -150 & 150 \end{bmatrix} \frac{N}{mm}$$

Global Stiffness Matrix

$$K = \begin{bmatrix} \epsilon & 200 & -200 & 0 & 0 \\ \hat{\epsilon} & -200 & 300 & -100 & 0 \\ \epsilon & 0 & -100 & 750 & -200 \\ \hat{\epsilon} & 0 & 0 & -200 & 200 \\ \epsilon & 0 & 0 & -450 & 0 \\ \hat{\epsilon} & 0 & 0 & -450 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \epsilon & 200 & -200 & 0 & 0 \\ \hat{\epsilon} & -200 & 300 & -100 & 0 \\ \epsilon & 0 & -100 & 750 & -200 \\ \hat{\epsilon} & 0 & 0 & -200 & 200 \\ \epsilon & 0 & 0 & -450 & 0 \\ \hat{\epsilon} & 0 & 0 & -450 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ P \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$u_1=0, u_4=0, u_5=0 \rightarrow$

$$\begin{bmatrix} \textcircled{u_1} & & & \textcircled{u_4} & \textcircled{u_5} \\ \begin{bmatrix} 200 & 200 & 0 & 0 & 0 \\ -200 & 300 & -100 & 0 & 0 \\ 0 & -100 & 750 & -200 & -450 \\ 0 & 0 & 200 & 200 & 0 \\ 0 & 0 & 450 & 0 & 450 \end{bmatrix} \end{bmatrix} \begin{bmatrix} 0 \\ u_2 \\ u_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ P \\ F_4 \\ F_5 \end{bmatrix}$$

$$\begin{bmatrix} \epsilon & 300 & -100 \\ \hat{\epsilon} & -100 & 750 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ P \end{bmatrix}$$

Nodal Displacements

$$300 \cdot u_2 - 100 \cdot u_3 = 0$$

$$-100.u_2 + 750.u_3 = P$$

$$300.u_2 - 100.u_3 = 0$$

$$-100.u_2 + 750.u_3 = 5000$$

$$300.u_2 - 100.u_3 = 0$$

$$3/ \quad -100.u_2 + 750.u_3 = 5000$$

$$300.u_2 - 100.u_3 = 0$$

$$-300.u_2 + 2250.u_3 = 15000$$

$$2150u_3 = 15000$$

$$u_3 = 15000/2150$$

$$\mathbf{u_3 = 300/43 \text{ mm} = 6.9767441}$$

$$300.u_2 - 100.300/43 = 0$$

$$300.u_2 = 100.300/43$$

$$\mathbf{u_2 = 100/43 \text{ mm} = 2.3255813}$$

YAY ELEMAN İÇİN ÖRNEK PROBLEM PROGRAM YARDIMI İLE ÇÖZÜMÜ

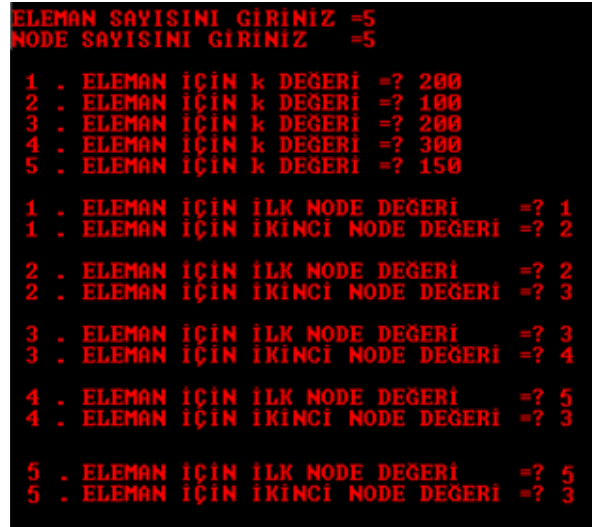
Program çalıştırıldıktan sonra ilk gelen ekranda “ENTER” tuşuna basılarak ekran Ana Menü gelmesi sağlanır.



Eleman tipi seçimi için klavyedeki yön tuşlarını kullanarak “Analyze Başla” seçeneğine gelip “ENTER” tuşuna basın. Ekranı gelen Eleman Tipi Menüden “Yay Eleman” seçilir



Bir önceki bölümde el ile çözülen örnek problemin Analizci programı ile çözümü için şu şekilde bir yol izlenir. Soruda verilenler uygun şekilde program tarafından sırası ile sorulur ve kullanıcı tarafından doğru verilerin girişi beklenir. Örnek problemimiz için programa aşağıdaki gibi verileri giriniz.



Bu işlemten sonra sıra yer değiştirmelerin girişi verilmelidir.

1. NODE İÇİN YER DEĞİŞTİRME: YOK
2. NODE İÇİN YER DEĞİŞTİRME: VAR
3. NODE İÇİN YER DEĞİŞTİRME: VAR
4. NODE İÇİN YER DEĞİŞTİRME: YOK
5. NODE İÇİN YER DEĞİŞTİRME: YOK

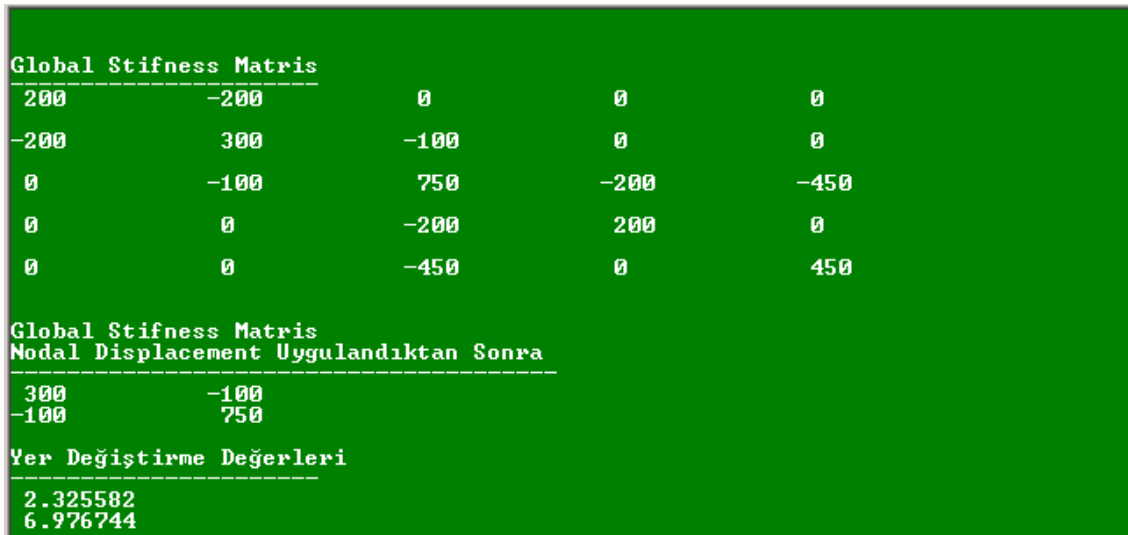


Yer deęiřtirmeler hakkında programa doęru bilgiler verildikten sonra sırada Kuvvet Deęerleri hakkında veri giriři yapılır. Problemdede sadece üçüncü node'da kuvvet uygulaması vardır. Bu node dışındaki tüm kuvvet deęerleri için "YOK" seçeneęi seçilmelidir. Üçüncü node'da "VAR" seçeneęi seçilerek program tarafından bu node'daki kuvvet deęeri için 5000N girilir.

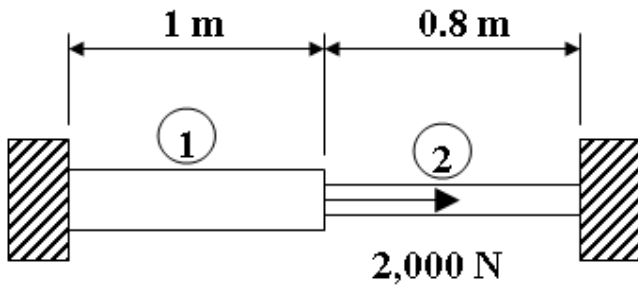
1. NODE İÇİN KUVVET DEęERİ: YOK
2. NODE İÇİN KUVVET DEęERİ: YOK
3. NODE İÇİN KUVVET DEęERİ: VAR » 5000 N
4. NODE İÇİN KUVVET DEęERİ: YOK
5. NODE İÇİN KUVVET DEęERİ: YOK



Son olarak bu veriler girildikten sonra program arka planda oluşturduęu her bir eleman için stifness matrisleri toplayarak Global Stifness Matris'i hesaplar ve Nodal Displacement'leri de uygulayarak; bu matrisin tersi alınarak kuvvet matrisi ile çarpılır ve bilinmeyen yer deęiřtirmeler bulunur.



KİRİŞ ELEMAN İÇİN ÖRNEK PROBLEM



$$A_1 = 10^{-4} \text{ m}^2$$

$$A_2 = 0.25 \times 10^{-4} \text{ m}^2$$

$$E = 70 \times 10^9 \text{ N/m}^2$$

İstenilen:

- 1- Connectivity table
- 2- Sınır Koşulları (Boundary conditions)
- 3- Eleman Direngenlik matrisleri (Elementary stiffness matrix) Üzerlerinde değişkenleri belirterek
- 4- Sistemin global direngenlik matrisi (Global stiffness matrix) Üzerlerinde değişkenleri belirterek
- 5- Sınır koşullarını uygulayınız
- 6- Bilinmeyen yerdeğiştirme terimlerini bulunuz (nodal displacements)

Çözüm:

Displacements:

$$u_1 = 0$$

$$u_2 \neq 0$$

$$u_3 = 0$$

Forces:

$$F_1 = 0$$

$$F_2 \neq 0$$

$$F_3 = 0$$

Connectivity Table:

Element	Node i (1)	Node j (2)
1	1	2
2	2	3

Elementary Stiffness Matrices:

$$\begin{array}{c}
 \begin{array}{ccc}
 & u_1 & u_2 \\
 k_1 = \frac{E \cdot A_1}{L_1} \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} 1 \\ -1 \\ 1 \\ -1 \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} = \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} \frac{E \cdot A_1}{L_1} \\ -\frac{E \cdot A_1}{L_1} \\ \frac{E \cdot A_1}{L_1} \\ -\frac{E \cdot A_1}{L_1} \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} \\
 \begin{array}{cc}
 \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} \frac{E \cdot A_1}{L_1} \\ -\frac{E \cdot A_1}{L_1} \\ \frac{E \cdot A_1}{L_1} \\ -\frac{E \cdot A_1}{L_1} \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} = \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} 7000000 \\ -7000000 \\ 7000000 \\ -7000000 \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} \\
 & u_2 & u_3 \\
 \end{array}
 \end{array}
 \quad \text{N/m}$$

$$\begin{array}{c}
 \begin{array}{ccc}
 & u_2 & u_3 \\
 k_2 = \frac{E \cdot A_2}{L_2} \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} 1 \\ -1 \\ 1 \\ -1 \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} = \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} \frac{E \cdot A_2}{L_2} \\ -\frac{E \cdot A_2}{L_2} \\ \frac{E \cdot A_2}{L_2} \\ -\frac{E \cdot A_2}{L_2} \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} \\
 \begin{array}{cc}
 \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} \frac{E \cdot A_2}{L_2} \\ -\frac{E \cdot A_2}{L_2} \\ \frac{E \cdot A_2}{L_2} \\ -\frac{E \cdot A_2}{L_2} \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} = \begin{array}{c} \hat{e} \\ \hat{e} \\ \hat{e} \\ \hat{e} \end{array} \begin{array}{c} 2187500 \\ -2187500 \\ 2187500 \\ -2187500 \end{array} \begin{array}{c} \hat{u} \\ \hat{u} \\ \hat{u} \\ \hat{u} \end{array} \\
 & u_2 & u_3 \\
 \end{array}
 \end{array}
 \quad \text{N/m}$$

Global Stiffness Matrix:

$$K = \begin{bmatrix} 7000000 & -7000000 & 0 \\ -7000000 & 9187500 & -2187500 \\ 0 & 2187500 & 2187500 \end{bmatrix} \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} = \begin{matrix} 0 \\ 2000 \\ 0 \end{matrix}$$

$$\begin{matrix} 7000000 & -7000000 & 0 \\ -7000000 & 9187500 & -2187500 \\ 0 & 2187500 & 2187500 \end{matrix} \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} = \begin{matrix} 0 \\ 2000 \\ 0 \end{matrix}$$

$$\begin{matrix} 7000000 & -7000000 & 0 \\ -7000000 & 9187500 & -2187500 \\ 0 & 2187500 & 2187500 \end{matrix} \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} = \begin{matrix} 0 \\ 2000 \\ 0 \end{matrix}$$

Global Stiffness Matrix (Sınır Şartlar Uygulandıktan Sonra):

$$\begin{bmatrix} 7000000 & -7000000 & 0 \\ -7000000 & 9187500 & -2187500 \\ 0 & 2187500 & 2187500 \end{bmatrix} \begin{matrix} u_1 \\ u_2 \\ u_3 \end{matrix} = \begin{matrix} 0 \\ 2000 \\ 0 \end{matrix}$$

$9187500u_2 = 2000$
 $u_2 = 2,1769 \cdot 10^{-4} \text{ m}$

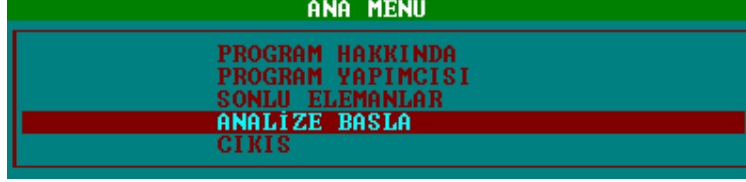
Element Stresses:

$$\sigma_1 = E \cdot \varepsilon_1 = E \cdot B_1 \cdot u_1 = E \cdot \frac{-1}{L_1} \cdot \frac{1}{L_1} \cdot (u_1 - u_2) = E \cdot \frac{u_2 - u_1}{L_1} = \frac{70 \cdot 10^9 \cdot (2,1769 \cdot 10^{-4} - 0)}{1} = 1,5238 \cdot 10^7 \text{ N/m}^2$$

$$\sigma_2 = E \cdot \varepsilon_2 = E \cdot B_2 \cdot u_2 = E \cdot \frac{-1}{L_2} \cdot \frac{1}{L_2} \cdot (u_3 - u_2) = E \cdot \frac{u_3 - u_2}{L_2} = \frac{70 \cdot 10^9 \cdot (0 - 2,1769 \cdot 10^{-4})}{0,8} = -1,9048 \cdot 10^7 \text{ N/m}^2$$

KİRİŞ ELEMAN İÇİN ÖRNEK PROBLEM PROGRAM YARDIMI İLE ÇÖZÜMÜ

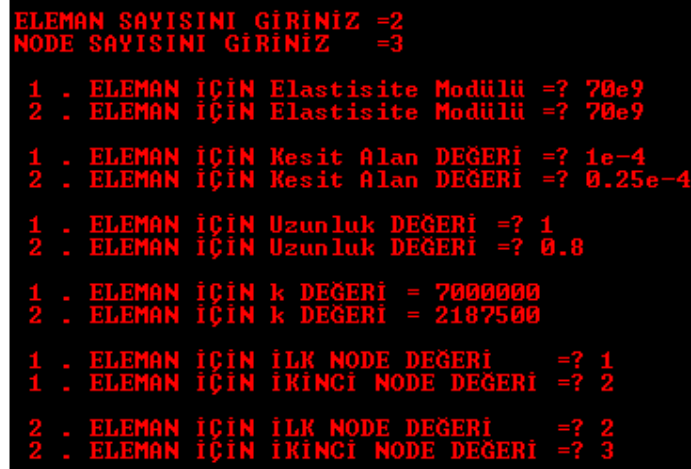
Program çalıştırıldıktan sonra ilk gelen ekranda “ENTER” tuşuna basılarak ekran Ana Menü’nün gelmesi sağlanır.



Eleman tipi seçimi için klavyedeki yön tuşlarını kullanarak “Analyze Başla” seçeneğine gelip “ENTER” tuşuna basın. Ekrana gelen Eleman Tipi Menü’sünden “Kiriş Eleman” seçilir



Bir önceki bölümde el ile çözülen örnek problemin Analizci programı ile çözümü için şu şekilde bir yol izlenir. Soruda verilenler uygun şekilde program tarafından sırası ile sorulur ve kullanıcı tarafından doğru verilerin girişi beklenir. Örnek problemimiz için programa aşağıdaki gibi verileri giriniz.



Bu işlemten sonra sıra yerdeğiştirmelerin girişi verilmelidir.

1. NODE İÇİN YER DEĞİŞTİRME: YOK
2. NODE İÇİN YER DEĞİŞTİRME: VAR
3. NODE İÇİN YER DEĞİŞTİRME: YOK



Yer<değiřtirmeler hakkında programa dođru bilgiler verildikten sonra sırada Kuvvet Deđerleri hakkında veri giriři yapılır. Problemdede sadece üçüncü node'da kuvvet uygulaması vardır. Bu node dışındaki tüm kuvvet deđerleri için "YOK" seçeneđi seçilmelidir. İkinci node'da "VAR" seçeneđi seçilerek program tarafından bu node'daki kuvvet deđerleri için 2000N girilir.

1. NODE İÇİN KUVVET DEĐERİ: YOK
2. NODE İÇİN KUVVET DEĐERİ: VAR » 2000 N
3. NODE İÇİN KUVVET DEĐERİ: YOK



Son olarak bu veriler girildikten sonra program arka planda oluşturduđu her bir eleman için stifness matrisleri toplayarak Global Stifness Matris'i hesaplar ve Nodal Displacement'leri de uygulayarak; bu matrisin tersi alınarak kuvvet matrisi ile çarpılır ve bilinmeyen yer deđiřtirmeler bulunur.

```

Connectivity Table
-----
ELEMAN - NODE i <1> - NODE j<2>
-----
1      1      2
2      2      3

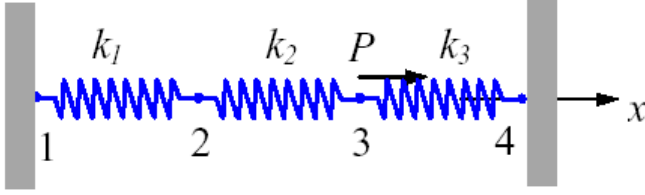
Global Stifness Matris
-----
7000000  -7000000  0
-7000000  9187500  -2187500
0         -2187500  2187500

Global Stifness Matris
Nodal Displacement Uygulandıktan Sonra
-----
9187500

Yer Deđiřtirme Deđerleri
-----
2.176871E-04

```

YAY ELEMAN İÇİN SORU :



Verilen:

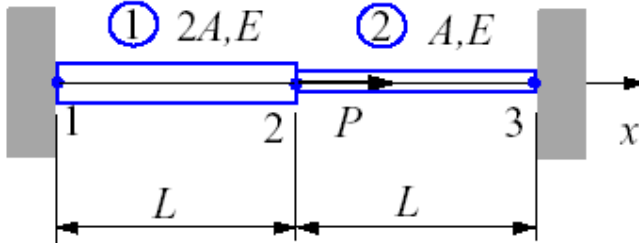
$$k_1 = 100 \text{ N/mm} \quad k_2 = 200 \text{ N/mm} \quad k_3 = 100 \text{ N/mm}$$
$$P = 5,00 \text{ N} \quad u_1 = u_4 = 0$$

İstenilen:

- 1- Connectivity table
- 2- Sınır Koşulları (Boundary conditions)
- 3- Eleman Direngenlik matrisleri (Elementary stiffness matrix) Üzerlerinde değişkenleri belirterek
- 4- Sistemin global direngenlik matrisi (Global stiffness matrix) Üzerlerinde değişkenleri belirterek
- 5- Sınır koşullarını uygulayınız
- 6- Bilinmeyen yerdeğiştirme terimlerini bulunuz (nodal displacements)

Yukarıda verilen bilgileri kullanarak Analizci programı ile çözümü yapınız

KİRİŞ ELEMAN İÇİN SORU :



Verilen:

$$P = 6.0 \times 10^4 \text{ N}, \quad E = 2.0 \times 10^4 \text{ N/mm}^2, \quad A = 250 \text{ mm}^2, \quad L = 150 \text{ mm}$$

İstenilen:

- 1- Connectivity table
- 2- Sınır Koşulları (Boundary conditions)
- 3- Eleman Direngenlik matrisleri (Elementary stiffness matrix) Üzerlerinde değişkenleri belirterek
- 4- Sistemin global direngenlik matrisi (Global stiffness matrix) Üzerlerinde değişkenleri belirterek
- 5- Sınır koşullarını uygulayınız
- 6- Bilinmeyen yerdeğiştirme terimlerini bulunuz (nodal displacements)

Yukarıda verilen bilgileri kullanarak Analizci programı ile çözümü yapınız