

UNIVERSIDADE DO PORTO  
FACULDADE DE ENGENHARIA

2.19

DESENVOLVIMENTO DE UM PROGRAMA DE PRE E PÓS-PROCESSAMENTO PARA  
VALIDAÇÃO DE DADOS E VISUALIZAÇÃO DE RESULTADOS DE ANÁLISES DE  
PROBLEMAS DE TENSÃO PELO MÉTODO DOS ELEMENTOS FINITOS

MANUEL JORGE LEÃO CABRAL



Universidade do Porto  
Faculdade de Engenharia  
Biblioteca 4

Nº  
CDU 629(093.0)/LCC1992/CABm  
Data 02/10/2009

PARECER

O presente parecer diz respeito ao estágio PRODEP nº 2.19 - FEUP/4.3/7/5/92/93 realizado por MANUEL JORGE LEÃO CABRAL no IC-Instituto da Construção.

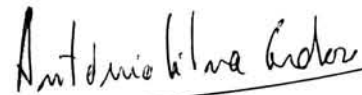
O estagiário demonstrou dedicação ao trabalho e, ao longo do tempo, foi revelando a competência necessária ao seu desenvolvimento.

Como consequência do estágio realizado, o IC aumentou as suas capacidades no domínio da aplicação das técnicas de computação gráfica, passando a dispôr de um programa de pré e pós processamento para validação dos dados e interpretação dos resultados de problemas planos de tensão e de deformação, analisados pelo Método dos Elementos Finitos.

Assim, conclui-se que o desempenho técnico do estagiário foi positivo e de interesse para o IC.

Porto e Faculdade de Engenharia. 29 de Novembro de 1993

O ORIENTADOR PELO IC



António Silva Cardoso  
(Professor Auxiliar)



UNIVERSIDADE DO PORTO  
FACULDADE DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA CIVIL  
GEOTECNIA

## PARECER

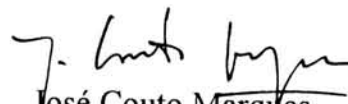
O presente Parecer diz respeito ao ESTÁGIO PRODEP nº 2.19 - FEUP/4.3/7/5/92/93 realizado por MANUEL JORGE LEÃO CABRAL no IC-Instituto da Construção.

O estágio efectuado teve como objectivo a aplicação de técnicas de computação gráfica no desenvolvimento de um programa de pré e pós-processamento para a validação de dados e a interpretação de resultados referentes a análises de problemas de tensão pelo Método dos Elementos Finitos.

O estagiário cumpriu plenamente os objectivos propostos, tendo o estágio atingido bom nível científico.

Porto e Faculdade de Engenharia, 26 de Novembro de 1993

O ORIENTADOR DO ESTÁGIO POR PARTE DA FEUP

  
José Couto Marques  
(Professor Auxiliar)

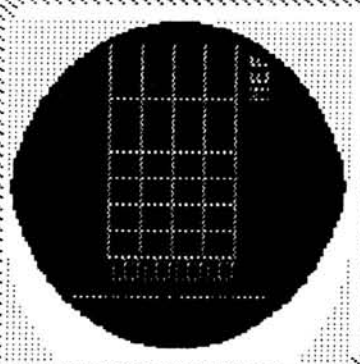
DESENVOLVIMENTO DE UM PROGRAMA DE PRÉ E PÓS-PROCESSAMENTO PARA  
VALIDAÇÃO DE DADOS E VISUALIZAÇÃO DE RESULTADOS DE ANÁLISES DE  
PROBLEMAS DE TENSÃO PELO MÉTODO DOS ELEMENTOS FINITOS

Manuel Jorge Leão Cabral

O Inventado

J. Leão Cabral

1900  
1901  
1902



# Draper Lane

1900  
1901  
1902

1900 1901 1902

Mamuel Jorge Leã Cabral

F  
E  
U  
P



**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

## 1 - INTRODUÇÃO

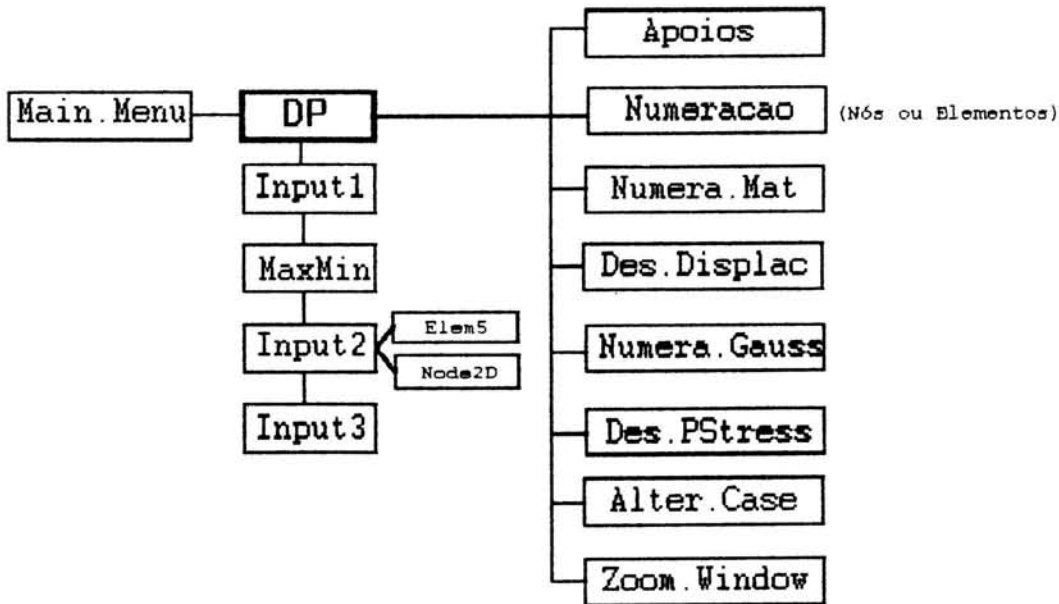
Com o crescente uso de programas de cálculo mais sofisticados e consequente incremento de dados a fornecer e informação a extrair desses programas, torna-se cada vez mais necessário a utilização de programas de pré e pós-processamento, com o intuito de controlar com maior eficácia quer a informação introduzida nos programas de cálculo, quer o toda a informação resultante do cálculo.

Foi com esse intuito que se desenvolveu o programa "DP" de pré e pós-processamento de dados e resultados. Este programa foi realizado para apoio ao programa "PLANE", de resolução de problemas planos elásticos pelo método dos elementos finitos, o qual foi desenvolvido pelo Dr. COUTO MARQUES (Prof. Auxiliar da F.E.U.P.), embora seja facilmente aplicado a outros programas do género conforme se explicará mais à frente neste relatório.



**2 - FLUXOGRAMA E DESCRIÇÃO DAS DIVERSAS ROTINAS QUE CONSTITUEM O PROGRAMA "DP"**

**2.1 - FLUXOGRAMA**



**2.2 - DESCRIÇÃO SUMÁRIA DAS DIVERSAS ROTINAS E SUBROTINAS**

**2.2.1 - DP :**

Programa principal, controla todo o fluxo de informação, abre o canal de leitura dos dados, dimensiona e redimensiona as diversas variáveis, verifica a chamada do programa, verifica se o nome do ficheiro introduzido existe em disco, imprime o écran (PrtSc), apaga a zona de desenho.

**2.2.2 - Main.Menu:**

Retorna à rotina principal a opção escolhida do menu.

**2.2.3 - Input1, Input2, Input3 :**

Leem os dados referentes à topologia da malha, deslocamentos, coordenadas dos pontos de Gauss, tensões principais.





**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

**2.2.3.1 - Elem5 :**

Ordena os nós dos elementos infinitos.

**2.2.3.2 - Node2D (1) :**

Gera as coordenadas de nós situados a meio dos lados rectilíneos de elementos planos de 8 e 9 nós.

Gera o nó central de elementos de 9 nós.

**2.2.4 - MaxMin :**

Determina o número máximo de nós por elemento.

Determina o número máximo de pontos de Gauss utilizados.

**2.2.5 - Apoios:**

Desenha os apoios nos respectivos nós.

**2.2.6 - Numeracao:**

Identifica os nós ou os elementos consoante a opção efectuada.

**2.2.7 - Numera.Mat:**

Identifica o tipo de material de que são constituídos os diversos elementos.

**2.2.8 - Des.Displac:**

Desenha a deformada da malha para cada caso de carga.

**2.2.9 - Numera.Gauss:**

Identifica os pontos onde são calculadas as tensões (pontos de Gauss).

**2.2.10 - Des.PStress:**

Desenha as tensões principais para cada caso de carga.

= → Compressão

- → Tracção

**2.2.11 - Alter.Case:**

Altera o caso de carga.

(1) Subrotina desenvolvida pelo Dr. Couto Marques (Prof. Auxiliar da F.E.U.P.).



**DrawPlane**  
MÉTODO DOS ELEMENTOS FINITOS  
MJLC/1993

**2.2.12 - Zoom.Window:**

Permite que se faça um zoom de parte da malha.

A zona que se pretende aumentar tanto pode ser definida através do rato, como através do teclado, sendo necessário definir o canto superior esquerdo da janela a abrir e o canto inferior direito da mesma .

**2.3 - Outras subrotinas auxiliares**

**2.3.1 - Scale:**

Determina as escalas gráficas dos nós, deslocamentos, tensões e pontos de Gauss.

**2.3.2 - Des.Mesh:**

Desenha a malha de elementos.

Esta subrotina é chamada logo após a determinação das escalas e sempre que se faça uma limpeza do écran, estando a malha de elementos sempre presente no écran.

**2.3.3 - EditeLine:**

Lê o novo ficheiro a editar.

**2.3.4 - InKeyCode, KeyCode:**

Subrotinas de controle do teclado.

**2.3.5 - DrawChar, DrawCharA, DrawCharS, LoadFont, Exist, SetVideo:**

Subrotinas fornecidas no Package "QuicPak".

**2.3.6 - Mouse, MouseRange, MouseHide, MouseInstall, MousePressLeft, MousePressRight, MouseReleaseLeft, MouseReleaseRight, MouseNow, MouseShow:**

Subrotinas de controle do rato.

Estas subrotinas encontram-se na biblioteca que acompanha o programa.



**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

**3 - ESTRUTURA DO FICHEIRO DE DADOS PARA O PROGRAMA "DP"**

O ficheiro de dados para o programa "DP" é um ficheiro não formatado, podendo ser efectuado através de um simples processador de texto ou directamente do programa de cálculo - opção que se recomenda.

Tem a seguinte estrutura:

**TITULO DO PROBLEMA** [1 Linha]  
**NPOIN NELEM NVFIX NMATS NODEQ NODET NODEI NTYPE NGAUQ NGAUT NCASE** [1 Linha]

NPOIN - NÚMERO DE NÓS  
 NELEM - NÚMERO DE ELEMENTOS  
 NVFIX - NÚMERO DE APOIOS  
 NMATS - NÚMERO DE DIFERENTES TIPOS DE MATERIAIS  
 NODEQ - NÚMERO DE NÓS DE ELEMENTOS QUADRANGULARES (4, 8 ou 9)  
 NODET - NÚMERO DE NÓS DE ELEMENTOS TRIÂNGULARES (3 ou 6)  
 NODEI - NÚMERO DE NÓS DE ELEMENTOS INFINITOS (5)  
 NTYPE - TIPO DE PROBLEMA (1 - ESTADO PLANO DE TENSÃO  
   2 - ESTADO PLANO DE DEFORMAÇÃO  
   3 - PROBLEMA AXISSIMÉTRICO)  
 NGAUQ - NÚMERO DE PONTOS DE GAUSS (4 ou 9)  
 NGAUT - NÚMERO DE PONTOS DE "RADAU" (3 ou 7)  
 NCASE - NÚMERO DE CASOS DE CARGA

**NÚMERO DO NÓ    COORD. X    COORD Y** [NPOIN Linhas]

**NOTA:**

- As coordenadas de nós situados a meio de lados rectilíneos em elementos de 8 ou 9 nós podem ser interpolados a partir dos respectivos nós de canto.

- As coordenadas do nó central de elementos de 9 nós podem ser interpoladas a partir das coordenadas dos restantes nós.

- Tem sempre que se terminar a lista das coordenadas nodais com o nó **NPOIN quer este seja nó de canto ou não.**

**NÚMERO DO NÓ    CÓDIGO DO APOIO** [NVFIX Linhas]

- Código - 01
- Código - 10
- Código - 11

**NÚMERO DO NÓ    DESLOC. X    DESLOC. Y** [NPOIN Linhas]

F  
E  
U  
P



**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

NÚMERO DO Pto. DE GAUSS    COORD. X    COORD. Y     $\sigma_1$      $\sigma_2$      $\theta$     [NGAUSS Linhas\*]

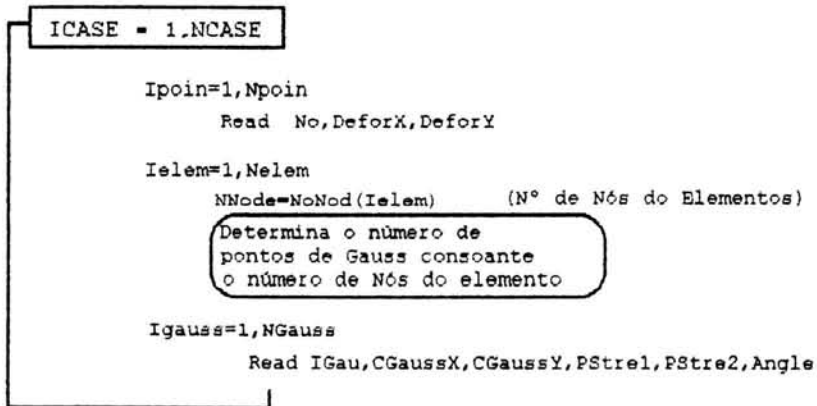
- $\sigma_1$  - Tensão principal máxima
- $\sigma_2$  - Tensão Principal mínima
- $\theta$  - Ângulo da Tensão Principal  $\sigma_1$  com a horizontal

**NOTA :**

- (\*) O número de pontos de Gauss utilizados, está dependente do tipo de elemento.
- Este bloco repete-se NELEM vezes.

**NOTA :**

- ESTES DOIS ÚLTIMOS BLOCOS REPETEM-SE NCASE VEZES CONFORME O FLUXOGRAMA QUE SE APRESENTA.





**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

**4 - LINGUAGEM UTILIZADA**

O programa "DP" foi escrito em "QuickBasic v4.5".

**5 - DIMENSÕES MÁXIMAS PERMITIDAS PARA OS ARRAYS**

MNode=9 - N° máximo de nós por elemento  
MElem=100 - N° máximo de elementos  
MPoin=200 - N° máximo de nós na malha  
MCase=5 - N° máximo de casos de carga  
MGauss=9 - N° máximo de pontos de Gauss por elemento  
MVfix=200 - N° máximo de apoios

LNods(MNode,MElem) - Ligações nodais  
Coord(2,MPoin) - Coordenadas dos nós  
NoNod(MElem) - Número de nós por elemento  
Matno(MElem) - Tipo de material de cada elemento  
IFpre(MVfix) - Código dos apoios  
Defor(2,MPoin,MCase) - Deslocamentos dos nós  
CGaus(2,MGauss,MCase) - Coordenadas dos pontos de Gauss  
PStr1(MGauss,MElem,MCase) - Tensão principal máxima  
PStr2(MGauss,MElem,MCase) - Tensão principal mínima  
Angle(MGauss,MElem,MCase) - Ângulo da tensão principal máxima com a horizontal

**6 - Requisitos para o programa "DP"**

Processador - 286 ou superior  
Écran - VGA (não está preparado para outro tipo de écran)  
- CÔR (preferencial)  
Mouse - Opcional

F  
E  
U  
P

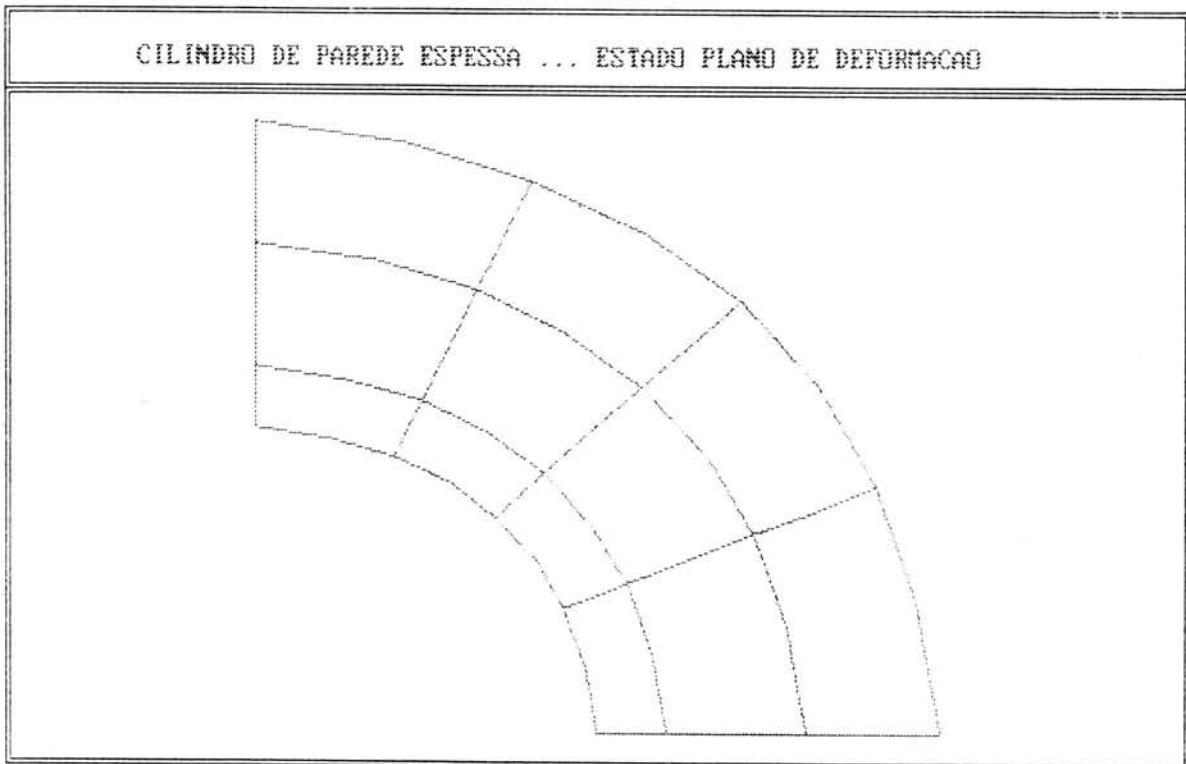


**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

## 7 - EXPLICAÇÃO SUMÁRIA DAS DIVERSAS OPÇÕES DO MENU

### 7.1 - CLS

Apaga o écran mantendo a malha dos elementos sempre visível



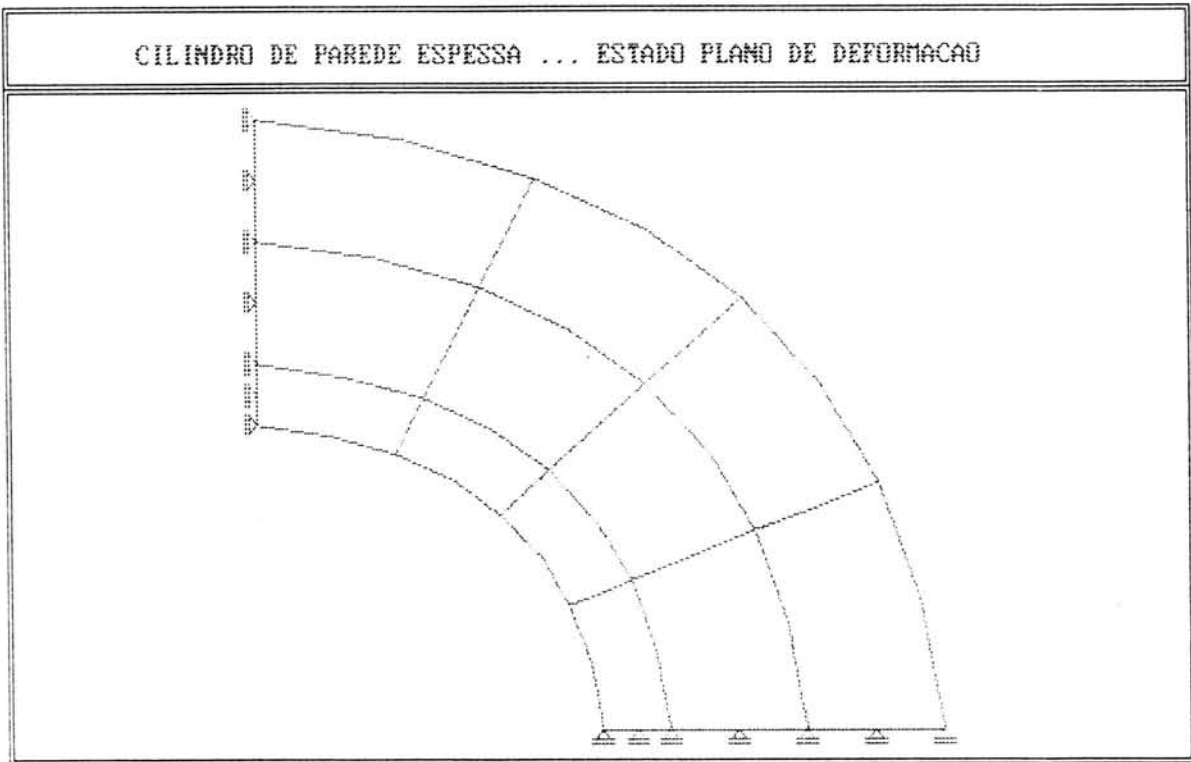
Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.



**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

**7.2 - APOIOS**

Desenha os apoios nos nós

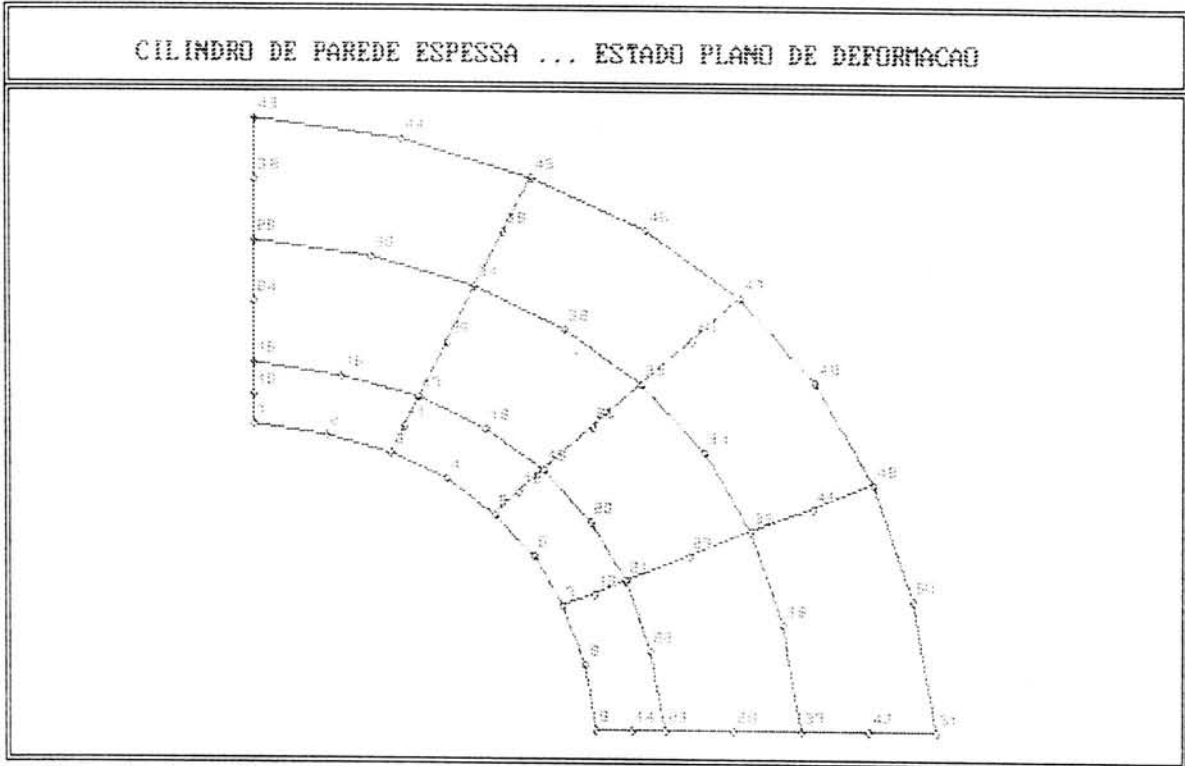


Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.



**7.3 - NÓS**

Identifica os nós da malha



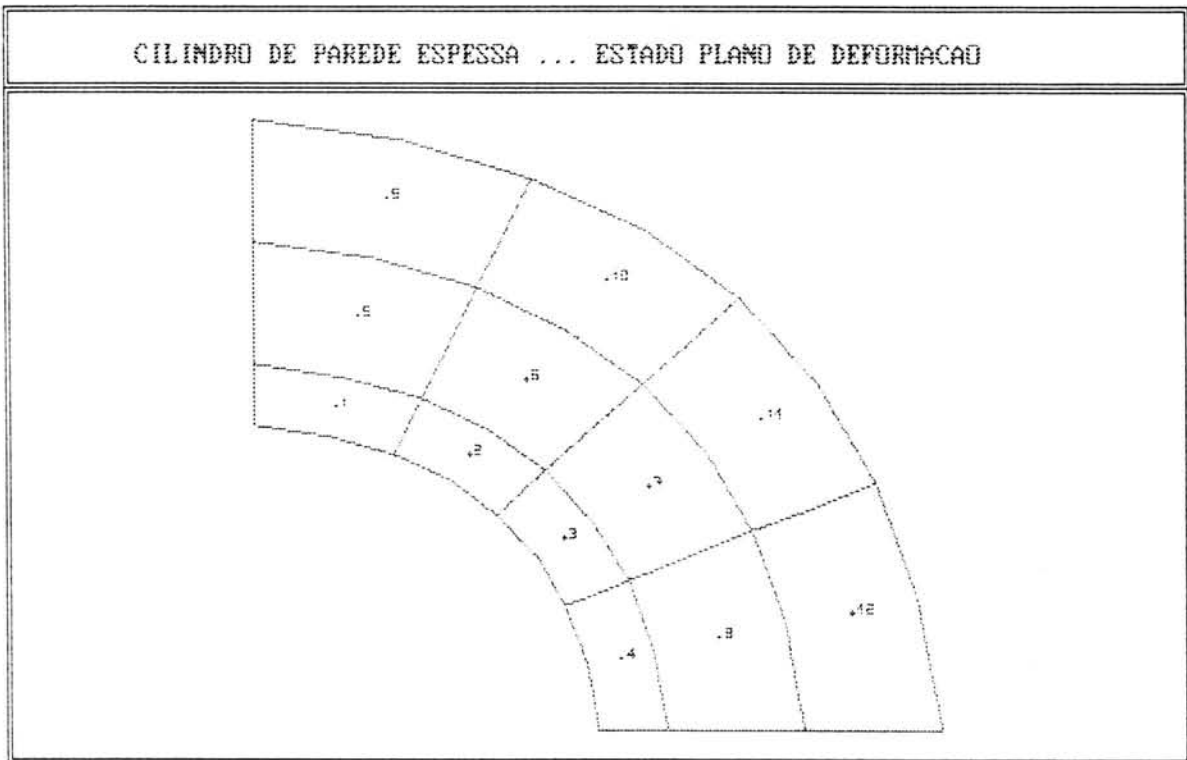
Cls Apoios Nós Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.





**7.4 - ELEMENTOS**

Identifica os elementos da malha

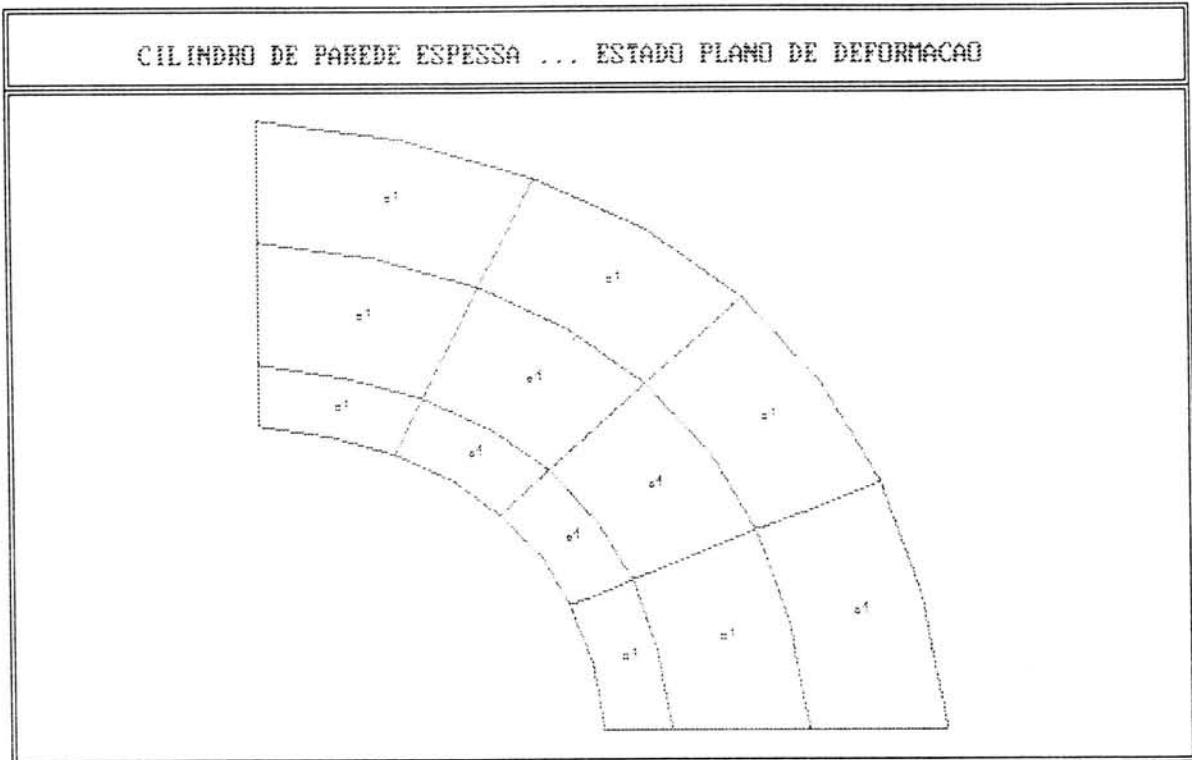


Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.

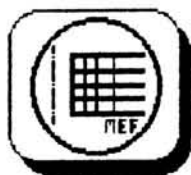


**7.5 - MATERIAIS**

Identifica o tipo de material de cada elemento

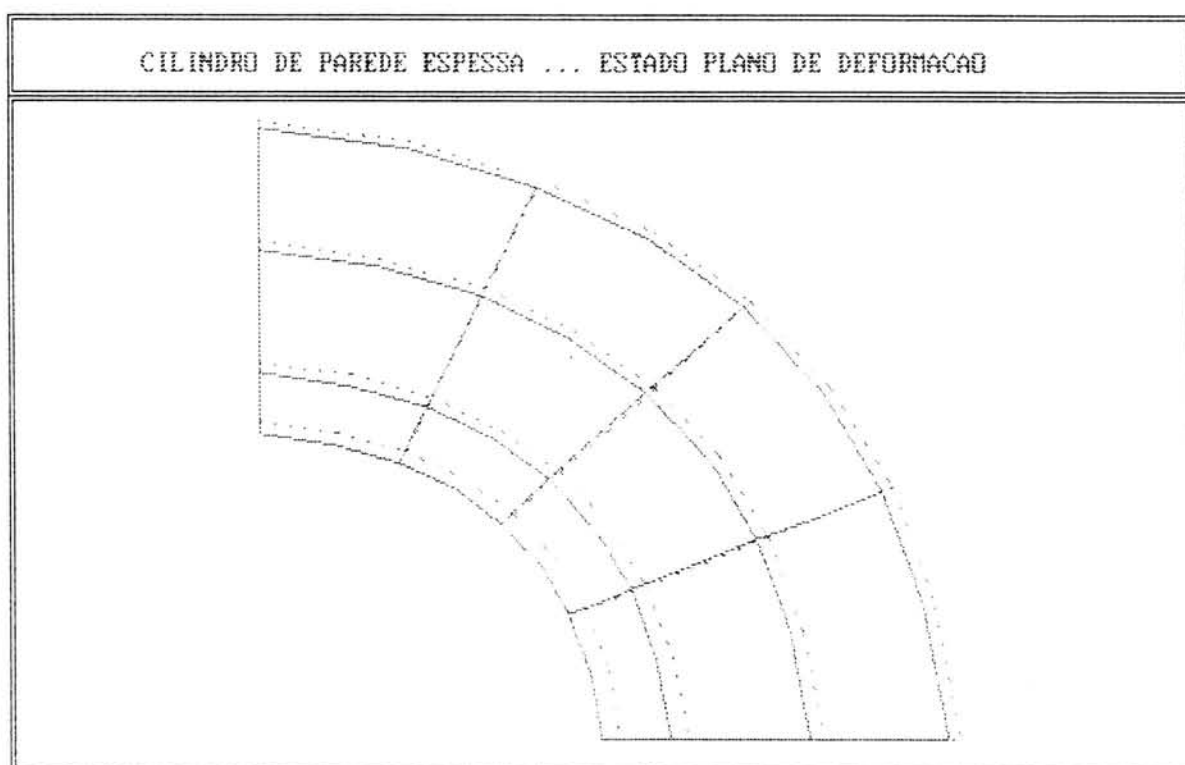


Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.



**7.6 - DEFORMADA**

Desenha a deformada da malha para cada caso de carga



Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.

F  
E  
U  
P

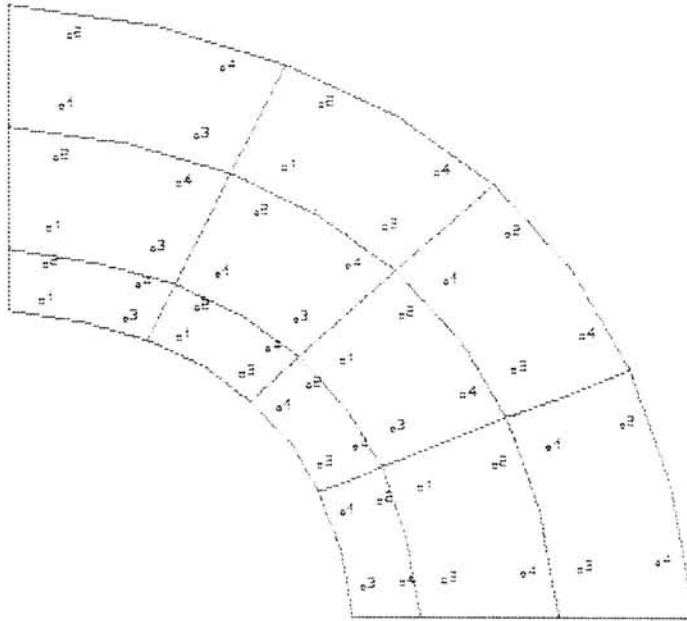


**DrawPlane**  
MÉTODO DOS ELEMENTOS FINITOS  
MJLC/1993

**7.7 - P\_GAUSS**

Identifica os pontos de Gauss - pontos onde são calculadas as tensões

CILINDRO DE PAREDE ESPESSA ... ESTADO PLANO DE DEFORMACAO

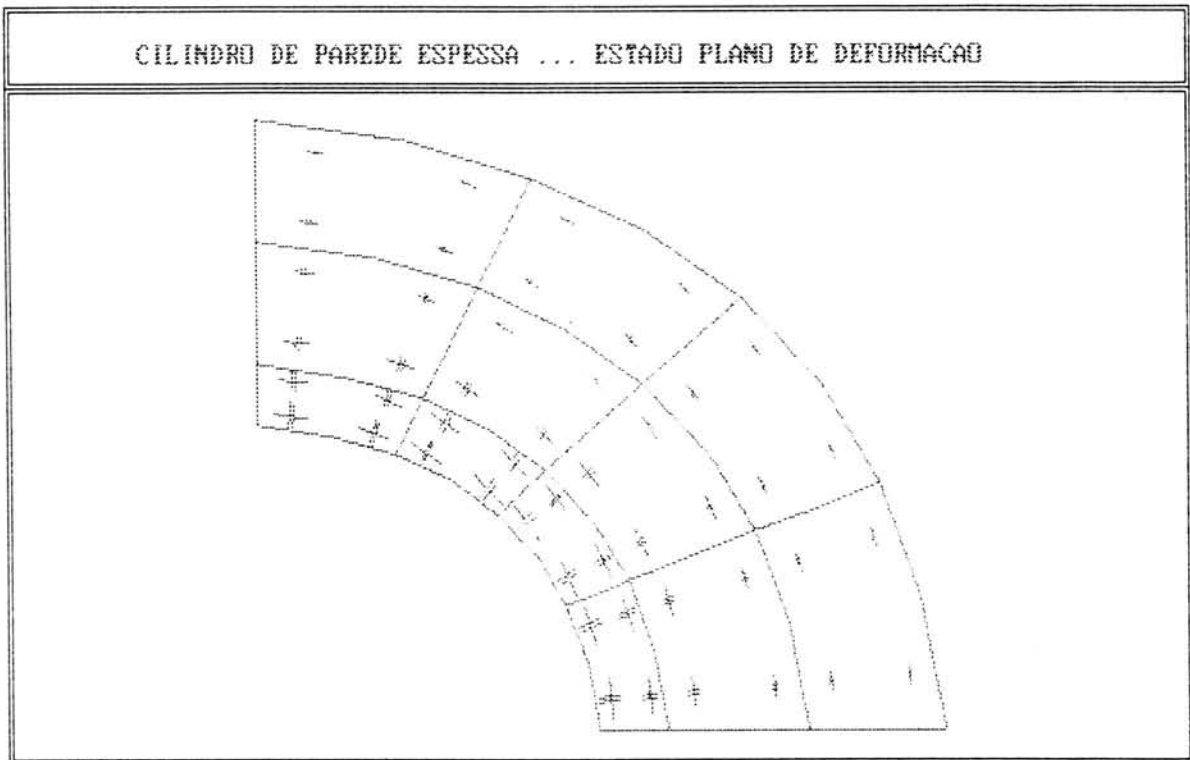


Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.



**7.8 -  $\sigma_1$ - $\sigma_2$**

Desenha as tensões principais nos pontos de Gauss para cada caso de carga



Cls Apoios Nos Elementos Materiais Deformada P\_Gauss  $\sigma_1$ - $\sigma_2$  N\_Solic.

**7.9 - N\_SOLIC.**

Muda para outro caso de carga.

**7.10 - N\_PROB.**

Faculta ao utilizador a opção de escolher outro ficheiro de dados que se encontra no disco, sem que se saia do programa.

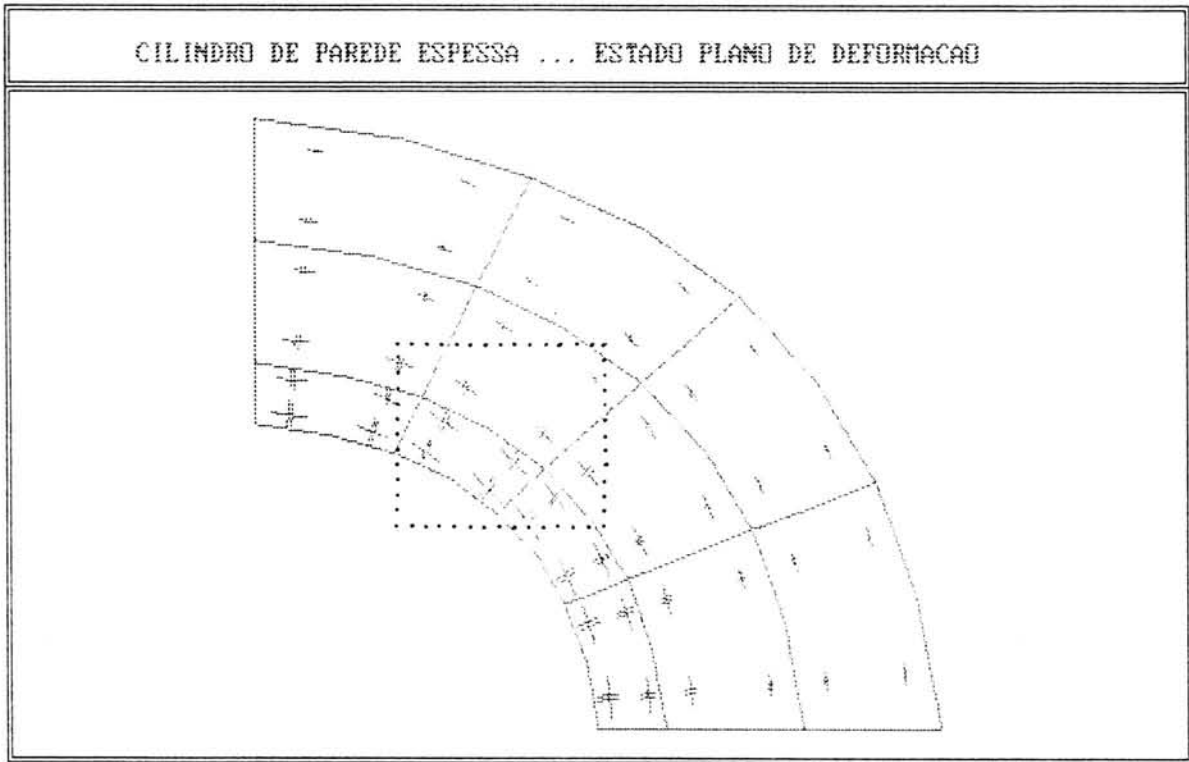
F  
E  
C  
U  
P



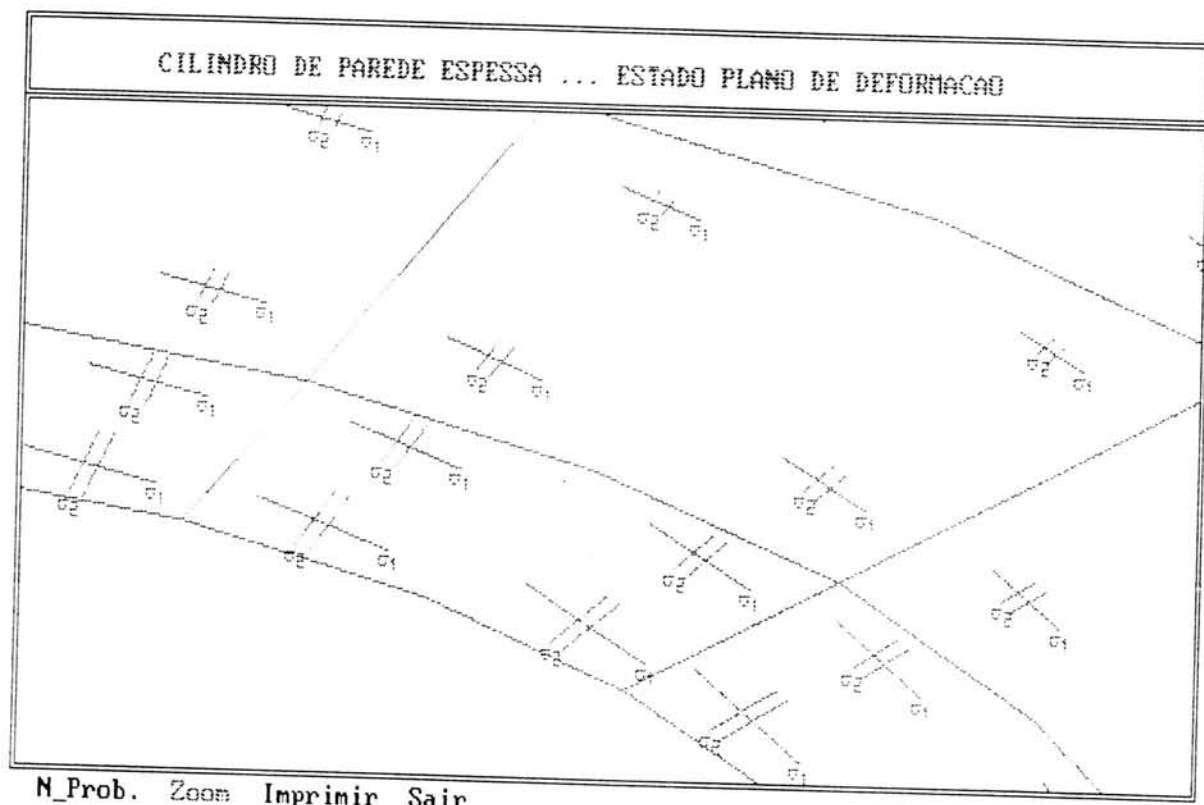
**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

### 7.11 - ZOOM

Permite ao utilizador definir uma janela, da qual se fará um zoom



N\_Prob. Zoom Imprimir Sair



Zoom da janela

### 7.12 - IMPRIMIR

Permite imprimir o que se encontra no écran. A impressão dos écrans só foi testada numa impressora EPSON e numa que poderia ser emulada como tal. Todos os écrans aqui apresentados foram imprimidos através desta opção.

### 7.13 - SAIR

Permite a saída do programa.

F  
E  
U  
P



**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

# **ANEXO I**

**LISTAGENS**



```

$DYNAMIC
C*****
C
C*** PROGRAMA "DP" DE APOIO AO PROGRAMA "PLANE"
C
C*** VISUALIZACAO GRAFICA DA MALHA, DEFORMADA, NOS, ELEMENTOS, MATERIAIS
C*** APOIOS, PONTOS DE GAUSS, TENSOES PRINCIPAIS
C
C**** 1993 - MJLC *****
C
COMMON SHARED Xmin, Xmax, Ymin, Ymax, ScaleX, ScaleY, X0, Y0

*** DECLARACAO DAS SUBROTINAS

DECLARE SUB Main_Menu (Item$( ), Choice%)
DECLARE SUB Apoios (NVFix%, NoFix%( ), Coord( ), IFPRE$( ))
DECLARE SUB Alter_Case (iCase%, NCase%)
DECLARE SUB Elem5 (AUX( ), jElem%, LNods%( ))
DECLARE SUB MaxMin (MNode%, MGauss%, NodeQ%, NodeT%, NodeI%, NGauQ%, NGauT%)
DECLARE SUB Scale (NGauQ%, NGauT%, NCase%, NPoin%, Nelem%, LNods%( ), NoNod%( ), Coord( ), Deform( ), CGauss( ), PStr1( ), PStr2( ), Angle( ))
DECLARE SUB Des_Displac (iCase%, Nelem%, NoNod%( ), LNods%( ), Deform( ))
DECLARE SUB Des_Mesh (Nelem%, NoNod%( ), Coord( ), LNods%( ))
DECLARE SUB Des_PStress (Zoom%, NGauQ%, NGauT%, Nelem%, iCase%, NoNod%( ), PStr1( ), PStr2( ), CGauss( ), Angle( ))
DECLARE SUB ZoomWindow (Zoom%, CrossPointer%( ), Nelem%, NoNod%( ), Coord( ), LNods%( ))
DECLARE SUB Input1 (TEXT0$, NPoin%, Nelem%, NVFix%, NMats%, NodeQ%, NodeT%, NodeI%, MNode%, NType%, NGauQ%, NGauT%, MGauss%, NCase%)
DECLARE SUB Input2 (NVFix%, NPoin%, Nelem%, Nodes%( ), LNods%( ), Coord( ), NoNod%( ), Matno%( ), NoFix%( ), IFPRE$( ))
DECLARE SUB Input3 (NCase%, Nelem%, NPoin%, NGauQ%, NGauT%, NoNod%( ), Deform( ), CGauss( ), PStr1( ), PStr2( ), Angle( ))
DECLARE SUB Node2D (Nodes%( ), Coord( ), LNods%( ), NPoin%, Nelem%, NoNod%( ))
DECLARE SUB Numeracao (kNo%, kElem%, Nelem%, NPoin%, NoNod%( ), LNods%( ), Coord( ))
DECLARE SUB Numeracao_Gauss (Nelem%, NGauQ%, NGauT%, NoNod%( ), CGauss( ))
DECLARE SUB Numeracao_Mat (Nelem%, Coord( ), LNods%( ), Matno%( ), NoNod%( ))
DECLARE SUB StartDrawPlane ( )
DECLARE FUNCTION InKeyCode% ( )
DECLARE FUNCTION KeyCode% ( )

Subrotinas de apoio
DECLARE SUB EditLine (a$, exitCode%)
DECLARE SUB Mouse (m1%, m2%, m3%, m4%)
DECLARE SUB MouseRange (X1%, Y1%, X2%, Y2%)
DECLARE SUB MouseHide ( )
DECLARE SUB MouseInstall (mflag%)
DECLARE SUB MousePressLeft (leftCount%, xMouse%, yMouse%)
DECLARE SUB MousePressRight (rightCount%, xMouse%, yMouse%)
DECLARE SUB MouseReleaseLeft (leftCount%, xMouse%, yMouse%)
DECLARE SUB MouseReleaseRight (rightCount%, xMouse%, yMouse%)
DECLARE SUB MouseNow (leftButton%, rightButton%, xMouse%, yMouse%)
DECLARE SUB MouseShow ( )
DECLARE SUB DrawText (Xx%, Yy%, Text$, Angle%, Colr%, TextSize#, Spacing%)
DECLARE SUB DrawChar (X%, Y%, Letter%, Colr%, Size#, Expand%)
DECLARE SUB DrawCharA (X%, Y%, SAngle#, Letter%, Colr%, Size#, Expand%)
DECLARE SUB DrawCharS (X%, Y%, Letter%, Colr%, Size#, Expand%, Direct%)
DECLARE SUB StepText (Xx%, Yy%, Text$, Angle%, Colr%, TextSize#, Spacing%)
DECLARE SUB SetVideo ( )

```

```

*$INCLUDE: "DP.INC"

```

\*\*\* FUNCAO PARA VERIFICAR SE EXISTE A FILE NO DISCO

```
DEF FnExist (FileNames$)
  STATIC CodeExist%
  CALL Exist(FileNames$, CodeExist%)
  FnExist = CodeExist%
END DEF
```

\*\*\* DATA PARA A CRUZ

```
DIM CrossPointer%(15)
DataCrossPointer:
DATA 7,7,1010,1010,1010,1010,1010,1010,FEFF,FEFE,1010,1010
DATA 1010,1010,1010,1010
```

\*\*\* INTRODUCAO DO NOME DOS FICHEIROS DE DADOS

```
DatFile$ = COMMAND$
```

\*\*\* VERIFICA SE A CHAMADA DO PROGRAMA E CORRECTA

```
IF DatFile$ = "" THEN
  SCREEN 0, , 0, 0
  SHELL "CLS"
  PRINT "Modo de Uso : DP NOME_DO_FICHEIRO"
  END
END IF
```

```
Ext% = INSTR(".", DatFile$)
IF Ext% <> 0 THEN
  DatFile$ = ""
ELSE
  DatFile$ = DatFile$ + ".DES"
END IF
IF NOT FnExist(DatFile$) THEN
  SHELL "CLS"
  PRINT "O Ficheiro "; DatFile$; " nao existe !!!"
  END
END IF
```

\*\*\* SUBROTINA DE ARRANQUE

```
CALL SetVideo: CLS
StartDrawPlane
SLEEP 2
```

\*\*\* DIMENSIONAMENTO DAS VARIAVEIS

```
MNode% = 9: Melem% = 100: MPoin% = 200: MCase% = 5: MGauss% = 9: MVFix% =
```

```
200
DIM LNods%(MNode%, Melem%), Coord(2, MPoin%), NoNod%(Melem%), Matno%(Melem%
```

```
%),
DIM NoFix%(MVFix%), IFPRE$(MVFix%), Defor(2, MPoin%, MCase%)
DIM CGauss(2, MGauss%, Melem%)
DIM PStr1(MGauss%, Melem%, MCase%), PStr2(MGauss%, Melem%, MCase%)
DIM Angle(MGauss%, Melem%, MCase%)
DIM SHARED AUX(5), Nodes%(9)
```

```
GOSUB Start
```

\*\*\* MENU PRINCIPAL

```
CLS : SCREEN 9, , 0, 0: COLOR , 0
LINE (0, 0)-(639, 36), 1, B: LINE (2, 2)-(637, 34), 1, B
```

```
LINE (0, 36)-(639, 335), 1, B: LINE (2, 38)-(637, 333), 1, B
VIEW SCREEN (3, 3)-(636, 14): CLS 1: VIEW
LENA% = LEN(TEXT0$)
COMPTXT% = INT(78 - LENA%)
COMPTXT% = COMPTXT% \ 2
LOCATE 2, COMPTXT%: COLOR 4: PRINT TEXT0$: COLOR 7
CALL Des.Mesh(Nelem%, NoNod%, Coord(), LNods%())
```

```
*** LEITURA DAS OPCOES DO MENU
```

```
DIM Choice$(13)
RESTORE DataMenu
FOR iMenu% = 1 TO 13
  READ Choice$(iMenu%)
NEXT iMenu%
```

```
DataMenu:
```

```
DATA "Cls ", "Apoios", "Nos", "Elementos", "Materiais", "Deformada"
DATA "P_Gauss", "σ1-σ2", "N_Solic.", "N_Prob.", "Zoom", "Imprimir", "Sair"
```

```
*** PROGRAMA PRINCIPAL
```

```
iCase% = 1
```

```
FirstMainLoop:
```

```
DO
```

```
VIEW PRINT 25 TO 25: CALL Main_Menu(Choice$, Choice%)
```

```
IF Choice% = 1 THEN
```

```
VIEW SCREEN (3, 39)-(636, 332): CLS 1
CALL Des.Mesh(Nelem%, NoNod%, Coord(), LNods%())
iDeform% = 0: kNo% = 0: kElem% = 0: iPStress% = 0: iNumGauss% = 0
iNode% = 0: iElem% = 0: iApoios% = 0: iCase% = 1: iMats% = 0
```

```
ELSEIF Choice% = 2 THEN
```

```
CALL Apoios(NVFix%, NoFix%, Coord(), IFPRE$())
iApoios% = 1
```

```
ELSEIF Choice% = 3 THEN
```

```
kNo% = 1: kElem% = 0
CALL Numeracao(kNo%, kElem%, Nelem%, NPoin%, NoNod%, LNods%, Coord())
iNode% = 1
```

```
ELSEIF Choice% = 4 THEN
```

```
IF iMats% = 1 THEN
  iMats% = 0
  VIEW SCREEN (3, 39)-(636, 332): CLS 1
  CALL Des.Mesh(Nelem%, NoNod%, Coord(), LNods%())
  GOSUB StateOfThings
END IF
kNo% = 0: kElem% = 1
CALL Numeracao(kNo%, kElem%, Nelem%, NPoin%, NoNod%, LNods%, Coord())
iElem% = 1
```

```
ELSEIF Choice% = 5 THEN
```

```
IF iElem% = 1 THEN
  iElem% = 0
  VIEW SCREEN (3, 39)-(636, 332): CLS 1
  CALL Des.Mesh(Nelem%, NoNod%, Coord(), LNods%())
  GOSUB StateOfThings
END IF
CALL Numeracao(Mat(Nelem%, Coord(), LNods%, MatNo%, NoNod%()))
iMats% = 1
```

```

ELSEIF Choice% = 6 THEN
    CALL Des.Displac(iCase%, Nelem%, NoNod%(), LNods%(), DeFor())
    iDeform% = 1

ELSEIF Choice% = 7 THEN
    CALL Numera.Gauss(Nelem%, NGauQ%, NGauT%, NoNod%(), CGauss())
    iNumGauss% = 1

ELSEIF Choice% = 8 THEN
    CALL Des.PStress(Zoom%, NGauQ%, NGauT%, Nelem%, iCase%, NoNod%(), PS
1(), PStr2(), CGauss(), Angle())
    iPStress% = 1

ELSEIF Choice% = 9 THEN
    Alter.Case iCase%, NCase%

ELSEIF Choice% = 10 THEN
    CLS 2: LOCATE 25, 1: COLOR 7
    PRINT "File Spec : "; : NewFile$ = SPACE$(8)
    EditLine NewFile$, exitCode%
    NewFile$ = LTRIM$(RTRIM$(NewFile$))
    DatFile$ = NewFile$ + ".DES"
    IF FnExist(DatFile$) THEN
        VIEW PRINT: LOCATE 2, 2: COLOR 0: PRINT STRING$(78, 219);
        VIEW SCREEN (3, 39)-(636, 332): CLS 1
        GOSUB Start
        LENA% = LEN(TEXT0$): COMPTEXT% = INT(78 - LENA%) \ 2
        LOCATE 2, COMPTEXT%: COLOR 4: PRINT TEXT0$: COLOR 7
        iDeform% = 0: kNo% = 0: kElem% = 0: iPStress% = 0: iNumGauss% = 0
        iNode% = 0: iElem% = 0: iApoios% = 0: iCase% = 1: iMats% = 0
        CALL Des.Mesh(Nelem%, NoNod%(), Coord(), LNods%())
    ELSE
        CLS 2
        LOCATE 25, 1: COLOR 7: PRINT "A file "; DatFile$; " nao existe";
        DO: LOOP UNTIL INKEY$ <> ""
    END IF

ELSEIF Choice% = 11 THEN
    CALL ZoomWindow(Zoom%, CrossPointer%(), Nelem%, NoNod%(), Coord(), LN
ds%())
    GOSUB StateOfThings

ELSEIF Choice% = 12 THEN
    CLS 2: Length% = 73 - LEN(DatFile$)
    LOCATE 25, Length%: PRINT USING "FILE : &"; DatFile$;
    IF iPStress% = 1 THEN
        LOCATE 25, 2: PRINT "=== Compressao : — Traccao";
    END IF
    FOR iCor% = 1 TO 15
        PALETTE iCor%, 7
    NEXT iCor%
    DO: LOOP UNTIL INKEY$ <> ""
    SHELL "Graphics"
    CALL PrtSc
    PALETTE
ELSEIF Choice% = 13 THEN
    - END
END IF
LOOP

```

StateOfThings:

```

IF iDeform% = 1 THEN
    CALL Des.Displac(iCase%, Nelem%, NoNod%(), LNods%(), DeFor())
END IF
IF iNode% = 1 THEN
    kNo% = 1: kElem% = 0

```

```

CALL Numeracao(kNo%, kElem%, Nelem%, NPoin%, NoNod%(), LNods%(), Coord())
END IF
IF iElem% = 1 THEN
  kNo% = 0: kElem% = 1
  CALL Numeracao(kNo%, kElem%, Nelem%, NPoin%, NoNod%(), LNods%(), Coord())
END IF
IF iNumGauss% = 1 THEN
  CALL Numera.Gauss(NoElem%, NGauQ%, NGauT%, NoNod%(), CGauss())
END IF
IF iApoios% = 1 THEN
  CALL Apoios(NVFix%, NoFix%(), Coord(), IFPRE$())
END IF
IF iPStress% = 1 THEN
  CALL Des.PStress(Zoom%, NGauQ%, NGauT%, Nelem%, iCase%, NoNod%(), PStr1(),
PStr2(), CGauss(), Angle())
END IF
RETURN

```

Start:

```

*** ABERTURA DOS CANAIS DE LEITURA DOS PARAMETROS DE CONTROLE
*** E RESTANTES DADOS

OPEN DatFile$ FOR INPUT AS #1

*** LEITURA DOS PARAMETROS DE CONTROLE

CALL Input1(TEXT0$, NPoin%, Nelem%, NVFix%, NMats%, NodeQ%, NodeI%, NodeL%,
MNode%, NType%, NGauQ%, NGauT%, MGauss%, NCase%)
CALL MaxMin(MNode%, MGauss%, NodeQ%, NodeI%, NodeL%, NGauQ%, NGauT%)

*** DIMENSIONAMENTO DAS VARIAVEIS

REDIM LNods%(MNode%, Nelem%), Coord(2, NPoin%), NoNod%(Nelem%), Matno%(Nelem%)
REDIM NoFix%(NVFix%), IFPRE$(NVFix%), Defor(2, NPoin%, NCase%)
REDIM CGauss(2, MGauss%, Nelem%)
REDIM PStr1(MGauss%, Nelem%, NCase%), PStr2(MGauss%, Nelem%, NCase%)
REDIM Angle(MGauss%, Nelem%, NCase%)
REDIM SHARED AUX(5), Nodes%(9)

*** LEITURA DAS LIGACOES NODAIS, COORDENADAS, MATERIAIS,
*** LIGACOES EXTERIORES, etc.

CALL Input2(NVFix%, NPoin%, Nelem%, Nodes%(), LNods%(), Coord(), NoNod%(),
Matno%(), NoFix%(), IFPRE$())

*** LEITURA DOS DESLOCAMENTOS, COORDENADAS DOS PONTOS
*** DE GAUSS E DAS TENSOES TENSOES PRINCIPAIS

CALL Input3(NCase%, Nelem%, NPoin%, NGauQ%, NGauT%, NoNod%(), Defor(), CGauss(),
PStr1(), PStr2(), Angle())
CLOSE

*** DETERMINACAO DAS ESCALAS
*** TRANSFORMACAO DAS COORDENADAS GEOMETRICAS EM COORDENADAS GRAFICAS

CALL Scale(NGauQ%, NGauT%, NCase%, NPoin%, Nelem%, LNods%(), NoNod%(), Coord(),
Defor(), CGauss(), PStr1(), PStr2(), Angle())

RETURN

```

```
UB Input1 (TEXTOS$, NPoint%, Nelem%, NVFix%, NMats%, NodeQ%, NodeT%, NodeI%, MNod
%, NType%, NGauQ%, NGauT%, MGauss%, NCase%)
```

```
*** LEITURA DOS PARAMETROS DE CONTROLE
```

```
INPUT #1, TEXTOS$
INPUT #1, NPoint%, Nelem%, NVFix%, NMats%, NodeQ%, NodeT%, NodeI%, NType%,
GauQ%, NGauT%, NCase%
ND SUB
```

```
UB Input2 (NVFix%, NPoint%, Nelem%, Nodes%(), LNods%(), Coord(), NoNod%(), Matno
%), NoFix%(), IFPRE$())
```

```
*** LEITURA DOS NOS DE Nos, MATERIAL, LIGACOES NODAIS
```

```
FOR iElem% = 1 TO Nelem%
  INPUT #1, jElem%, Matno%(jElem%), NoNod%(jElem%)
  FOR iNode% = 1 TO NoNod%(jElem%)
    INPUT #1, LNods%(iNode%, jElem%)
  NEXT iNode%
  IF NoNod%(jElem%) = 5 THEN
    CALL Elem5(AUX(), jElem%, LNods%())
  END IF
NEXT iElem%
KOUNT = 0: iPoint% = 0
DO WHILE iPoint% <> NPoint%
  INPUT #1, iPoint%, Coord(1, iPoint%), Coord(2, iPoint%)
  KOUNT = KOUNT + 1
LOOP
IF KOUNT <> NPoint% THEN
  CALL Node2D(Nodes%(), Coord(), LNods%(), NPoint%, Nelem%, NoNod%())
END IF
```

```
*** LEITURA DAS LIGACOES EXTERIORES
```

```
FOR IVFIX = 1 TO NVFix%
  INPUT #1, No%, IFPRE$(IVFIX)
  NoFix%(IVFIX) = No%
NEXT IVFIX
ND SUB
```

```
UB Input3 (NCase%, Nelem%, NPoint%, NGauQ%, NGauT%, NoNod%(), Defor(), CGauss(),
PStr1(), PStr2(), Angle())
```

```
*** CICLO SOBRE OS CASOS DE CARGA
```

```
FOR iCase% = 1 TO NCase%
```

```
*** LEITURA DOS DESLOCAMENTOS
```

```
FOR iPoint% = 1 TO NPoint%
  INPUT #1, No%, Defor(1, No%, iCase%), Defor(2, No%, iCase%)
NEXT iPoint%
```

```
*** LEITURA DAS COORDENADAS DOS PONTOS DE GAUSS,  $\sigma_1$ ,  $\sigma_2$ , ang
```

```
FOR iElem% = 1 TO Nelem%
```

```

      NNode% = NoNod%(iElem%)
      NGauss% = NGauQ%
      IF NNode% = 3 OR NNode% = 6 THEN NGauss% = NGauT%
      FOR iGauss% = 1 TO NGauss%
        INPUT #1, iGau%
        INPUT #1, CGauss(1, iGau%, iElem%)
        INPUT #1, CGauss(2, iGau%, iElem%)
        INPUT #1, PStr1(iGau%, iElem%, iCase%)
        INPUT #1, PStr2(iGau%, iElem%, iCase%)
        INPUT #1, Angle(iGau%, iElem%, iCase%)
      NEXT iGauss%
    NEXT iElem%
  NEXT iCase%
END SUB

```

```

SUBROUTINE IPAPUS(IPAPUS)
  IF IPAPUS = 1 THEN
    GO TO 1000
  ELSE IF IPAPUS = 2 THEN
    GO TO 2000
  ELSE IF IPAPUS = 3 THEN
    GO TO 3000
  ELSE IF IPAPUS = 4 THEN
    GO TO 4000
  END IF

```

```

1000 IPAPUS

```

```

2000 IPAPUS
RETURN

```

```

3000 IPAPUS

```

```

4000 IPAPUS

```

```

5000 IPAPUS

```

```

6000 IPAPUS

```

```

7000 IPAPUS

```

```

8000 IPAPUS

```

```

9000 IPAPUS

```

```

10000 IPAPUS

```

```

11000 IPAPUS

```

```

12000 IPAPUS

```

```

13000 IPAPUS

```

```

14000 IPAPUS

```

```

15000 IPAPUS

```

```

16000 IPAPUS

```

```

17000 IPAPUS

```

```

18000 IPAPUS

```

```

19000 IPAPUS

```

```

20000 IPAPUS

```

```

21000 IPAPUS

```

```

22000 IPAPUS

```

```

23000 IPAPUS

```

```

24000 IPAPUS

```

```

25000 IPAPUS

```

```

26000 IPAPUS

```

```

27000 IPAPUS

```

```

28000 IPAPUS

```

```

29000 IPAPUS

```

```

30000 IPAPUS

```

```

31000 IPAPUS

```

```

32000 IPAPUS

```

```

33000 IPAPUS

```

```

34000 IPAPUS

```

```

35000 IPAPUS

```

```

36000 IPAPUS

```

```

37000 IPAPUS

```

```

38000 IPAPUS

```

```

39000 IPAPUS

```

```

40000 IPAPUS

```

SUB Apoios (NVFix%, NoFix%(), Coord(), IFPRES\$())

ORDENA O DESENHO DAS LIGACOES

```
9000 FOR IVFIX = 1 TO NVFix%
    No% = NoFix%(IVFIX)
    XNo = Coord(1, No%); YNo = Coord(2, No%)
    PSET (XNo, YNo), 4
    GOSUB 9000
NEXT IVFIX
EXIT SUB
```

IDENTIFICA O TIPO DE APOIO

```
9000 IFAP$ = IFPRES$(IVFIX)
IF IFAP$ = "11" THEN
    GOTO 9140
ELSEIF IFAP$ = "01" THEN
    GOTO 9160
ELSEIF IFAP$ = "10" THEN
    GOTO 9150
END IF
```

APOIO DUPL0

APOIO SIMPLES VERTICAL

APOIO DUPL0

```
9140 DRAW "C4" + " NG4 F4 BR2 L12 G2 BR3 E2 BR3 G2 BR3 E2 BR3 G2"
RETURN
```

APOIO SIMPLES VERTICAL

```
9150 IF XNo <= (X0 + ScaleX * (Xmax + Xmin) / 2) THEN
    DRAW "C4" + " NH4 G4 BD1 U10 BL2 D10"
ELSE
    DRAW "C4" + " NE4 F4 BD1 U10 BR2 D10"
END IF
RETURN
```

APOIO SIMPLES HORIZONTAL

```
9160 IF YNo >= 350! - (Y0 + ScaleY * (Ymax + Ymin) / 2) THEN
    DRAW "C4" + " NG4 F4 BR2 L12 BD2 R12"
ELSE
    DRAW "C4" + " NH4 E4 BR2 L12 BU2 R12"
END IF
RETURN
END SUB
```



SUB Numeracao (kNo%, kElem%, Nelem%, NPoin%, NoNod%(), LNods%(), Coord())

NUMERACAO GRAFICA DOS NOS

```
IF kNo% = 1 THEN
  FOR iPoin% = 1 TO NPoin%
    X = Coord(1, iPoin%)
    Y = Coord(2, iPoin%)
    PSET (X, Y), 0: DRAW "C9 BR2D1GILIH1U1E1R1F18U4BR1" "C9 BD1 BR1 L2 U2
R2 D2 BU4 BR2"
    VNUM$ = MID$(STR$(iPoin%), 2, LEN(STR$(iPoin%)))
    COR$ = "C9"
    GOSUB 6500
  NEXT iPoin%
```

NUMERACAO GRAFICA DOS ELEMENTOS

```
ELSEIF kElem% = 1 THEN
  FOR iElem% = 1 TO Nelem%
    NNode% = NoNod%(iElem%)

    IF NNode% = 3 THEN
      XNode1% = LNods%(1, iElem%)
      XNode2% = LNods%(2, iElem%)
      XNode3% = LNods%(3, iElem%)
      X1 = Coord(1, XNode1%): Y1 = Coord(2, XNode1%)
      X2 = Coord(1, XNode2%): Y2 = Coord(2, XNode2%)
      X3 = Coord(1, XNode3%): Y3 = Coord(2, XNode3%)
      xM = (X1 + X2 + X3) / 3
      yM = (Y1 + Y2 + Y3) / 3
    ELSEIF NNode% = 6 THEN
      XNode1% = LNods%(1, iElem%)
      XNode3% = LNods%(3, iElem%)
      XNode6% = LNods%(6, iElem%)
      X1 = Coord(1, XNode1%): Y1 = Coord(2, XNode1%)
      X3 = Coord(1, XNode3%): Y3 = Coord(2, XNode3%)
      X6 = Coord(1, XNode6%): Y6 = Coord(2, XNode6%)
      xM = (X1 + X3 + X6) / 3
      yM = (Y1 + Y3 + Y6) / 3
    ELSE
      IF NNode% = 5 THEN
        XNode2% = LNods%(2, iElem%)
        XNode5% = LNods%(5, iElem%)
        X1 = Coord(1, XNode2%)
        Y1 = Coord(2, XNode2%)
        X2 = Coord(1, XNode5%)
        Y2 = Coord(2, XNode5%)
      ELSE
        XNode2% = LNods%(2, iElem%)
        XNode6% = LNods%(6, iElem%)
        X1 = Coord(1, XNode2%)
        Y1 = Coord(2, XNode2%)
        X2 = Coord(1, XNode6%)
        Y2 = Coord(2, XNode6%)
      END IF
      xM = (X1 + X2) * .5: yM = (Y1 + Y2) * .5
    END IF
    PSET (xM, yM), 0: DRAW "C10 NLINUINDIR1BR2"
    VNUM$ = MID$(STR$(iElem%), 2, LEN(STR$(iElem%)))
    COR$ = "C10"
    GOSUB 6500
  NEXT iElem%
```

```
END IF
EXIT SUB
```

```

5500 FOR J% = 1 TO LEN(VNUM$)
    NUM$ = MID$(VNUM$, J%, 1)
    IF NUM$ = "1" THEN
        DRAW COR$ + "BR2 U4 G2 BD2 BR5"
    ELSEIF NUM$ = "2" THEN
        DRAW COR$ + "NR3 U2 R3 U2 L3 BD4 BR5"
    ELSEIF NUM$ = "3" THEN
        DRAW COR$ + "R3 U2 NL2 U2 L3 BD4 BR6"
    ELSEIF NUM$ = "4" THEN
        DRAW COR$ + "BR3 U4 G3 R4 BD1 BR2"
    ELSEIF NUM$ = "5" THEN
        DRAW COR$ + "R3 U2 L3 U2 R3 BD4 BR2"
    ELSEIF NUM$ = "6" THEN
        DRAW COR$ + "NR3 U4 R3 BD2 NL3 D2 BR3"
    ELSEIF NUM$ = "7" THEN
        DRAW COR$ + "BU3 U1 R3 D2 NL1 NR1 D2 BR3"
    ELSEIF NUM$ = "8" THEN
        DRAW COR$ + "BR1 NR3 U4 R3 D2 NL3 D2 BR2"
    ELSEIF NUM$ = "9" THEN
        DRAW COR$ + "R3 U4 L3 D2 R3 D2 BR3"
    ELSEIF NUM$ = "0" THEN
        DRAW COR$ + "U4 R3 D4 NL3 BR2"
    ELSEIF NUM$ = "-" THEN
        DRAW COR$ + "BU2 R3 BD2"
    END IF
NEXT J%
RETURN
END SUB

```

```

5500 FOR J% = 1 TO LEN(VNUM$)
    NUM$ = MID$(VNUM$, J%, 1)
    IF NUM$ = "1" THEN
        DRAW COR$ + "BR2 U4 G2 BD2 BR5"
    ELSEIF NUM$ = "2" THEN
        DRAW COR$ + "NR3 U2 R3 U2 L3 BD4 BR5"
    ELSEIF NUM$ = "3" THEN
        DRAW COR$ + "R3 U2 NL2 U2 L3 BD4 BR6"
    ELSEIF NUM$ = "4" THEN
        DRAW COR$ + "BR3 U4 G3 R4 BD1 BR2"
    ELSEIF NUM$ = "5" THEN
        DRAW COR$ + "R3 U2 L3 U2 R3 BD4 BR2"
    ELSEIF NUM$ = "6" THEN
        DRAW COR$ + "NR3 U4 R3 BD2 NL3 D2 BR3"
    ELSEIF NUM$ = "7" THEN
        DRAW COR$ + "BU3 U1 R3 D2 NL1 NR1 D2 BR3"
    ELSEIF NUM$ = "8" THEN
        DRAW COR$ + "BR1 NR3 U4 R3 D2 NL3 D2 BR2"
    ELSEIF NUM$ = "9" THEN
        DRAW COR$ + "R3 U4 L3 D2 R3 D2 BR3"
    ELSEIF NUM$ = "0" THEN
        DRAW COR$ + "U4 R3 D4 NL3 BR2"
    ELSEIF NUM$ = "-" THEN
        DRAW COR$ + "BU2 R3 BD2"
    END IF
NEXT J%
RETURN
END SUB

```

SUB Numera.Mat (Nelem%, Coord(), LNods%(), Matno%(), NoNod%())

NUMERACAO GRAFICA DOS MATERIAIS

```
FOR iElem% = 1 TO Nelem%
  NNode% = NoNod%(iElem%)
  IF NNode% = 3 THEN
    XNode1% = LNods%(1, iElem%)
    XNode2% = LNods%(2, iElem%)
    XNode3% = LNods%(3, iElem%)
    X1 = Coord(1, XNode1%): Y1 = Coord(2, XNode1%)
    X2 = Coord(1, XNode2%): Y2 = Coord(2, XNode2%)
    X3 = Coord(1, XNode3%): Y3 = Coord(2, XNode3%)
    xM = (X1 + X2 + X3) / 3
    yM = (Y1 + Y2 + Y3) / 3
  ELSEIF NNode% = 6 THEN
    XNode1% = LNods%(1, iElem%)
    XNode3% = LNods%(3, iElem%)
    XNode6% = LNods%(6, iElem%)
    X1 = Coord(1, XNode1%): Y1 = Coord(2, XNode1%)
    X3 = Coord(1, XNode3%): Y3 = Coord(2, XNode3%)
    X6 = Coord(1, XNode6%): Y6 = Coord(2, XNode6%)
    xM = (X1 + X3 + X6) / 3
    yM = (Y1 + Y3 + Y6) / 3
  ELSE
    IF NNode% = 5 THEN
      XNode2% = LNods%(2, iElem%)
      XNode5% = LNods%(5, iElem%)
      X1 = Coord(1, XNode2%)
      Y1 = Coord(2, XNode2%)
      X2 = Coord(1, XNode5%)
      Y2 = Coord(2, XNode5%)
    ELSE
      XNode2% = LNods%(2, iElem%)
      XNode6% = LNods%(6, iElem%)
      X1 = Coord(1, XNode2%)
      Y1 = Coord(2, XNode2%)
      X2 = Coord(1, XNode6%)
      Y2 = Coord(2, XNode6%)
    END IF
    xM = (X1 + X2) * .5: yM = (Y1 + Y2) * .5
  END IF
  PSET (xM, yM), 0: DRAW "C12 BD1 BR1 L2 U2 R2 D2 BU1 BR2"
  VNUM$ = MID$(STR$(Matno%(iElem%)), 2, LEN(STR$(Matno%(iElem%))))
  GOSUB 8200
NEXT iElem%
EXIT SUB
```

DESENHO DOS NUMEROS

```
8200 COR$ = "C12"
FOR J% = 1 TO LEN(VNUM$)
  NUM$ = MID$(VNUM$, J%, 1)
  IF NUM$ = "1" THEN
    DRAW COR$ + "BR2 U4 G2 BD2 BR5"
  ELSEIF NUM$ = "2" THEN
    DRAW COR$ + "NR3 U2 R3 U2 L3 BD4 BR5"
  ELSEIF NUM$ = "3" THEN
    DRAW COR$ + "R3 U2 NL2 U2 L3 BD4 BR6"
  ELSEIF NUM$ = "4" THEN
    DRAW COR$ + "BR3 U4 G3 R4 BD1 BR2"
  ELSEIF NUM$ = "5" THEN
    DRAW COR$ + "R3 U2 L3 U2 R3 BD4 BR2"
  ELSEIF NUM$ = "6" THEN
    DRAW COR$ + "NR3 U4 R3 BD2 NL3 D2 BR3"
  END IF
NEXT J%
```

```
ELSEIF NUM$ = "7" THEN
  DRAW COR$ + "BU3 U1 R3 D2 NL1 NR1 D2 BR3"
ELSEIF NUM$ = "8" THEN
  DRAW COR$ + "BR1 NR3 U4 R3 D2 NL3 D2 BR2"
ELSEIF NUM$ = "9" THEN
  DRAW COR$ + "R3 U4 L3 D2 R3 D2 BR3"
ELSEIF NUM$ = "0" THEN
  DRAW COR$ + "U4 R3 D4 NL3 BR2"
ELSEIF NUM$ = "-" THEN
  DRAW COR$ + "BU2 R3 BD2 "
END IF
```

```

  X2 = Defor(1, LNode$(iNode) + 1, iElem), iCase)
NEXT J%
RETURN
ND SUB
  X2 = Defor(1, LNode$(iNode), iElem), iCase)
  Y2 = Defor(2, LNode$(iNode), iCase)
ELSEIF iNode = NNode AND NNode < 5 THEN
  X2 = Defor(1, LNode$(5), iElem), iCase)
  Y2 = Defor(2, LNode$(5), iCase)
END IF
LINE (X1, Y1)-(X2, Y2), J, , ANDED
NEXT iNode
NEXT iElem
END
```

SUB Des.Displac (iCase%, Nelem%, NoNod%(), LNods%(), Defor())

DESENHO DA DEFORMADA

```

FOR iElem% = 1 TO Nelem%
  NNode% = NoNod%(iElem%)
  IF NNode% = 9 THEN NNode% = 8
  FOR iNode% = 1 TO NNode%
    X1 = Defor(1, LNods%(iNode%, iElem%), iCase%)
    Y1 = Defor(2, LNods%(iNode%, iElem%), iCase%)
    IF iNode% < NNode% THEN
      X2 = Defor(1, LNods%(iNode% + 1, iElem%), iCase%)
      Y2 = Defor(2, LNods%(iNode% + 1, iElem%), iCase%)
    ELSEIF iNode% = NNode% AND NNode% <> 5 THEN
      X2 = Defor(1, LNods%(1, iElem%), iCase%)
      Y2 = Defor(2, LNods%(1, iElem%), iCase%)
    ELSEIF iNode% = NNode% AND NNode% = 5 THEN
      X2 = Defor(1, LNods%(5, iElem%), iCase%)
      Y2 = Defor(2, LNods%(5, iElem%), iCase%)
    END IF
    LINE (X1, Y1)-(X2, Y2), 7, , &HCOCO
  NEXT iNode%
NEXT iElem%

```

```

END SUB
IF NUM% = 1 THEN
  DRAW COR% + " 50 10 10 50 50 50"
ELSEIF NUM% = 2 THEN
  DRAW COR% + " 50 10 10 50 10 50"
ELSEIF NUM% = 3 THEN
  DRAW COR% + " 50 10 10 50 10 10"
ELSEIF NUM% = 4 THEN
  DRAW COR% + " 50 10 10 50 10 10"
ELSEIF NUM% = 5 THEN
  DRAW COR% + " 50 10 10 50 10 10"
ELSEIF NUM% = 6 THEN
  DRAW COR% + " 50 10 10 50 10 10"
ELSEIF NUM% = 7 THEN
  DRAW COR% + " 50 10 10 50 10 10"
ELSEIF NUM% = 8 THEN
  DRAW COR% + " 50 10 10 50 10 10"
ELSEIF NUM% = 9 THEN
  DRAW COR% + " 50 10 10 50 10 10"

```

SUB Numera Gauss (Nelem%, NGauQ%, NGauT%, NoNod%), CGauss()

\*\*\* DESENHO DOS PONTOS DE GAUSS

```
FOR iElem% = 1 TO Nelem%
  NNode% = NoNod%(iElem%)
  NGauss% = NGauQ%
  IF NNode% = 3 OR NNode% = 6 THEN NGauss% = NGauT%
  FOR iGauss% = 1 TO NGauss%
    PSET (CGauss(1, iGauss%, iElem%), CGauss(2, iGauss%, iElem%)), 0: DR
W "C14 BD1 BR1 L2 U2 R2 D2 BU1 BR2"
    VNUM$ = MID$(STR$(iGauss%), 2, LEN(STR$(iGauss%)))
    GOSUB 8100
  NEXT iGauss%
NEXT iElem%
EXIT SUB
```

DESENHO DOS NUMEROS

8100 COR\$ = "C14"

```
FOR J% = 1 TO LEN(VNUM$)
  NUM$ = MID$(VNUM$, J%, 1)
  IF NUM$ = "1" THEN
    DRAW COR$ + "BR2 U4 G2 BD2 BR5"
  ELSEIF NUM$ = "2" THEN
    DRAW COR$ + "NR3 U2 R3 U2 L3 BD4 BR5"
  ELSEIF NUM$ = "3" THEN
    DRAW COR$ + "R3 U2 NL2 U2 L3 BD4 BR6 "
  ELSEIF NUM$ = "4" THEN
    DRAW COR$ + "BR3 U4 G3 R4 BD1 BR2 "
  ELSEIF NUM$ = "5" THEN
    DRAW COR$ + "R3 U2 L3 U2 R3 BD4 BR2"
  ELSEIF NUM$ = "6" THEN
    DRAW COR$ + "NR3 U4 R3 BD2 NL3 D2 BR3"
  ELSEIF NUM$ = "7" THEN
    DRAW COR$ + "BU3 U1 R3 D2 NL1 NR1 D2 BR3"
  ELSEIF NUM$ = "8" THEN
    DRAW COR$ + "BR1 NR3 U4 R3 D2 NL3 D2 BR2"
  ELSEIF NUM$ = "9" THEN
    DRAW COR$ + "R3 U4 L3 D2 R3 D2 BR3"
  ELSEIF NUM$ = "0" THEN
    DRAW COR$ + "U4 R3 D4 NL3 BR2"
  ELSEIF NUM$ = "-" THEN
    DRAW COR$ + "BU2 R3 BD2 "
  END IF
NEXT J%
RETURN
END SUB
```

```
SUB Des.PStress (Zoom%, NGauQ%, NGauT%, Nelem%, iCase%, NoNod(), PStr1(), PStr2  
( ), CGauss(), Angle())
```

```
*** DESENHO DAS TENSOES PRINCIPAIS
```

```
VIEW SCREEN (3, 39)-(636, 332)
```

```
Pi = 3.141592654#
```

```
Stre1$ = "BD3L5D3F1R2E1U1BR3E1D5"
```

```
Stre2$ = "BD3L5D3F1R2E1U1BR3BU1R3D3L3D3R3"
```

```
FOR iElem% = 1 TO Nelem%
```

```
  NNode% = NoNod%(iElem%)
```

```
  NGauss% = NGauQ%
```

```
  IF NNode% = 3 OR NNode% = 6 THEN NGauss% = NGauT%
```

```
  FOR iGauss% = 1 TO NGauss%
```

```
    GX = CGauss(1, iGauss%, iElem%)
```

```
    GY = CGauss(2, iGauss%, iElem%)
```

```
    Ang = Angle(iGauss%, iElem%, iCase%)
```

```
    Stre1 = PStr1(iGauss%, iElem%, iCase%)
```

```
    Stre2 = PStr2(iGauss%, iElem%, iCase%)
```

```
    S1X = Stre1 * COS(Ang)
```

```
    S1Y = Stre1 * SIN(Ang)
```

```
    S2X = Stre2 * COS(Pi * .5 + Ang)
```

```
    S2Y = Stre2 * SIN(Pi * .5 + Ang)
```

```
    Stre = Stre1: S1 = ABS(S1X): S2 = ABS(S1Y): StreP$ = Stre1$
```

```
    FOR iStre% = 1 TO 2
```

```
      SinalAng = SGN(Ang)
```

```
      SinAbs = ABS(SIN(Pi * .5 - ABS(Ang)))
```

```
      CosAbs = ABS(COS(Pi * .5 - ABS(Ang)))
```

```
    IF Stre < 0 THEN
```

```
      X1 = GX + SinalAng * S1 * .5 + CosAbs
```

```
      Y1 = GY - S2 * .5 + SinalAng * SinAbs
```

```
      X2 = GX - SinalAng * S1 * .5 + CosAbs
```

```
      Y2 = GY + S2 * .5 + SinalAng * SinAbs
```

```
      PSET (X1, Y1), 0
```

```
      LINE -(X2, Y2), 7
```

```
      X1 = GX + SinalAng * S1 * .5 - CosAbs
```

```
      Y1 = GY - S2 * .5 - SinalAng * SinAbs
```

```
      X2 = GX - SinalAng * S1 * .5 - CosAbs
```

```
      Y2 = GY + S2 * .5 - SinalAng * SinAbs
```

```
      PSET (X1, Y1), 0
```

```
      LINE -(X2, Y2), 7
```

```
    ELSE
```

```
      X1 = GX + SinalAng * S1 * .5
```

```
      Y1 = GY - S2 * .5
```

```
      X2 = GX - SinalAng * S1 * .5
```

```
      Y2 = GY + S2 * .5
```

```
      PSET (X1, Y1), 0
```

```
      LINE -(X2, Y2), 7
```

```
    END IF
```

```
  IF Zoom% = 1 THEN DRAW StreP$
```

```
  Stre = Stre2: S1 = ABS(S2X): S2 = ABS(S2Y): StreP$ = Stre2$
```

```
  Ang = Ang - SinalAng * Pi * .5
```

```
  NEXT iStre%
```

```
  NEXT iGauss%
```

```
  NEXT iElem%
```

```
END SUB
```

```
UB ZoomWindow (Zoom%, CrossPointer%(), Nelem%, NoNod%(), Coord(), LNods%())
```

```
IF Zoom% = 1 THEN  
WINDOW: VIEW  
VIEW SCREEN (3, 39)-(636, 332): CLS 1: VIEW  
CALL Des.Mesh(Nelem%, NoNod%(), Coord(), LNods%())  
ELSE  
VIEW SCREEN (3, 39)-(636, 332)  
Zoom% = 0  
END IF
```

```
MouseInstall mflag%
```

```
IF mflag% THEN  
MouseRange 3, 40, 636, 332  
MouseShow
```

```
ainLoop:
```

```
DO  
MouseNow leftButton%, rightButton%, xMouse%, yMouse%  
MousePressLeft leftCount%, xpLeft%, ypLeft%  
MouseReleaseLeft leftReleaCount%, xrLeft%, yrLeft%  
MousePressRight rightCount%, xpRight%, ypRight%  
MouseReleaseRight rightCount%, xrRight%, yrRight%  
IF leftReleaCount% <> 0 THEN GOSUB DrawLine  
leftReleaCount% = 0  
LOOP UNTIL INKEY$ <> ""  
Zoom% = 0: MouseHide  
EXIT SUB
```

```
DrawLine:
```

```
IF ABS(xrLeft% - xpLeft%) < 10 OR ABS(ypLeft% - yrLeft%) < 10 THEN  
RETURN  
END IF  
MouseHide  
DO  
xM% = xMouse%: yM% = yMouse%  
LINE (xpLeft%, ypLeft%)-(xMouse% - 1, yMouse% - 1), 15, B, &H101  
LINE (xpLeft%, ypLeft%)-(xM% - 1, yM% - 1), 0, B, &H101  
MouseNow leftButton%, rightButton%, xMouse%, yMouse%  
MousePressLeft leftPressCount%, xpPressLeft%, ypPressLeft%  
LOOP UNTIL leftPressCount% <> 0  
leftPressCount% = 0  
WINDOW SCREEN (xpLeft% - 15, ypLeft% - 40)-(xM% + 15, yM% + 15)  
VIEW SCREEN (3, 39)-(636, 332): CLS 1  
CALL Des.Mesh(Nelem%, NoNod%(), Coord(), LNods%())  
Zoom% = 1: EXIT SUB
```

```
ELSE
```

```
COLOR 0: PRINT STRING$(79, 219); : COLOR 7  
Prompt1$ = " Para definir a janela : " + CHR$(27) + " " + CHR$(24) + "  
+ CHR$(25) + " " + CHR$(26) + " " + CHR$(27) + " " + CHR$(24) + "  
LOCATE , 2: PRINT Prompt1$;  
RESTORE DataCrossPointer  
FOR I% = 0 TO 15  
READ h$  
CrossPointer%(I%) = VAL("&H" + h$)  
NEXT I%
```

```
XPut% = 300: YPut% = 150  
DO
```

```
kee% = InKeyCode%  
IF kee% = 20480 THEN  
YPut% = YPut% + 3  
IF YPut% > 326 THEN YPut% = 326  
GOSUB PutCross
```



```

ELSEIF kee% = 18432 THEN
  YPut% = YPut% - 3
  IF YPut% < 39 THEN YPut% = 39
  GOSUB PutCross
ELSEIF kee% = 19200 THEN
  XPut% = XPut% - 3
  IF XPut% < 3 THEN XPut% = 3
  GOSUB PutCross
ELSEIF kee% = 19712 THEN
  XPut% = XPut% + 3
  IF XPut% > 630 THEN XPut% = 630
  GOSUB PutCross
ELSEIF kee% = 13 THEN
  xStart% = XPut%: yStart% = YPut%
  DO
    LINE (xStart%, yStart%)-(XPut%, YPut%), 7, B, &HF000
    FOR Itime% = 1 TO 5: NEXT Itime%
    LINE (xStart%, yStart%)-(XPut%, YPut%), 0, B, &HF000

```

```

    kee% = InKeyCode%

```

```

    IF kee% = 20480 THEN

```

```

      YPut% = YPut% + 3

```

```

      IF YPut% > 326 THEN YPut% = 326

```

```

      GOSUB PutCross

```

```

    ELSEIF kee% = 18432 THEN

```

```

      YPut% = YPut% - 3

```

```

      IF YPut% < 39 THEN YPut% = 39

```

```

      GOSUB PutCross

```

```

    ELSEIF kee% = 19200 THEN

```

```

      XPut% = XPut% - 3

```

```

      IF XPut% < 3 THEN XPut% = 3

```

```

      GOSUB PutCross

```

```

    ELSEIF kee% = 19712 THEN

```

```

      XPut% = XPut% + 3

```

```

      IF XPut% > 630 THEN XPut% = 630

```

```

      GOSUB PutCross

```

```

    ELSEIF kee% = 0 THEN

```

```

      GOSUB PutCross

```

```

    END IF

```

```

  LOOP WHILE kee% <> 13

```

```

  IF xStart% = XPut% OR yStart% = YPut% THEN

```

```

    xStart% = 15: yStart% = 40: XPut% = 625: YPut% = 335

```

```

    Zoom% = 0

```

```

  ELSE

```

```

    Zoom% = 1

```

```

  END IF

```

```

  WINDOW SCREEN (xStart% - 15, yStart% - 40)-(XPut% + 18, YPut% + 18)

```

```

  VIEW SCREEN (3, 39)-(636, 332): CLS 1

```

```

  CALL Des.Mesh(Nelem%, NoNod(), Coord(), LNods())

```

```

  EXIT SUB

```

```

ELSEIF kee% = 27 THEN

```

```

  Zoom% = 0: EXIT SUB

```

```

ELSEIF kee% = 0 THEN

```

```

  GOSUB PutCross

```

```

END IF

```

```

LOOP

```

```

END IF

```

```

PutCross:

```

```

  PUT (XPut%, YPut%), CrossPointer%, XOR

```

```

  FOR Itime% = 1 TO 5: NEXT Itime%

```

```

  PUT (XPut%, YPut%), CrossPointer%, XOR

```

```

  RETURN

```

```

ND SUB

```

SUB Des.Mesh (Nelem%, NoNod%(), Coord(), LNods%())

\*\*\* DESENHO DOS ELEMENTOS

```

FOR iElem% = 1 TO Nelem%
  NNode% = NoNod%(iElem%)
  IF NNode% = 9 THEN NNode% = 8
  FOR iNode% = 1 TO NNode%
    X1 = Coord(1, LNods%(iNode%, iElem%))
    Y1 = Coord(2, LNods%(iNode%, iElem%))
    IF iNode% < NNode% THEN
      X2 = Coord(1, LNods%(iNode% + 1, iElem%))
      Y2 = Coord(2, LNods%(iNode% + 1, iElem%))
    ELSEIF iNode% = NNode% AND NNode% <> 5 THEN
      X2 = Coord(1, LNods%(1, iElem%))
      Y2 = Coord(2, LNods%(1, iElem%))
    ELSEIF iNode% = NNode% AND NNode% = 5 THEN
      X2 = Coord(1, LNods%(5, iElem%))
      Y2 = Coord(2, LNods%(5, iElem%))
    END IF
    LINE (X1, Y1)-(X2, Y2), 4
  NEXT iNode%
NEXT iElem%
END SUB

```

END SUB

```

Coord(1, iPoint) = 0
Coord(2, iPoint) = 0
NEXT iPoint

```

DESENHO DOS ELEMENTOS

```

FOR iElem% = 1 TO Nelem%
  NNode% = NoNod%(iElem%)
  IF NNode% = 9 THEN NNode% = 8
  FOR iNode% = 1 TO NNode%
    X1 = Coord(1, LNods%(iNode%, iElem%))
    Y1 = Coord(2, LNods%(iNode%, iElem%))
    IF iNode% < NNode% THEN
      X2 = Coord(1, LNods%(iNode% + 1, iElem%))
      Y2 = Coord(2, LNods%(iNode% + 1, iElem%))
    ELSEIF iNode% = NNode% AND NNode% <> 5 THEN
      X2 = Coord(1, LNods%(1, iElem%))
      Y2 = Coord(2, LNods%(1, iElem%))
    ELSEIF iNode% = NNode% AND NNode% = 5 THEN
      X2 = Coord(1, LNods%(5, iElem%))
      Y2 = Coord(2, LNods%(5, iElem%))
    END IF
    LINE (X1, Y1)-(X2, Y2), 4
  NEXT iNode%
NEXT iElem%

```

DESENHO DOS ELEMENTOS

```

Stress(1, iElem%) = 0
Stress(2, iElem%) = 0
FOR iNode% = 1 TO NNode%
  X1 = Coord(1, LNods%(iNode%, iElem%))
  Y1 = Coord(2, LNods%(iNode%, iElem%))
  IF iNode% < NNode% THEN
    X2 = Coord(1, LNods%(iNode% + 1, iElem%))
    Y2 = Coord(2, LNods%(iNode% + 1, iElem%))
  ELSEIF iNode% = NNode% AND NNode% <> 5 THEN
    X2 = Coord(1, LNods%(1, iElem%))
    Y2 = Coord(2, LNods%(1, iElem%))
  ELSEIF iNode% = NNode% AND NNode% = 5 THEN
    X2 = Coord(1, LNods%(5, iElem%))
    Y2 = Coord(2, LNods%(5, iElem%))
  END IF
  LINE (X1, Y1)-(X2, Y2), 4
NEXT iNode%

```

```
SUB Scale (NGauQ%, NGauT%, NCase%, NPoint, Nelem%, LNods(), NoNod(), Coord(),  
Defor(), CGauss(), PStr1(), PStr2(), Angle())
```

— ESCALA DAS COORDENADAS

```
Aspect = 48! / 35!: XLFT = 30!: XRHT = 610!: YHIGH = 300!: YLOW = 30!  
Xmin = Coord(1, 1): Xmax = Coord(1, 1): Ymin = Coord(2, 1): Ymax = Coord(2,
```

1)

```
FOR iPoint% = 2 TO NPoint%  
  IF Coord(1, iPoint%) < Xmin THEN Xmin = Coord(1, iPoint%)  
  IF Coord(1, iPoint%) > Xmax THEN Xmax = Coord(1, iPoint%)  
  IF Coord(2, iPoint%) < Ymin THEN Ymin = Coord(2, iPoint%)  
  IF Coord(2, iPoint%) > Ymax THEN Ymax = Coord(2, iPoint%)  
NEXT iPoint%
```

```
XL = ABS(Xmax - Xmin): YL = ABS(Ymax - Ymin)  
ScaleX = (XRHT - XLFT) / XL: ScaleY = ScaleX / Aspect  
IF (YL ≠ ScaleY) < (YHIGH - YLOW) GOTO 2000  
ScaleY = (YHIGH - YLOW) / YL: ScaleX = ScaleY * Aspect
```

```
2000 X0 = XLFT + (XRHT - XLFT) / 2! - ScaleX * (Xmax + Xmin) / 2!  
Y0 = YLOW + (YHIGH - YLOW) / 2! - ScaleY * (Ymax + Ymin) / 2!
```

— TRANSFORMACAO DAS COORDENADAS REAIS

```
FOR iPoint% = 1 TO NPoint%  
  X1 = X0 + ScaleX * Coord(1, iPoint%): Y1 = 350! - (Y0 + ScaleY * Coord(2,  
iPoint%))  
  Coord(1, iPoint%) = X1  
  Coord(2, iPoint%) = Y1  
NEXT iPoint%
```

— TRANSFORMACAO DAS COORDENADAS REAIS DOS PONTOS DE GAUSS

```
FOR iElem% = 1 TO Nelem%  
  NNode% = NoNod%(iElem%)  
  NGauss% = NGauQ%  
  IF NNode% = 3 OR NNode% = 6 THEN NGauss% = NGauT%  
  FOR iGauss% = 1 TO NGauss%  
    CGauss(1, iGauss%, iElem%) = X0 + ScaleX * CGauss(1, iGauss%, iElem%)  
    CGauss(2, iGauss%, iElem%) = 350! - (Y0 + ScaleY * CGauss(2, iGauss%,  
iElem%))  
  NEXT iGauss%  
NEXT iElem%
```

— DETERMINACAO DAS ESCALAS DAS TENSOES

```
StressMax = -1000000000000000#: StressMin = 1000000000000000#  
FOR iCase% = 1 TO NCase%  
  FOR iElem% = 1 TO Nelem%  
    NNode% = NoNod%(iElem%)  
    NGauss% = NGauQ%  
    IF NNode% = 3 OR NNode% = 6 THEN NGauss% = NGauT%  
    FOR iGauss% = 1 TO NGauss%  
      IF PStr1(iGauss%, iElem%, iCase%) > StressMax THEN StressMax = PSt  
1(iGauss%, iElem%, iCase%)  
      IF PStr1(iGauss%, iElem%, iCase%) < StressMin THEN StressMin = PSt  
1(iGauss%, iElem%, iCase%)  
      IF PStr2(iGauss%, iElem%, iCase%) > StressMax THEN StressMax = PSt  
2(iGauss%, iElem%, iCase%)  
      IF PStr2(iGauss%, iElem%, iCase%) < StressMin THEN StressMin = PSt  
2(iGauss%, iElem%, iCase%)  
    NEXT iGauss%  
  NEXT iElem%  
NEXT iCase%
```

```

MaxStress = 0
IF ABS(StressMax) > MaxStress THEN MaxStress = ABS(StressMax)
IF ABS(StressMin) > MaxStress THEN MaxStress = ABS(StressMin)
ScaleT = 10 / MaxStress

```

---

TRANSFORMACAO DAS TENGUES EM COORDENADAS GRAFICAS

---

```

FOR iCase% = 1 TO NCase%
  FOR iElem% = 1 TO Nelem%
    NNode% = NoNod%(iElem%)
    NGauss% = NGauQ%
    IF NNode% = 3 OR NNode% = 6 THEN NGauss% = NGauT%
    FOR iGauss% = 1 TO NGauss%
      PStr1(iGauss%, iElem%, iCase%) = PStr1(iGauss%, iElem%, iCase%) * ScaleT * 2
      PStr2(iGauss%, iElem%, iCase%) = PStr2(iGauss%, iElem%, iCase%) * ScaleT * 2
    NEXT iGauss%
  NEXT iElem%
NEXT iCase%

```

---

DETERMINACAO DA ESCALA DOS DESLOCAMENTOS

---

```

DEXMAX = Defor(1, 1, 1): DEXMIN = Defor(1, 1, 1)
DEYMAX = Defor(2, 1, 1): DEYMIN = Defor(2, 1, 1)
FOR iPoin% = 2 TO NPoin%
  FOR iCase% = 1 TO NCase%
    IF Defor(1, iPoin%, iCase%) > DEXMAX THEN DEXMAX = Defor(1, iPoin%, iCase%)
    IF Defor(1, iPoin%, iCase%) < DEXMIN THEN DEXMIN = Defor(1, iPoin%, iCase%)
    IF Defor(2, iPoin%, iCase%) > DEYMAX THEN DEYMAX = Defor(2, iPoin%, iCase%)
    IF Defor(2, iPoin%, iCase%) < DEYMIN THEN DEYMIN = Defor(2, iPoin%, iCase%)
  NEXT iCase%
NEXT iPoin%
DESMAX = 0
IF ABS(DEXMAX) > DESMAX THEN DESMAX = ABS(DEXMAX)
IF ABS(DEXMIN) > DESMAX THEN DESMAX = ABS(DEXMIN)
IF ABS(DEYMAX) > DESMAX THEN DESMAX = ABS(DEYMAX)
IF ABS(DEYMIN) > DESMAX THEN DESMAX = ABS(DEYMIN)
Scaled = 5 / DESMAX

```

---

TRANSFORMACAO DOS DESLOCAMENTOS EM COORDENADAS GRAFICAS

---

```

FOR iCase% = 1 TO NCase%
  FOR iPoin% = 1 TO NPoin%
    Defor(1, iPoin%, iCase%) = Coord(1, iPoin%) + Defor(1, iPoin%, iCase%) * Scaled * 2
    Defor(2, iPoin%, iCase%) = Coord(2, iPoin%) - Defor(2, iPoin%, iCase%) * Scaled
  NEXT iPoin%
NEXT iCase%
END SUB

```

```
SUB SetAspect (Angle1%, Angle2%) STATIC
XaAspect# = SIN((Angle1% / 180) * (22 / 7))
XbAspect# = COS((Angle1% / 180) * (22 / 7))
YaAspect# = SIN((Angle2% / 180) * (22 / 7))
YbAspect# = COS((Angle2% / 180) * (22 / 7))
END SUB
```

```
SUB MaxMin (MNode%, MGauss%, NodeQ%, NodeI%, NodeT%, NGauQ%, NGauT%)
*** DETERMINACAO DO NQ DE NOS MAXIMO POR ELEMENTO E NQ MAXIMO DE
*** PONTOS DE GAUSS
MNode% = 0
IF NodeQ% <> 0 THEN
MNode% = NodeQ%
ELSEIF NodeQ% = 0 AND NodeI% > NodeT% THEN
MNode% = NodeI%
ELSEIF (NodeQ% = 0 AND NodeI% < NodeT%) THEN
MNode% = NodeT%
END IF
IF NGauQ% > NGauT% THEN
MGauss% = NGauQ%
ELSE
MGauss% = NGauT%
END IF
END SUB
```

```
IF NODOS < 7 THEN
TOTAL = NODOS * 2
FOR I = 1 TO NODOS
FOR J = 1 TO 2
NEXT J
NEXT I
END IF
```

SUB Node2D (Nodes%(), Coord(), LNods%(), NPoin%, Nelem%, NoNod%())



```
*** GERA AS COORDENADAS DE:  
* NOS SITUADOS A MEIO DE LADOS RECTILINEOS DE ELEMENTOS PLANOS  
  DE 6, 8 E 9 Nos  
* No CENTRAL DE ELEMENTOS DE 9 Nos
```

```
*** CICLO SOBRE OS ELEMENTOS
```

```
FOR iElem% = 1 TO Nelem%
```

```
  NNode% = NoNod%(iElem%)
```

```
  IF NNode% = 3 OR NNode% = 4 OR NNode% = 5 THEN GOTO 1000
```

```
  FOR iNode% = 1 TO NNode%
```

```
    Nodes%(iNode%) = LNods%(iNode%, iElem%)
```

```
  NEXT iNode%
```

```
  FOR INOD2 = 2 TO NNode% STEP 2
```

```
    IPOI2 = Nodes%(INOD2)
```

```
    TOTAL = ABS(Coord(1, IPOI2)) + ABS(Coord(2, IPOI2))
```

```
    IF TOTAL = 0 THEN
```

```
      INOD1 = INOD2 - 1
```

```
      INOD3 = INOD2 + 1
```

```
      IF INOD3 >= NNode% THEN INOD3 = 1
```

```
      IPOI1 = Nodes%(INOD1)
```

```
      IPOI3 = Nodes%(INOD3)
```

```
      FOR iDime% = 1 TO 2
```

```
        Coord(iDime%, IPOI2) = (Coord(iDime%, IPOI1) + Coord(iDime%, I
```

```
POI3)) * .5
```

```
      NEXT iDime%
```

```
    END IF
```

```
  NEXT INOD2
```

```
*** No CENTRAL ( ELEMENTOS DE 9 Nos)
```

```
IF NNode% = 9 THEN
```

```
  IPOI9 = Nodes%(9)
```

```
  TOTAL = ABS(Coord(1, IPOI9)) + ABS(Coord(2, IPOI9))
```

```
  IF TOTAL = 0 THEN
```

```
    FOR iNode% = 1 TO 7 STEP 2
```

```
      NCORN = Nodes%(iNode%)
```

```
      NMIDS = Nodes%(iNode% + 1)
```

```
      FOR iDime% = 1 TO 2
```

```
        Coord(iDime%, IPOI9) = Coord(iDime%, IPOI9) - Coord(iDime%,  
NCORN) / 4 + Coord(iDime%, NMIDS) / 2
```

```
      NEXT iDime%
```

```
    NEXT iNode%
```

```
  END IF
```

```
END IF
```

```
1000 NEXT iElem%
```

```
END SUB
```

```
SUB Elem5 (AUX(), jElem%, LNods%())
```

```
*** ORDENA AS LIGACOES PARA ELEMENTOS INFINITOS
```

```
FOR IAUX = 1 TO 5
```

```
  AUX(IAUX) = LNods%(IAUX, jElem%)
```

```
NEXT IAUX
```

```
LNods%(1, jElem%) = AUX(5)
```

```
FOR IAUX = 1 TO 4
```

```
  LNods%(IAUX + 1, jElem%) = AUX(IAUX)
```

```
NEXT IAUX
```

```
END SUB
```

```

SUB Main.Menu (Item$, Choice%) STATIC
  STATIC X$, Length%, Place, Max.Num, First%, Last
  FG = 7: BG = 4: COLOR FG
  Max.Num = UBOUND(Item$)
  WHILE Item$(Max.Num) = ""
    Max.Num = Max.Num - 1
  WEND
  First% = 1
  Last = 1
  Choice% = 1
Clear.Line:
  COLOR 0: PRINT STRING$(79, 219): : COLOR FG
Pre.Print:
  LOCATE , 2
  Place = 1
Print.Loop:
  IF Last > Max.Num GOTO Begin
  Length% = LEN(Item$(Last)) + 2
  IF Place + Length% > 79 GOTO Begin
  IF Last = Choice% THEN COLOR BG
  PRINT SPC(1); LEFT$(Item$(Last), Length% - 2); SPC(1);
  COLOR FG
  Last = Last + 1
  Place = Place + Length%
  GOTO Print.Loop
Begin:
  Last = Last - 1
  IF X$ <> "K" THEN LOCATE , 2: GOTO Get.Key
  FOR X = 1 TO LEN(Item$(Last)) + 2
    PRINT CHR$(29);
  NEXT
Get.Key:
  X$ = INKEY$
  IF X$ = CHR$(13) THEN EXIT SUB
  IF X$ = CHR$(27) THEN
    Choice% = 0
    EXIT SUB
  END IF
  IF LEN(X$) <> 2 GOTO Get.Key
  X$ = RIGHT$(X$, 1)
  ON INSTR("KMG", X$) GOTO Left.Arrow, Right.Arrow, HOME
  GOTO Get.Key
Left.Arrow:
  IF Choice% = 1 GOTO Get.Key

```



Choice% = Choice% - 1

```
IF Choice% < First% THEN
  First% = Choice%
  Length% = 1
  GOSUB Right.To.Left
  Last = First%
  GOTO Clear.Line
END IF
```

```
PRINT SPC(1); Item$(Choice% + 1); SPC(1);
```

```
FOR X = 1 TO LEN(Item$(Choice%)) + LEN(Item$(Choice% + 1)) + 4
  PRINT CHR$(29);
NEXT
```

```
GOSUB Display.Item
GOTO Get.Key
```

Right.Arrow:

```
IF Choice% = Max.Num GOTO Get.Key
Choice% = Choice% + 1
```

```
IF Choice% > Last THEN
  First% = Choice%
  Last = Choice%
  GOTO Clear.Line
END IF
```

```
PRINT SPC(1); Item$(Choice% - 1); SPC(1);
GOSUB Display.Item
GOTO Get.Key
```

## ANEXO II

HOME:

```
Choice% = 1
Last = 1
IF First% = 1 GOTO Pre.Print
First% = 1
GOTO Clear.Line
```

Display Item:

```
COLOR BC
PRINT SPC(1); LEFT$(Item$(Choice%), 77); SPC(1);
```

```
FOR X = 1 TO LEN(Item$(Choice%)) + 2
  PRINT CHR$(29);
NEXT
```

```
COLOR FG
RETURN
```

Right.To.Left:

```
Length% = Length% + LEN(Item$(First%)) + 2
IF First% = 1 THEN RETURN
IF LEN(Item$(First% - 1)) + Length% + 2 > 79 THEN RETURN
First% = First% - 1
GOTO Right.To.Left
```

END SUB

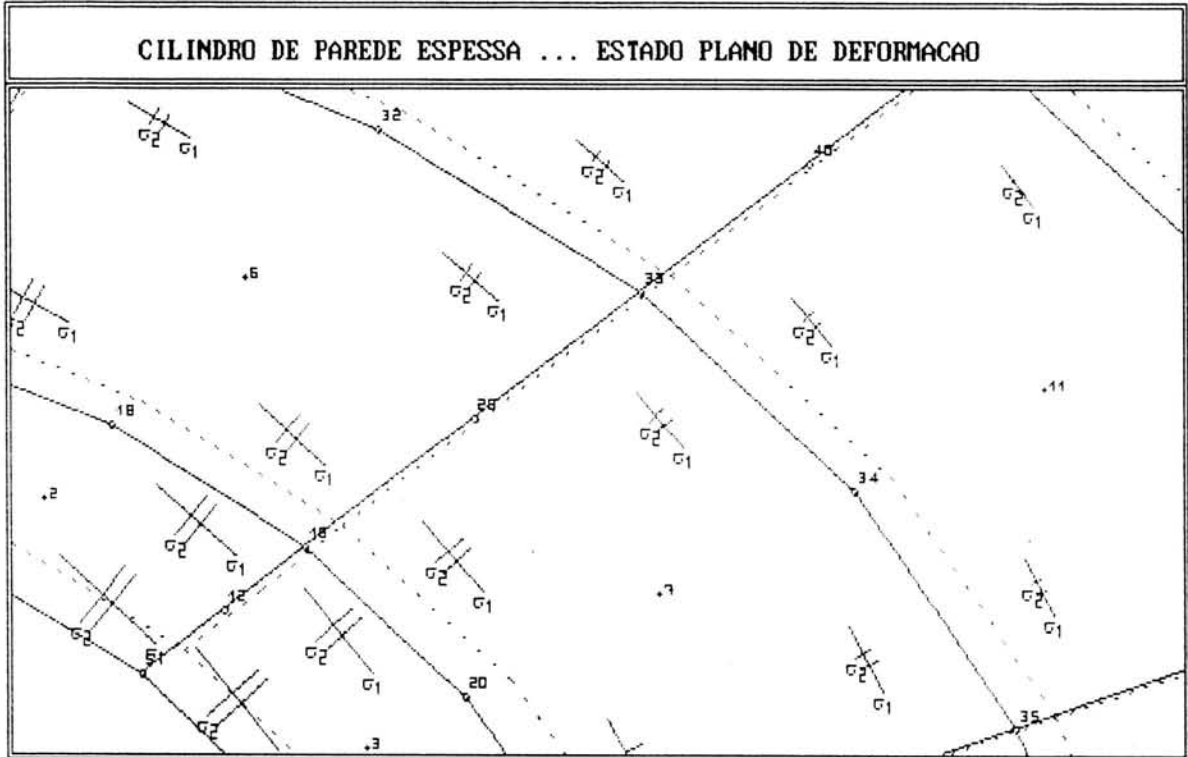
F  
E  
U  
P



**DrawPlane**  
**MÉTODO DOS ELEMENTOS FINITOS**  
**MJLC/1993**

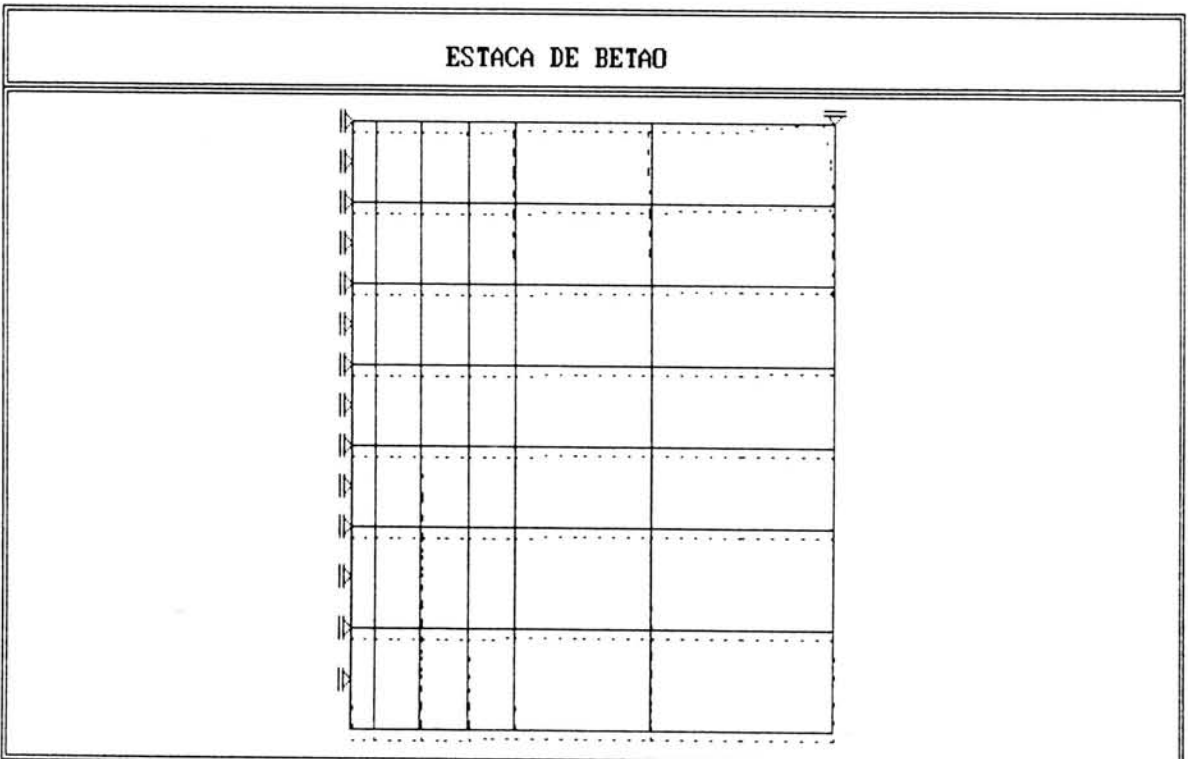
## **ANEXO II**

**EXEMPLOS**

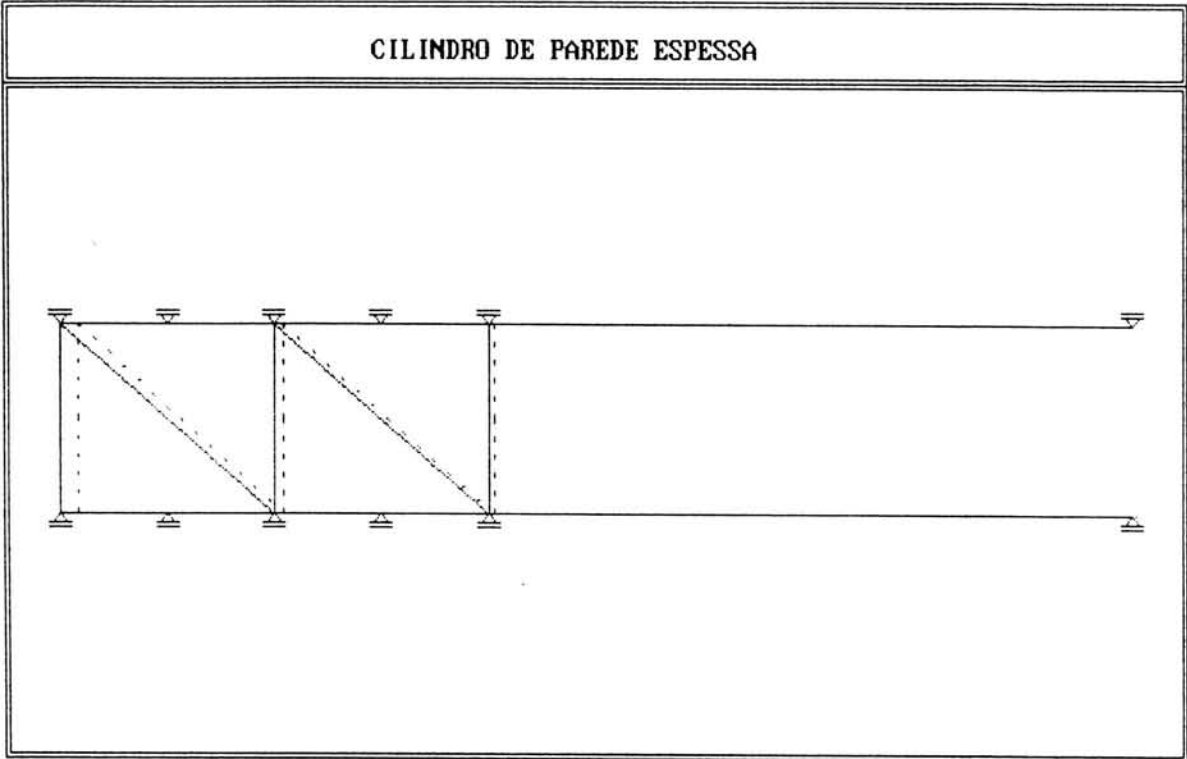


— Compressao : — Traccao

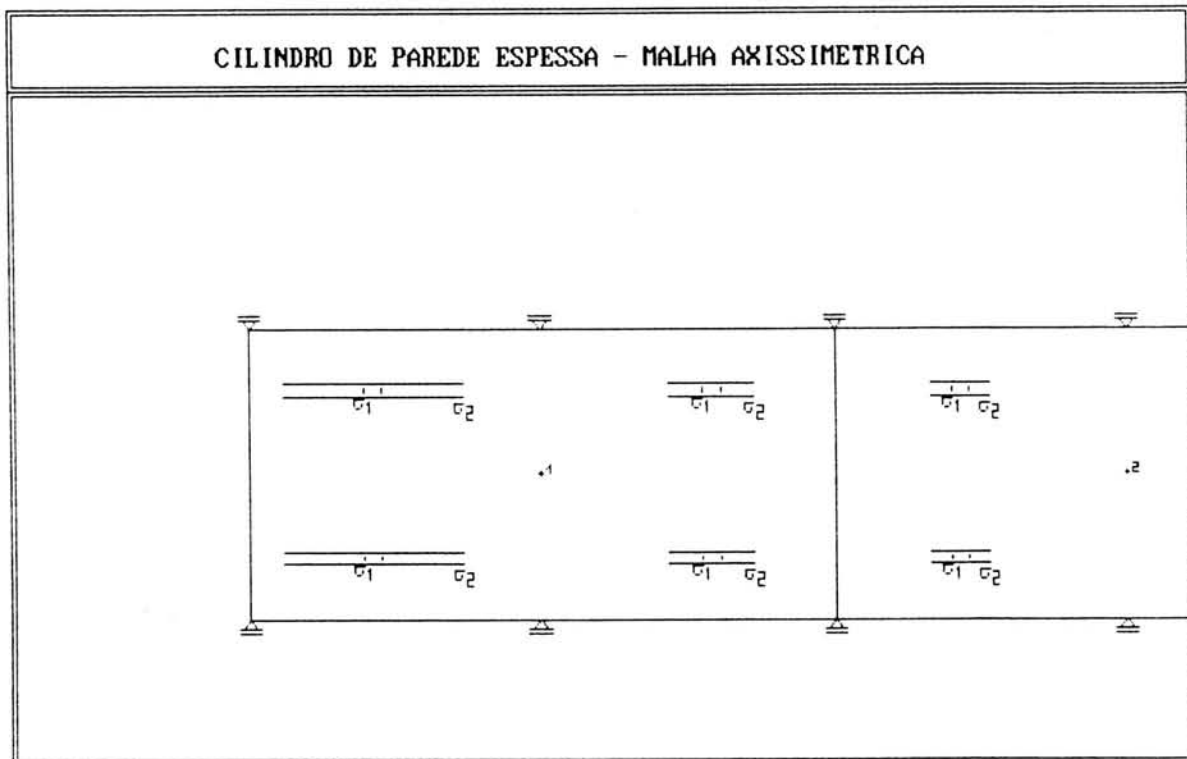
FILE : CIL.DES



FILE : estaca.DES



FILE : 4ft1i.DES



— Compressao : — Traccao

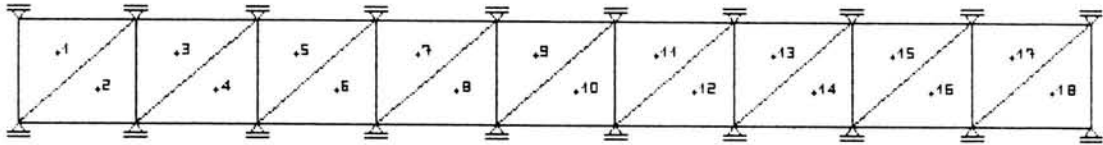
FILE : 9ffx.DES

F  
E  
U  
P



**DrawPlane**  
MÉTODO DOS ELEMENTOS FINITOS  
MJLC/1993

CILINDRO DE PAREDE ESPESSA



FILE : 18ft3dfr.DES

## BIBLIOGRAFIA

- (i) ZIENKIEWICZ, O. C., The Finite Elements Method, 3.rd ed., Mc Graw-Hill, London, 1977.
- (ii) HINTON, E., OWEN, D. R. J., An Introduction to Finite Element Computations, Pineridge Press, Swansea, 1979.
- (iii) HINTON, E., OWEN, D. R. J., Finite Element Programming, Academic Press, London, 1977.
- (iv) MARQUES, J. M. M. C., Textos de Apoio do Seminário "Métodos Numéricos em Geotecnia", FEUP, 1993.
- (v) MARQUES, J. M. M. C., Textos de Apoio de "Complementos de Cálculo Numérico", FEUP, 1993.



FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000101596