# Referências Bibliográficas


[ 1 ] Cook, R. D.; Malkus, D. S.; Plesha, M. E., Concepts and Applications of Finite Element Analysis, Third edition, John Wiley & Sons, 1989.

[ 2 ] Wilson, E. L., Three-Dimensional Static and Dynamic Analysis of Structures, Third edition, Computers & Structures, 2002.

[ 3 ] Zienkiewicz, O. C.; Taylor, R. L., The Finite Element Method, Fifth edition, Butterworth-Heinemann, 2000.

[ 4 ] Bathe, K. J., Finite Element Procedures in Engineering Analysis, Prentice-Hall, 1982.

[ 5 ] Weaver, W. J.; Johnston, P. R., Finite Elements for Structural Analysis, Prentice-Hall, 1984.

[ 6 ] Timoshenko, S. P.; Gere, J. M., Theory of Elastic Stability, Second edition, McGraw-Hill, 1961

[ 7 ] Cheung, Y. K., Finite Strip Method in Structural Analysis, First edition, Pergamon Press, 1976.

[ 8 ] CSI Analysis Reference Manual for SAP 2000, Computers & Structures, 2005.

[ 9 ] Elsgoltz, L., Ecuaciones Diferenciales y Cálculo Variacional, Editorial MIR, 1969.

[ 10 ] Cheung, Y. K.; Au F. T. K., “Isoparametric Spline Finite Strip for Degenerate Shells”, Thin-Walled Struct., 21 (1995) 65-92.

[ 11 ] Au, F. T. K.; Cheung, Y. K., “Free Vibration and Stability Analysis of Shells by the Isoparametric Spline Finite Strip Method”, Thin-Walled Struct., 24 (1996) 53-82.

[ 12 ] Au, F. T. K.; Cheung, Y. K., “Static and Free Vibration Analysis of Variable-depth Bridges of arbitrary alignments Using the Isoparametric Spline Finite Strip Method”, Thin-Walled Struct., 24 (1996) 19-51.

[ 13 ] Choi, C. K.; Kim, K. H.; Hong, H. S., “Spline Finite Strip Analysis of Prestressed Concrete Box-girder Bridges”, Engng. Struct., 24 (2002) 1575-1586.

[ 14 ] Bergamini, A.; Biondini, F., “Finite Strip Modeling for Optimal Desing of Prestressed Folded Plate Structures”, Engng. Struct., 26 (2004) 1043-1054.

[ 15 ] Lau, D, T.; Cheung, M. S.; Cheng, S. H., “3D Flutter Analysis of Bridges by Spline Finite Strip Method”, J. Struct. Engng., 126 (10) (2000) 1246-1254.

[ 16 ] Knight, N. F. J., “Assumed-stress Hybrid Elements with Drilling Degrees of Freedom for Nonlinear Analysis of Composite Structures”, Depart. Aerospace Engng., Final report, Dec. 1995.

[ 17 ] Shuli, S.; Mingwu, Y.; Pu, C., “A Practical Quadrilateral Membrane Element with Drilling Degrees of Freedom” Acta Mech. Solid. 10 (2) (1997) 179-188.

[ 18 ] Papachristidis, A. G.; Badaloukas, G. N.; Badalouka, B. G., “Experimental Verification of Shear Wall Modeling Using Finite Element Analysis”, 6th National Congress of Mechanics, Greece, 2001

[ 19 ] Pimpinelli, G., “An Assumed Strain Quadrilateral Element with Drilling Degrees of Freedom”, Finite Elem. Analysis Design, 41 (2004) 267-283.

# A
# ENTRADA E SAÍDA DE DADOS

O programa utiliza dois arquivos de texto para ingresso de dados denominados “DADOS1” e “DADOS2”. O primeiro contém dados gerais sobre a análise e pode ser utilizado pelo programa para gerar o segundo arquivo com todas as variáveis iguais a zero. As Figuras A.1 e A.2 mostram os dados para uma viga em balanço com carga pontual no extremo livre, dividida transversalmente em dois elementos e com quatro divisões longitudinais.

Os arquivos “RESEST”, “RESDIN” E “RESINS” contêm os resultados da análise estática, dinâmica e de instabilidade, respectivamente. As Figuras A.3, A.4 e A.5 são exemplos de como são apresentados os dados em estes arquivos.



```
PROGRAMA "ELEMENTOS FINITOS COM FUNÇÕES SPLINE"

DADOS GERAIS

Numero de tramos            :1
Numero de nós               :5
Numero de elementos         :2

Calculo estático? (s/n)     :s

Calculo de frequencias
naturais? (s/n)             :s

Incluir efeito da inercia
rotacional? (s/n)           :s

Numero de freqüências       :2

Calculo
da carga critica? (s/n)     :s

Numero de modos             :2
```

Figura A.1 Arquivo de texto DADOS1.

```
PROGRAMA "ELEMENTOS FINITOS COM FUNÇÕES SPLINE"

 PROPRIEDADES DO MATERIAL
 Modulo de elasticidad em [KN/m^2]    :  0.21E+09
 Coeficiente de Poisson               :  0.3000
 Densidade especifica [KN*seg^2/m^4]  :  7.9500

 INFORMAÇÃO DOS NOS
 COORDENADAS SECAO  1
 x:   0.00
 No  1    y:   0.00              z:   0.00
 No  2    y:   0.00              z:   0.10
 No  3    y:   0.00              z:   0.20
 No  4    y:   0.00              z:   0.30
 No  5    y:   0.00              z:   0.40
 COORDENADAS SECAO  2
 x:   5.00
 No  1    y:   0.00              z:   0.00
 No  2    y:   0.00              z:   0.10
 No  3    y:   0.00              z:   0.20
 No  4    y:   0.00              z:   0.30
 No  5    y:   0.00              z:   0.40

 NUMERO DE DIVISOES POR TRAMO
 Para o calculo       Para os resultados
 Tramo  1:  4         Tramo  1:  4

 INFORMAÇÃO DOS ELEMENTOS

 Elemento  1   No inic.:  1   No inter:  2   No final:  3       Espes.:   0.025
 Elemento  2   No inic.:  3   No inter:  4   No final:  5       Espes.:   0.025


 CARREGAMENTO UNIFORME POR LINHA NODAL

 TRAMO  1
 No  1   qx:   0.00              qy:   0.00              qz:   0.00
 No  2   qx:   0.00              qy:   0.00              qz:   0.00
 No  3   qx:   0.00              qy:   0.00              qz:   0.00
 No  4   qx:   0.00              qy:   0.00              qz:   0.00
 No  5   qx:   0.00              qy:   0.00              qz:   0.00

 CARREGAMENTO PONTUAL NOS NÓS

 SECAO  1
 No  1   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  2   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  3   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  4   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  5   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00

 SECAO  2
 No  1   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  2   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  3   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  4   Fx: 0.00    Fy: 0.00    Fz:  0.00   Mx:  0.00   My:  0.00   Mz:  0.00
 No  5   Fx: 0.00    Fy: 0.00    Fz: 10.00   Mx:  0.00   My:  0.00   Mz:  0.00


 RESTRIÇÕES

 SEÇÃO  1
 No  1   Rx:  1      Ry:  1      Rz:  1      Mx:  1      My:  1      Mz:  1
 No  2   Rx:  1      Ry:  1      Rz:  1      Mx:  1      My:  1      Mz:  1
 No  3   Rx:  1      Ry:  1      Rz:  1      Mx:  1      My:  1      Mz:  1
 No  4   Rx:  1      Ry:  1      Rz:  1      Mx:  1      My:  1      Mz:  1
 No  5   Rx:  1      Ry:  1      Rz:  1      Mx:  1      My:  1      Mz:  1

 SEÇÃO  2
 No  1   Rx:  0      Ry:  0      Rz:  0      Mx:  0      My:  0      Mz:  0
 No  2   Rx:  0      Ry:  0      Rz:  0      Mx:  0      My:  0      Mz:  0
 No  3   Rx:  0      Ry:  0      Rz:  0      Mx:  0      My:  0      Mz:  0
 No  4   Rx:  0      Ry:  0      Rz:  0      Mx:  0      My:  0      Mz:  0
 No  5   Rx:  0      Ry:  0      Rz:  0      Mx:  0      My:  0      Mz:  0
```

Figura A.2 Arquivo de texto DADOS2.

```
Deslocamentos e Rotações
Nó N°   1
    x              Dx              Dy              Dz              Rx              Ry              Rz
    0.0000     0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
    1.2500     0.38681213E-03  0.00000000E+00  0.12758221E-02  0.00000000E+00 -0.19407311E-02  0.00000000E+00
    2.5000     0.66620888E-03  0.00000000E+00  0.46403785E-02  0.00000000E+00 -0.33433061E-02  0.00000000E+00
    3.7500     0.83326904E-03  0.00000000E+00  0.93970335E-02  0.00000000E+00 -0.41740764E-02  0.00000000E+00
    5.0000     0.88880185E-03  0.00000000E+00  0.14851074E-01  0.00000000E+00 -0.44484467E-02  0.00000000E+00
Nó N°   2
    x              Dx              Dy              Dz              Rx              Ry              Rz
    0.0000    -0.67762636E-20  0.00000000E+00 -0.27105054E-19  0.00000000E+00  0.00000000E+00  0.00000000E+00
    1.2500     0.19310570E-03  0.00000000E+00  0.12699057E-02  0.00000000E+00 -0.19399136E-02  0.00000000E+00
    2.5000     0.33260242E-03  0.00000000E+00  0.46360886E-02  0.00000000E+00 -0.33376816E-02  0.00000000E+00
    3.7500     0.41627878E-03  0.00000000E+00  0.93952296E-02  0.00000000E+00 -0.41724572E-02  0.00000000E+00
    5.0000     0.44417443E-03  0.00000000E+00  0.14851426E-01  0.00000000E+00 -0.44498595E-02  0.00000000E+00
Nó N°   3
    x              Dx              Dy              Dz              Rx              Ry              Rz
    0.0000     0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
    1.2500    -0.31036634E-07  0.00000000E+00  0.12679296E-02  0.00000000E+00 -0.19377107E-02  0.00000000E+00
    2.5000     0.63160388E-07  0.00000000E+00  0.46346573E-02  0.00000000E+00 -0.33322884E-02  0.00000000E+00
    3.7500    -0.69599861E-07  0.00000000E+00  0.93946240E-02  0.00000000E+00 -0.41697241E-02  0.00000000E+00
    5.0000    -0.10459666E-06  0.00000000E+00  0.14851958E-01  0.00000000E+00 -0.44497932E-02  0.00000000E+00
Nó N°   4
    x              Dx              Dy              Dz              Rx              Ry              Rz
    0.0000     0.00000000E+00  0.00000000E+00  0.27105054E-19  0.00000000E+00  0.00000000E+00  0.00000000E+00
    1.2500    -0.19315946E-03  0.00000000E+00  0.12699386E-02  0.00000000E+00 -0.19397633E-02  0.00000000E+00
    2.5000    -0.33248722E-03  0.00000000E+00  0.46360426E-02  0.00000000E+00 -0.33378799E-02  0.00000000E+00
    3.7500    -0.41640119E-03  0.00000000E+00  0.93952475E-02  0.00000000E+00 -0.41721077E-02  0.00000000E+00
    5.0000    -0.44463307E-03  0.00000000E+00  0.14852955E-01  0.00000000E+00 -0.44557606E-02  0.00000000E+00
Nó N°   5
    x              Dx              Dy              Dz              Rx              Ry              Rz
    0.0000    -0.13552527E-19  0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00
    1.2500    -0.38684287E-03  0.00000000E+00  0.12758901E-02  0.00000000E+00 -0.19404392E-02  0.00000000E+00
    2.5000    -0.66612413E-03  0.00000000E+00  0.46402841E-02  0.00000000E+00 -0.33436900E-02  0.00000000E+00
    3.7500    -0.83333937E-03  0.00000000E+00  0.93970721E-02  0.00000000E+00 -0.41733730E-02  0.00000000E+00
    5.0000    -0.89012743E-03  0.00000000E+00  0.14854183E-01  0.00000000E+00 -0.44608110E-02  0.00000000E+00
```

Figura A.3 Arquivo de texto RESEST.

```
Frequencias Naturais da Estrutura
  0.840988773800051        6.10692587848329
 MODO N°   1
 Nó N°   1
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00   -0.43368087E-18    0.00000000E+00
      1.2500      0.00000000E+00    0.21906722E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.77561756E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.15227937E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.23346430E+00    0.00000000E+00
 Nó N°   2
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.22012007E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.77610666E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.15230139E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.23345812E+00    0.00000000E+00
 Nó N°   3
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.22047507E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.77626591E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.15230900E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.23345641E+00    0.00000000E+00
 Nó N°   4
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.22012007E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.77610666E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.15230139E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.23345812E+00    0.00000000E+00
 Nó N°   5
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00   -0.86736174E-18    0.00000000E+00
      1.2500      0.00000000E+00    0.21906722E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.77561756E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.15227937E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.23346430E+00    0.00000000E+00
 MODO N°   2
 Nó N°   1
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00   -0.17347235E-17    0.00000000E+00
      1.2500      0.00000000E+00   -0.42031146E-01    0.00000000E+00
      2.5000      0.00000000E+00   -0.75932224E-01    0.00000000E+00
      3.7500      0.00000000E+00   -0.22018457E-01    0.00000000E+00
      5.0000      0.00000000E+00    0.12217808E+00    0.00000000E+00
 Nó N°   2
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00    0.17347235E-17    0.00000000E+00
      1.2500      0.00000000E+00   -0.42010691E-01    0.00000000E+00
      2.5000      0.00000000E+00   -0.75653286E-01    0.00000000E+00
      3.7500      0.00000000E+00   -0.21752124E-01    0.00000000E+00
      5.0000      0.00000000E+00    0.12226751E+00    0.00000000E+00
 Nó N°   3
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00    0.86736174E-18    0.00000000E+00
      1.2500      0.00000000E+00   -0.42004924E-01    0.00000000E+00
      2.5000      0.00000000E+00   -0.75559451E-01    0.00000000E+00
      3.7500      0.00000000E+00   -0.21664100E-01    0.00000000E+00
      5.0000      0.00000000E+00    0.12229710E+00    0.00000000E+00
 Nó N°   4
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00   -0.86736174E-18    0.00000000E+00
      1.2500      0.00000000E+00   -0.42010691E-01    0.00000000E+00
      2.5000      0.00000000E+00   -0.75653286E-01    0.00000000E+00
      3.7500      0.00000000E+00   -0.21752124E-01    0.00000000E+00
      5.0000      0.00000000E+00    0.12226751E+00    0.00000000E+00
 Nó N°   5
     x                Dx                Dy                Dz
      0.0000      0.00000000E+00    0.86736174E-18    0.00000000E+00
      1.2500      0.00000000E+00   -0.42031146E-01    0.00000000E+00
      2.5000      0.00000000E+00   -0.75932224E-01    0.00000000E+00
      3.7500      0.00000000E+00   -0.22018457E-01    0.00000000E+00
      5.0000      0.00000000E+00    0.12217808E+00    0.00000000E+00
```

Figura A.4 Arquivo de texto RESDIN.

```
Fatores de Cargas Criticas da Estrutura
  -2.43854774991853        2.63758106412435
 MODO N°   1
 Nó N°   1
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.34589146E-02    0.00000000E+00
      2.5000      0.00000000E+00    0.36521377E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.10688441E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.20350093E+00    0.00000000E+00
 Nó N°   2
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.61947221E-02    0.00000000E+00
      2.5000      0.00000000E+00    0.41524595E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.11252586E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.20937077E+00    0.00000000E+00
 Nó N°   3
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.88685884E-02    0.00000000E+00
      2.5000      0.00000000E+00    0.46464307E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.11811463E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.21523376E+00    0.00000000E+00
 Nó N°   4
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.11479898E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.51340147E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.12365060E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.22109092E+00    0.00000000E+00
 Nó N°   5
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.14029054E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.56150887E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.12913410E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.22694254E+00    0.00000000E+00
 MODO N°   2
 Nó N°   1
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00   -0.21684043E-18    0.00000000E+00
      1.2500      0.00000000E+00    0.14315144E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.57068186E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.13049084E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.22793273E+00    0.00000000E+00
 Nó N°   2
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00    0.00000000E+00    0.00000000E+00
      1.2500      0.00000000E+00    0.11862376E-01    0.00000000E+00
      2.5000      0.00000000E+00    0.52619777E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.12573934E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.22321532E+00    0.00000000E+00
 Nó N°   3
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00   -0.10842022E-18    0.00000000E+00
      1.2500      0.00000000E+00    0.93459842E-02    0.00000000E+00
      2.5000      0.00000000E+00    0.48105925E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.12093702E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.21849362E+00    0.00000000E+00
 Nó N°   4
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00   -0.81315163E-19    0.00000000E+00
      1.2500      0.00000000E+00    0.67655041E-02    0.00000000E+00
      2.5000      0.00000000E+00    0.43527953E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.11608353E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.21376756E+00    0.00000000E+00
 Nó N°   5
     x                Dx               Dy               Dz
      0.0000      0.00000000E+00   -0.54210109E-19    0.00000000E+00
      1.2500      0.00000000E+00    0.41215503E-02    0.00000000E+00
      2.5000      0.00000000E+00    0.38886333E-01    0.00000000E+00
      3.7500      0.00000000E+00    0.11117894E+00    0.00000000E+00
      5.0000      0.00000000E+00    0.20903659E+00    0.00000000E+00
```

Figura A.5 Arquivo de texto RESINS.

# B
# LISTAGEM DO PROGRAMA

Este apêndice apresenta o programa implementado em Fortran. As rotinas estão divididas em cinco grupos: programa principal, definições, rotinas gerais, entrada e saída de dados.

## B.1. Programa Principal

```
PROGRAM  Mindlin_Spline
   USE geral
   CALL Leitura_dados
   CALL Calculos
   IF (CALC_EST.EQ."s") THEN
      CALL Resul_est
   END IF
   IF (CALC_DIN.EQ."s") THEN
      CALL Resul_din
   END IF
   IF (CALC_INS.EQ."s") THEN
      CALL Resul_ins
   END IF
END PROGRAM Mindlin_Spline

SUBROUTINE Calculos
   USE IMSLF90
   USE lin_geig_gen_int
   USE geral
   USE calc
   INTEGER :: IAUX
   IAUX=6*NNOS*(3*NTRAMOS+SUM(MPT))
   ALLOCATE (RKG(IAUX,IAUX),RFG(IAUX),UG(IAUX))
   RKG=0.0D0;RFG=0.0D0;UG=0.0D0
   IF (CALC_DIN.EQ."s") THEN
      ALLOCATE (RMG(IAUX,IAUX),BETA(IAUX),ALPHA(IAUX),AUTOV(IAUX,IAUX) &
               ,RMODO(IAUX,NFREQ))
      RMG=0.0D0;BETA=0.0D0;ALPHA=0.0D0;AUTOV=0.0D0;RMODO=0.0D0
      IDIN=1
   END IF
   IF (CALC_INS.EQ."s") THEN
      ALLOCATE (RGG(IAUX,IAUX),BETAI(IAUX),ALPHAI(IAUX),AUTOVI(IAUX,IAUX) &
               ,RMODOI(IAUX,NMODOS))
      RGG=0.0D0;BETAI=0.0D0;ALPHAI=0.0D0;AUTOVI=0.0D0;RMODOI=0.0D0
   END IF
   DO I=1,NTRAMOS
      DO J=1,NELEM
         CALL Coord_locais(I,J)
         CALL Matriz_Elem(I,J)
         CALL Ensamblar(I,J)
      END DO
   END DO
   CALL CargaUniforme
   CALL CargaPontual
   CALL DeslocFisicos(IAUX)
   CALL Restricoes(IAUX)
   IF (CALC_DIN.EQ."s") THEN
      CALL lin_geig_gen(RKG, RMG, ALPHA, BETA, v=AUTOV)
      RMODO=REAL(AUTOV(1:IAUX,1:NFREQ))
      CALL RMODOparametros
   END IF
   IF (CALC_INS.EQ."s") THEN
```

```
      ALLOCATE(RKGI(IAUX,IAUX))
      RKGI=0.0D0
      RKGI=RKG
   END IF
   IF (CALC_EST.EQ."s" .OR. CALC_INS.EQ."s") THEN
      CALL Sol(RKG,RFG,UG,IAUX)
      CALL UGparametros
   END IF
   IF (CALC_INS.EQ."s") THEN
     DO I=1,NTRAMOS
      DO J=1,NELEM
         CALL Param_elem(I,J)
         CALL Coord_locais(I,J)
         CALL Transf_local(MPT(I))
         CALL Matriz_Geom(I,J)
         CALL Ensam_geom(I,J)
      END DO
     END DO
     CALL DeslFis_geom(IAUX)
     CALL Restric_geom(IAUX)
     CALL lin_geig_gen(RKGI, RGG, ALPHAI, BETAI, v=AUTOVI)
     RMODOI=REAL(AUTOVI(1:IAUX,1:NMODOS))
     CALL RMODOIparam_geom
   END IF
END SUBROUTINE Calculos

SUBROUTINE Coord_locais(NTR,NEL)               !NTR=# de tramo; NEL=# de elemento
   USE geral
   USE calc
   REAL(KIND(0.0D0)), DIMENSION(3) ::P1,P2,P3,P4,V1,V2,V3
   REAL(KIND(0.0D0)) :: C1,C2,C3
   P1=COORD(NTR,INCID(NEL,1),:)
   P2=COORD(NTR+1,INCID(NEL,1),:)
   P3=COORD(NTR+1,INCID(NEL,3),:)
   P4=COORD(NTR,INCID(NEL,3),:)
   V1=P2-P1
   VLY=P4-P1
   VLZ(1)=V1(2)*VLY(3)-V1(3)*VLY(2)
   VLZ(2)=-V1(1)*VLY(3)+V1(3)*VLY(1)
   VLZ(3)=V1(1)*VLY(2)-V1(2)*VLY(1)
   VLX(1)=P2(1)-P1(1)
   C1=VLZ(2)*VLY(3)-VLZ(3)*VLY(2)
   C2=VLZ(3)*VLY(2)-VLZ(2)*VLY(3)
   VLX(2)=((VLZ(1)*VLY(3)-VLZ(3)*VLY(1))*VLX(1)-P1(2)*C1)/C2-P1(2)
   VLX(3)=((VLZ(2)*VLY(1)-VLZ(1)*VLY(2))*VLX(1)-P1(3)*C1)/C2-P1(3)
   V2=V1-VLX
   V3=P3-P2
   CALL ModuloVetor(VLX,C3)
   LONG=C3
   DO I=1,3
      COSDIR(1,I)=VLX(I)/C3
   END DO
   CALL ModuloVetor(V2,C3)
   Y(2)=C3
   CALL ModuloVetor(V3,C3)
   Y(3)=Y(2)+C3
   CALL ModuloVetor(VLY,C3)
   Y(4)=C3
   DO I=1,3
      COSDIR(2,I)=VLY(I)/C3
   END DO
   CALL ModuloVetor(VLZ,C3)
   DO I=1,3
      COSDIR(3,I)=VLZ(I)/C3
   END DO
END SUBROUTINE Coord_locais

SUBROUTINE Matriz_Elem(NTR,NEL)                !NTR=# de tramo; NEL=# de elemento
   USE geral
   USE calc
   USE pmatrizloc
   REAL(KIND(0.0D0)), DIMENSION(5) :: FX1,W1,FX2,W2             !p/Quad. de Gauss
   REAL(KIND(0.0D0)) :: AKA(18,18),AMA(18,18),Z,RN,RV,XIX,CESP
   INTEGER :: INROT
   IF (E1.EQ.0) THEN
      E1=ELAST/(1.-POISS**2)
      E2=ELAST*POISS/(1.-POISS**2)
```

```
   E3=ELAST/(1.+POISS)/2.
   GZ=E3
   RKO=0.25D0                                          !RKO=Constante drilling
END IF
CESP=ESPES(NEL)
M=MPT(NTR)
EP1=E1*CESP**3/12.
EP2=E2*CESP**3/12.
EP3=E3*CESP**3/12.
ALLOCATE (RKEL(18*(M+3),18*(M+3)),RKEG(18*(M+3),18*(M+3)))
RKEL=0.0D0;RKEG=0.0D0
IF (IDIN.EQ.1) THEN
   ALLOCATE (RMEL(18*(M+3),18*(M+3)),RMEG(18*(M+3),18*(M+3)))
   RMEL=0.0D0;RMEG=0.0D0
END IF
NN=4                                                   !Pontos de Gauss em x
NE=4                                                   !Pontos de Gauss em y
CALL Qgauss(NN,NE,FX1,W1,FX2,W2)
DO II=1,(M+3)
  DO JJ=II,(M+3)
   I=II-2
   J=JJ-2
   NT=0
   IF (I.EQ.-1 .AND. J.EQ.-1 .OR. I.EQ.-1 .AND. J.EQ.0 .OR. I.EQ.&
      -1 .AND. J.EQ.1 .OR. I.EQ.M+1 .AND. J.EQ.M+1 .OR. I.EQ.M .AND.&
      J.EQ.M+1 .OR. I.EQ.M-1 .AND. J.EQ.M+1) NT=1
   IF (I.EQ.0 .AND. J.EQ.0 .OR. I.EQ.0 .AND. J.EQ.1 .OR. I.EQ.M &
      .AND. J.EQ.M .OR. I.EQ.M-1 .AND. J.EQ.M) NT=2
   IF (I.EQ.1 .AND. J.EQ.1 .OR. I.EQ.M-1 .AND. J.EQ.M-1) NT=3
   IF (J-I.LT.4 .AND. NT.EQ.0) NT=I-J+4
   IF (NT.NE.0) THEN
     AKA=0.0D0
     AMA=0.0D0
     DO IQ=1,NT
      DO IG=1,NN
        Z=FX1(IG)
        CALL Pinic(I,IQ,NT,M,IP)
        RN=(Z+1.)/2.+REAL(IP,KIND(0.0D0))
        CALL Detfun(I,J,IQ,NT,RN,LONG,M,2)
        RV=RN*(-Y(2)+Y(3)-Y(4))/REAL(M,KIND(0.0D0))+Y(4)
        DO JG=1,NE
         XI=FX2(JG)
         XIX=((1.-XI)*(-Y(2))-(1.+XI)*(Y(3)-Y(4)))/LONG/RV
         R1=0.5*XI*(XI-1.)
         R2=1.-XI**2
         R3=0.5*XI*(XI+1.)
         S1Y=(XI-0.5)*2./RV
         S2Y=-4.*XI/RV
         S3Y=(XI+0.5)*2./RV
         S1X=(XI-0.5)*XIX
         S2X=-2.*XI*XIX
         S3X=(XI+0.5)*XIX
         CALL MakeKA(CESP)
         IF (IDIN.EQ.1) THEN
            IF (INER_ROT.EQ."s") THEN
               INROT=1
            END IF
            CALL MakeMA(CESP,RHO,INROT)
         END IF
         DO IS=1,18
           DO JS=1,18
            AKA(IS,JS)=AKA(IS,JS)+W1(IG)*W2(JG)*RKA(IS,JS)*RV
            IF (IDIN.EQ.1) THEN
               AMA(IS,JS)=AMA(IS,JS)+W1(IG)*W2(JG)*RMA(IS,JS)*RV
            END IF
           END DO
         END DO
        END DO
      END DO
     END DO
     DO IA=1,18
      DO IB=1,18
        RKEL(18*I+18+IA,18*J+18+IB)=LONG*AKA(IA,IB)/4./REAL(M,KIND(0.0D0))
        RKEL(18*J+18+IB,18*I+18+IA)=RKEL(18*I+18+IA,18*J+18+IB)
        IF (IDIN.EQ.1) THEN
         RMEL(18*I+18+IA,18*J+18+IB)=LONG*AMA(IA,IB)/4./REAL(M,KIND(0.0D0))
         RMEL(18*J+18+IB,18*I+18+IA)=RMEL(18*I+18+IA,18*J+18+IB)
```

```
          END IF
        END DO
      END DO
    END IF
  END DO
  END DO
  CALL Matriz_global(M)
END SUBROUTINE Matriz_Elem

!ROTINA PARA DETERMINAR O PONTO INICIAL DE INTEGRAÇÃO
SUBROUTINE Pinic(I,IQ,NT,M,IP)
  INTEGER, INTENT(IN) :: I,IQ,NT,M
  INTEGER, INTENT(OUT) :: IP
  IF (I.GE.-1 .AND. I.LE.M-2) THEN
    IP=1-NT+IQ+I
  ELSE IF (I.EQ.M-1 .OR. I.EQ.M) THEN
    IP=M-1-NT+IQ
  ELSE IF (I.EQ.M+1) THEN
    IP=M-1
  END IF
END SUBROUTINE Pinic

!ROTINA PARA DETERMINAR AS FUNÇÕES SPLINE, JJ=2 PARA CALCULAR PI E PJ, JJ=1 SÓ PI
SUBROUTINE Detfun(I,J,IQ,NT,RN,RL,M,JJ)
  USE pmatrizloc
  INTEGER, INTENT(IN) :: I,J,IQ,NT,M,JJ
  REAL(KIND(0.0D0)), INTENT(IN) :: RN,RL
  REAL(KIND(0.0D0)) :: RM,RK,PP,QQ,QQ2
  RM=REAL(M,KIND(0.0D0))
  DO KK=1,JJ
    IF (KK.EQ.1)THEN
      RK=REAL(I,KIND(0.0D0))
      IF (I.GE.-1 .AND. I.LE.M-2) THEN
        IC=4-NT+IQ
      ELSE IF (I.EQ.M-1) THEN
        IC=3-NT+IQ
      ELSE IF (I.EQ.M) THEN
        IC=2-NT+IQ
      ELSE IF (I.EQ.M+1) THEN
        IC=1
      END IF
    END IF
    IF (KK.EQ.2)THEN
      RK=REAL(J,KIND(0.0D0))
      IF (J.EQ.-1) THEN
        IC=4
      ELSE IF (J.EQ.0) THEN
        IC=2+IQ
      ELSE IF (J.EQ.1) THEN
        IC=1+IQ
      ELSE IF (J.GE.2 .AND. J.LE.M+1) THEN
        IC=IQ
      END IF
    END IF
    SELECT CASE (IC)
      CASE (1)
        PP=(RN-(RK-2.))**3/6.
        QQ=((RN-RK+2.)**2/2.)*RM/RL
      CASE (2)
        PP=(1.+3.*(RN-(RK-1.))+3.*(RN-(RK-1.))**2-3.*(RN-(RK-1.))**3)/6.
        QQ=(3./2.+RN-RK-3.*(RN-RK+1.)**2/2.)*RM/RL
      CASE (3)
        PP=(1.+3.*((RK+1.)-RN)+3.*((RK+1.)-RN)**2-3.*((RK+1.)-RN)**3)/6.
        QQ=(-3./2.-RK+RN+3.*(RK+1-RN)**2/2.)*RM/RL
      CASE (4)
        PP=(((RK+2.)-RN)**3)/6.
        QQ=(-(RK+2.-RN)**2/2.)*RM/RL
    END SELECT
    IF (KK.EQ.1)THEN
      FI=PP
      QI=QQ
    ELSE IF (KK.EQ.2)THEN
      FJ=PP
      QJ=QQ
    END IF
  END DO
END SUBROUTINE Detfun
```

```
!ROTINA PARA CALCULAR SUBMATRIZ KA 18x18
SUBROUTINE MakeKA(CESP)
   USE pmatrizloc
   REAL(KIND(0.0D0)), INTENT(IN) :: CESP
   RKA(1,1)=CESP*((R1*QI+S1X*FI)*E1*(R1*QJ+S1X*FJ)+S1Y**2*FI*E3*FJ)+RKO*GZ*CESP &
            *S1Y**2*FI*FJ/0.4D1
   RKA(1,2)=CESP*((R1*QI+S1X*FI)*E2*S1Y*FJ+S1Y*FI*E3*(R1*QJ+S1X*FJ))+RKO*GZ*CESP &
            *S1Y*FI*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)/0.2D1
   RKA(1,6)=RKO*GZ*CESP*S1Y*FI*R1*FJ/0.2D1
   RKA(1,7)=CESP*((R1*QI+S1X*FI)*E1*(R2*QJ+S2X*FJ)+S1Y*FI*E3*S2Y*FJ)+RKO*GZ*CESP &
            *S1Y*FI*S2Y*FJ/0.4D1
   RKA(1,8)=CESP*((R1*QI+S1X*FI)*E2*S2Y*FJ+S1Y*FI*E3*(R2*QJ+S2X*FJ))+RKO*GZ*CESP &
            *S1Y*FI*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)/0.2D1
   RKA(1,12)=RKO*GZ*CESP*S1Y*FI*R2*FJ/0.2D1
   RKA(1,13)=CESP*((R1*QI+S1X*FI)*E1*(R3*QJ+S3X*FJ)+S1Y*FI*E3*S3Y*FJ)+RKO*GZ &
             *CESP *S1Y*FI*S3Y*FJ/0.4D1
   RKA(1,14)=CESP*((R1*QI+S1X*FI)*E2*S3Y*FJ+S1Y*FI*E3*(R3*QJ+S3X*FJ))+RKO*GZ &
             *CESP*S1Y*FI*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)/0.2D1
   RKA(1,18)=RKO*GZ*CESP*S1Y*FI*R3*FJ/0.2D1
   RKA(2,1)=CESP*(S1Y*FI*E2*(R1*QJ+S1X*FJ)+(R1*QI+S1X*FI)*E3*S1Y*FJ)+RKO*GZ*CESP &
            *(-R1*QI/0.2D1-S1X*FI/0.2D1)*S1Y*FJ/0.2D1
   RKA(2,2)=CESP*(S1Y**2*FI*E1*FJ+(R1*QI+S1X*FI)*E3*(R1*QJ+S1X*FJ))+RKO*GZ*CESP &
            *(-R1*QI/0.2D1-S1X*FI/0.2D1)*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)
   RKA(2,6)=RKO*GZ*CESP*(-R1*QI/0.2D1-S1X*FI/0.2D1)*R1*FJ
   RKA(2,7)=CESP*(S1Y*FI*E2*(R2*QJ+S2X*FJ)+(R1*QI+S1X*FI)*E3*S2Y*FJ)+RKO*GZ*CESP &
            *(-R1*QI/0.2D1-S1X*FI/0.2D1)*S2Y*FJ/0.2D1
   RKA(2,8)=CESP*(S1Y*FI*E1*S2Y*FJ+(R1*QI+S1X*FI)*E3*(R2*QJ+S2X*FJ))+RKO*GZ*CESP &
            *(-R1*QI/0.2D1-S1X*FI/0.2D1)*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)
   RKA(2,12)=RKO*GZ*CESP*(-R1*QI/0.2D1-S1X*FI/0.2D1)*R2*FJ
   RKA(2,13)=CESP*(S1Y*FI*E2*(R3*QJ+S3X*FJ)+(R1*QI+S1X*FI)*E3*S3Y*FJ)+RKO*GZ &
             *CESP*(-R1*QI/0.2D1-S1X*FI/0.2D1)*S3Y*FJ/0.2D1
   RKA(2,14)=CESP*(S1Y*FI*E1*S3Y*FJ+(R1*QI+S1X*FI)*E3*(R3*QJ+S3X*FJ))+RKO*GZ &
             *CESP*(-R1*QI/0.2D1-S1X*FI/0.2D1)*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)
   RKA(2,18)=RKO*GZ*CESP*(-R1*QI/0.2D1-S1X*FI/0.2D1)*R3*FJ
   RKA(3,3)=((-R1*QI-S1X*FI)*(-R1*QJ-S1X*FJ)+S1Y**2*FI*FJ)*GZ*CESP/1.2D0
   RKA(3,4)=-S1Y*FI*R1*FJ*GZ*CESP/1.2D0
   RKA(3,5)=-(-R1*QI-S1X*FI)*R1*FJ*GZ*CESP/1.2D0
   RKA(3,9)=((-R1*QI-S1X*FI)*(-R2*QJ-S2X*FJ)+S1Y*FI*S2Y*FJ)*GZ*CESP/1.2D0
   RKA(3,10)=-S1Y*FI*R2*FJ*GZ*CESP/1.2D0
   RKA(3,11)=-(-R1*QI-S1X*FI)*R2*FJ*GZ*CESP/1.2D0
   RKA(3,15)=((-R1*QI-S1X*FI)*(-R3*QJ-S3X*FJ)+S1Y*FI*S3Y*FJ)*GZ*CESP/1.2D0
   RKA(3,16)=-S1Y*FI*R3*FJ*GZ*CESP/1.2D0
   RKA(3,17)=-(-R1*QI-S1X*FI)*R3*FJ*GZ*CESP/1.2D0
   RKA(4,3)=-S1Y*FI*R1*FJ*GZ*CESP/1.2D0
   RKA(4,4)=S1Y**2*FI*EP1*FJ+(-R1*QI-S1X*FI)*EP3*(-R1*QJ-S1X*FJ)+R1**2*FI*FJ*GZ &
            *CESP/1.2D0
   RKA(4,5)=-S1Y*FI*EP2*(R1*QJ+S1X*FJ)+(-R1*QI-S1X*FI)*EP3*S1Y*FJ
   RKA(4,9)=-R1*FI*S2Y*FJ*GZ*CESP/1.2D0
   RKA(4,10)=S1Y*FI*EP1*S2Y*FJ+(-R1*QI-S1X*FI)*EP3*(-R2*QJ-S2X*FJ)+R1*FI*R2*FJ &
             *GZ*CESP/1.2D0
   RKA(4,11)=-S1Y*FI*EP2*(R2*QJ+S2X*FJ)+(-R1*QI-S1X*FI)*EP3*S2Y*FJ
   RKA(4,15)=-R1*FI*S3Y*FJ*GZ*CESP/1.2D0
   RKA(4,16)=S1Y*FI*EP1*S3Y*FJ+(-R1*QI-S1X*FI)*EP3*(-R3*QJ-S3X*FJ)+R1*FI*R3*FJ &
             *GZ*CESP/1.2D0
   RKA(4,17)=-S1Y*FI*EP2*(R3*QJ+S3X*FJ)+(-R1*QI-S1X*FI)*EP3*S3Y*FJ
   RKA(5,3)=-R1*FI*(-R1*QJ-S1X*FJ)*GZ*CESP/1.2D0
   RKA(5,4)=-(R1*QI+S1X*FI)*EP2*S1Y*FJ+S1Y*FI*EP3*(-R1*QJ-S1X*FJ)
   RKA(5,5)=(R1*QI+S1X*FI)*EP1*(R1*QJ+S1X*FJ)+S1Y**2*FI*EP3*FJ+R1**2*FI*FJ*GZ &
            *CESP/1.2D0
   RKA(5,9)=-R1*FI*(-R2*QJ-S2X*FJ)*GZ*CESP/1.2D0
   RKA(5,10)=-(R1*QI+S1X*FI)*EP2*S2Y*FJ+S1Y*FI*EP3*(-R2*QJ-S2X*FJ)
   RKA(5,11)=(R1*QI+S1X*FI)*EP1*(R2*QJ+S2X*FJ)+S1Y*FI*EP3*S2Y*FJ+R1*FI*R2*FJ*GZ &
             *CESP/1.2D0
   RKA(5,15)=-R1*FI*(-R3*QJ-S3X*FJ)*GZ*CESP/1.2D0
   RKA(5,16)=-(R1*QI+S1X*FI)*EP2*S3Y*FJ+S1Y*FI*EP3*(-R3*QJ-S3X*FJ)
   RKA(5,17)=(R1*QI+S1X*FI)*EP1*(R3*QJ+S3X*FJ)+S1Y*FI*EP3*S3Y*FJ+R1*FI*R3*FJ*GZ &
             *CESP/1.2D0
   RKA(6,1)=RKO*GZ*CESP*S1Y*FI*R1*FJ/0.2D1
   RKA(6,2)=RKO*GZ*CESP*R1*FI*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)
   RKA(6,6)=RKO*GZ*CESP*R1**2*FI*FJ
   RKA(6,7)=RKO*GZ*CESP*R1*FI*S2Y*FJ/0.2D1
   RKA(6,8)=RKO*GZ*CESP*R1*FI*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)
   RKA(6,12)=RKO*GZ*CESP*R1*FI*R2*FJ
   RKA(6,13)=RKO*GZ*CESP*R1*FI*S3Y*FJ/0.2D1
   RKA(6,14)=RKO*GZ*CESP*R1*FI*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)
   RKA(6,18)=RKO*GZ*CESP*R1*FI*R3*FJ
```

```
RKA(7,1)=CESP*((R2*QI+S2X*FI)*E1*(R1*QJ+S1X*FJ)+S1Y*FI*E3*S2Y*FJ)+RKO*GZ*CESP &
         *S1Y*FI*S2Y*FJ/0.4D1
RKA(7,2)=CESP*((R2*QI+S2X*FI)*E2*S1Y*FJ+S2Y*FI*E3*(R1*QJ+S1X*FJ))+RKO*GZ*CESP &
         *S2Y*FI*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)/0.2D1
RKA(7,6)=RKO*GZ*CESP*R1*FI*S2Y*FJ/0.2D1
RKA(7,7)=CESP*((R2*QI+S2X*FI)*E1*(R2*QJ+S2X*FJ)+S2Y**2*FI*E3*FJ)+RKO*GZ*CESP &
         *S2Y**2*FI*FJ/0.4D1
RKA(7,8)=CESP*((R2*QI+S2X*FI)*E2*S2Y*FJ+S2Y*FI*E3*(R2*QJ+S2X*FJ))+RKO*GZ*CESP &
         *S2Y*FI*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)/0.2D1
RKA(7,12)=RKO*GZ*CESP*S2Y*FI*R2*FJ/0.2D1
RKA(7,13)=CESP*((R2*QI+S2X*FI)*E1*(R3*QJ+S3X*FJ)+S2Y*FI*E3*S3Y*FJ)+RKO*GZ &
          *CESP*S2Y*FI*S3Y*FJ/0.4D1
RKA(7,14)=CESP*((R2*QI+S2X*FI)*E2*S3Y*FJ+S2Y*FI*E3*(R3*QJ+S3X*FJ))+RKO*GZ &
          *CESP*S2Y*FI*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)/0.2D1
RKA(7,18)=RKO*GZ*CESP*S2Y*FI*R3*FJ/0.2D1
RKA(8,1)=CESP*(S2Y*FI*E2*(R1*QJ+S1X*FJ)+(R2*QI+S2X*FI)*E3*S1Y*FJ)+RKO*GZ*CESP &
         *(-R2*QI/0.2D1-S2X*FI/0.2D1)*S1Y*FJ/0.2D1
RKA(8,2)=CESP*(S1Y*FI*E1*S2Y*FJ+(R2*QI+S2X*FI)*E3*(R1*QJ+S1X*FJ))+RKO*GZ*CESP &
         *(-R2*QI/0.2D1-S2X*FI/0.2D1)*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)
RKA(8,6)=RKO*GZ*CESP*(-R2*QI/0.2D1-S2X*FI/0.2D1)*R1*FJ
RKA(8,7)=CESP*(S2Y*FI*E2*(R2*QJ+S2X*FJ)+(R2*QI+S2X*FI)*E3*S2Y*FJ)+RKO*GZ*CESP &
         *(-R2*QI/0.2D1-S2X*FI/0.2D1)*S2Y*FJ/0.2D1
RKA(8,8)=CESP*(S2Y**2*FI*E1*FJ+(R2*QI+S2X*FI)*E3*(R2*QJ+S2X*FJ))+RKO*GZ*CESP &
         *(-R2*QI/0.2D1-S2X*FI/0.2D1)*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)
RKA(8,12)=RKO*GZ*CESP*(-R2*QI/0.2D1-S2X*FI/0.2D1)*R2*FJ
RKA(8,13)=CESP*(S2Y*FI*E2*(R3*QJ+S3X*FJ)+(R2*QI+S2X*FI)*E3*S3Y*FJ)+RKO*GZ &
          *CESP*(-R2*QI/0.2D1-S2X*FI/0.2D1)*S3Y*FJ/0.2D1
RKA(8,14)=CESP*(S2Y*FI*E1*S3Y*FJ+(R2*QI+S2X*FI)*E3*(R3*QJ+S3X*FJ))+RKO*GZ &
          *CESP*(-R2*QI/0.2D1-S2X*FI/0.2D1)*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)
RKA(8,18)=RKO*GZ*CESP*(-R2*QI/0.2D1-S2X*FI/0.2D1)*R3*FJ
RKA(9,3)=((-R2*QI-S2X*FI)*(-R1*QJ-S1X*FJ)+S1Y*FI*S2Y*FJ)*GZ*CESP/1.2D0
RKA(9,4)=-R1*FI*S2Y*FJ*GZ*CESP/1.2D0
RKA(9,5)=-(-R2*QI-S2X*FI)*R1*FJ*GZ*CESP/1.2D0
RKA(9,9)=((-R2*QI-S2X*FI)*(-R2*QJ-S2X*FJ)+S2Y**2*FI*FJ)*GZ*CESP/1.2D0
RKA(9,10)=-S2Y*FI*R2*FJ*GZ*CESP/1.2D0
RKA(9,11)=-(-R2*QI-S2X*FI)*R2*FJ*GZ*CESP/1.2D0
RKA(9,15)=((-R2*QI-S2X*FI)*(-R3*QJ-S3X*FJ)+S2Y*FI*S3Y*FJ)*GZ*CESP/1.2D0
RKA(9,16)=-S2Y*FI*R3*FJ*GZ*CESP/1.2D0
RKA(9,17)=-(-R2*QI-S2X*FI)*R3*FJ*GZ*CESP/1.2D0
RKA(10,3)=-S1Y*FI*R2*FJ*GZ*CESP/1.2D0
RKA(10,4)=S1Y*FI*EP1*S2Y*FJ+(-R2*QI-S2X*FI)*EP3*(-R1*QJ-S1X*FJ)+R1*FI*R2*FJ &
          *GZ*CESP/1.2D0
RKA(10,5)=-S2Y*FI*EP2*(R1*QJ+S1X*FJ)+(-R2*QI-S2X*FI)*EP3*S1Y*FJ
RKA(10,9)=-S2Y*FI*R2*FJ*GZ*CESP/1.2D0
RKA(10,10)=S2Y**2*FI*EP1*FJ+(-R2*QI-S2X*FI)*EP3*(-R2*QJ-S2X*FJ)+R2**2*FI &
           *FJ*GZ*CESP/1.2D0
RKA(10,11)=-S2Y*FI*EP2*(R2*QJ+S2X*FJ)+(-R2*QI-S2X*FI)*EP3*S2Y*FJ
RKA(10,15)=-R2*FI*S3Y*FJ*GZ*CESP/1.2D0
RKA(10,16)=S2Y*FI*EP1*S3Y*FJ+(-R2*QI-S2X*FI)*EP3*(-R3*QJ-S3X*FJ)+R2*FI*R3 &
           *FJ*GZ*CESP/1.2D0
RKA(10,17)=-S2Y*FI*EP2*(R3*QJ+S3X*FJ)+(-R2*QI-S2X*FI)*EP3*S3Y*FJ
RKA(11,3)=-R2*FI*(-R1*QJ-S1X*FJ)*GZ*CESP/1.2D0
RKA(11,4)=-(R2*QI+S2X*FI)*EP2*S1Y*FJ+S2Y*FI*EP3*(-R1*QJ-S1X*FJ)
RKA(11,5)=(R2*QI+S2X*FI)*EP1*(R1*QJ+S1X*FJ)+S1Y*FI*EP3*S2Y*FJ+R1*FI*R2*FJ*GZ &
          *CESP/1.2D0
RKA(11,9)=-R2*FI*(-R2*QJ-S2X*FJ)*GZ*CESP/1.2D0
RKA(11,10)=-(R2*QI+S2X*FI)*EP2*S2Y*FJ+S2Y*FI*EP3*(-R2*QJ-S2X*FJ)
RKA(11,11)=(R2*QI+S2X*FI)*EP1*(R2*QJ+S2X*FJ)+S2Y**2*FI*EP3*FJ+R2**2*FI*FJ*GZ &
           *CESP/1.2D0
RKA(11,15)=-R2*FI*(-R3*QJ-S3X*FJ)*GZ*CESP/1.2D0
RKA(11,16)=-(R2*QI+S2X*FI)*EP2*S3Y*FJ+S2Y*FI*EP3*(-R3*QJ-S3X*FJ)
RKA(11,17)=(R2*QI+S2X*FI)*EP1*(R3*QJ+S3X*FJ)+S2Y*FI*EP3*S3Y*FJ+R2*FI*R3*FJ*GZ &
           *CESP/1.2D0
RKA(12,1)=RKO*GZ*CESP*S1Y*FI*R2*FJ/0.2D1
RKA(12,2)=RKO*GZ*CESP*R2*FI*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)
RKA(12,6)=RKO*GZ*CESP*R1*FI*R2*FJ
RKA(12,7)=RKO*GZ*CESP*S2Y*FI*R2*FJ/0.2D1
RKA(12,8)=RKO*GZ*CESP*R2*FI*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)
RKA(12,12)=RKO*GZ*CESP*R2**2*FI*FJ
RKA(12,13)=RKO*GZ*CESP*R2*FI*S3Y*FJ/0.2D1
RKA(12,14)=RKO*GZ*CESP*R2*FI*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)
RKA(12,18)=RKO*GZ*CESP*R2*FI*R3*FJ
RKA(13,1)=CESP*((R3*QI+S3X*FI)*E1*(R1*QJ+S1X*FJ)+S1Y*FI*E3*S3Y*FJ)+RKO*GZ &
          *CESP*S1Y*FI*S3Y*FJ/0.4D1
RKA(13,2)=CESP*((R3*QI+S3X*FI)*E2*S1Y*FJ+S3Y*FI*E3*(R1*QJ+S1X*FJ))+RKO*GZ &
          *CESP*S3Y*FI*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)/0.2D1
RKA(13,6)=RKO*GZ*CESP*R1*FI*S3Y*FJ/0.2D1
```

```
   RKA(13,7)=CESP*((R3*QI+S3X*FI)*E1*(R2*QJ+S2X*FJ)+S2Y*FI*E3*S3Y*FJ)+RKO*GZ &
             *CESP*S2Y*FI*S3Y*FJ/0.4D1
   RKA(13,8)=CESP*((R3*QI+S3X*FI)*E2*S2Y*FJ+S3Y*FI*E3*(R2*QJ+S2X*FJ))+RKO*GZ &
             *CESP*S3Y*FI*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)/0.2D1
   RKA(13,12)=RKO*GZ*CESP*R2*FI*S3Y*FJ/0.2D1
   RKA(13,13)=CESP*((R3*QI+S3X*FI)*E1*(R3*QJ+S3X*FJ)+S3Y**2*FI*E3*FJ)+RKO*GZ &
              *CESP*S3Y**2*FI*FJ/0.4D1
   RKA(13,14)=CESP*((R3*QI+S3X*FI)*E2*S3Y*FJ+S3Y*FI*E3*(R3*QJ+S3X*FJ))+RKO*GZ &
              *CESP*S3Y*FI*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)/0.2D1
   RKA(13,18)=RKO*GZ*CESP*S3Y*FI*R3*FJ/0.2D1
   RKA(14,1)=CESP*(S3Y*FI*E2*(R1*QJ+S1X*FJ)+(R3*QI+S3X*FI)*E3*S1Y*FJ)+RKO*GZ &
             *CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*S1Y*FJ/0.2D1
   RKA(14,2)=CESP*(S1Y*FI*E1*S3Y*FJ+(R3*QI+S3X*FI)*E3*(R1*QJ+S1X*FJ))+RKO*GZ &
             *CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)
   RKA(14,6)=RKO*GZ*CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*R1*FJ
   RKA(14,7)=CESP*(S3Y*FI*E2*(R2*QJ+S2X*FJ)+(R3*QI+S3X*FI)*E3*S2Y*FJ)+RKO*GZ &
             *CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*S2Y*FJ/0.2D1
   RKA(14,8)=CESP*(S2Y*FI*E1*S3Y*FJ+(R3*QI+S3X*FI)*E3*(R2*QJ+S2X*FJ))+RKO*GZ &
             *CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)
   RKA(14,12)=RKO*GZ*CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*R2*FJ
   RKA(14,13)=CESP*(S3Y*FI*E2*(R3*QJ+S3X*FJ)+(R3*QI+S3X*FI)*E3*S3Y*FJ)+RKO*GZ &
              *CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*S3Y*FJ/0.2D1
   RKA(14,14)=CESP*(S3Y**2*FI*E1*FJ+(R3*QI+S3X*FI)*E3*(R3*QJ+S3X*FJ))+RKO*GZ &
              *CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)
   RKA(14,18)=RKO*GZ*CESP*(-R3*QI/0.2D1-S3X*FI/0.2D1)*R3*FJ
   RKA(15,3)=((-R3*QI-S3X*FI)*(-R1*QJ-S1X*FJ)+S1Y*FI*S3Y*FJ)*GZ*CESP/1.2D0
   RKA(15,4)=-R1*FI*S3Y*FJ*GZ*CESP/1.2D0
   RKA(15,5)=-(-R3*QI-S3X*FI)*R1*FJ*GZ*CESP/1.2D0
   RKA(15,9)=((-R3*QI-S3X*FI)*(-R2*QJ-S2X*FJ)+S2Y*FI*S3Y*FJ)*GZ*CESP/1.2D0
   RKA(15,10)=-R2*FI*S3Y*FJ*GZ*CESP/1.2D0
   RKA(15,11)=-(-R3*QI-S3X*FI)*R2*FJ*GZ*CESP/1.2D0
   RKA(15,15)=((-R3*QI-S3X*FI)*(-R3*QJ-S3X*FJ)+S3Y**2*FI*FJ)*GZ*CESP/1.2D0
   RKA(15,16)=-S3Y*FI*R3*FJ*GZ*CESP/1.2D0
   RKA(15,17)=-(-R3*QI-S3X*FI)*R3*FJ*GZ*CESP/1.2D0
   RKA(16,3)=-S1Y*FI*R3*FJ*GZ*CESP/1.2D0
   RKA(16,4)=S1Y*FI*EP1*S3Y*FJ+(-R3*QI-S3X*FI)*EP3*(-R1*QJ-S1X*FJ)+R1*FI*R3*FJ &
             *GZ*CESP/1.2D0
   RKA(16,5)=-S3Y*FI*EP2*(R1*QJ+S1X*FJ)+(-R3*QI-S3X*FI)*EP3*S1Y*FJ
   RKA(16,9)=-S2Y*FI*R3*FJ*GZ*CESP/1.2D0
   RKA(16,10)=S2Y*FI*EP1*S3Y*FJ+(-R3*QI-S3X*FI)*EP3*(-R2*QJ-S2X*FJ)+R2*FI*R3*FJ &
              *GZ*CESP/1.2D0
   RKA(16,11)=-S3Y*FI*EP2*(R2*QJ+S2X*FJ)+(-R3*QI-S3X*FI)*EP3*S2Y*FJ
   RKA(16,15)=-S3Y*FI*R3*FJ*GZ*CESP/1.2D0
   RKA(16,16)=S3Y**2*FI*EP1*FJ+(-R3*QI-S3X*FI)*EP3*(-R3*QJ-S3X*FJ)+R3**2*FI*FJ &
              *GZ*CESP/1.2D0
   RKA(16,17)=-S3Y*FI*EP2*(R3*QJ+S3X*FJ)+(-R3*QI-S3X*FI)*EP3*S3Y*FJ
   RKA(17,3)=-R3*FI*(-R1*QJ-S1X*FJ)*GZ*CESP/1.2D0
   RKA(17,4)=-(R3*QI+S3X*FI)*EP2*S1Y*FJ+S3Y*FI*EP3*(-R1*QJ-S1X*FJ)
   RKA(17,5)=(R3*QI+S3X*FI)*EP1*(R1*QJ+S1X*FJ)+S1Y*FI*EP3*S3Y*FJ+R1*FI*R3*FJ*GZ &
             *CESP/1.2D0
   RKA(17,9)=-R3*FI*(-R2*QJ-S2X*FJ)*GZ*CESP/1.2D0
   RKA(17,10)=-(R3*QI+S3X*FI)*EP2*S2Y*FJ+S3Y*FI*EP3*(-R2*QJ-S2X*FJ)
   RKA(17,11)=(R3*QI+S3X*FI)*EP1*(R2*QJ+S2X*FJ)+S2Y*FI*EP3*S3Y*FJ+R2*FI*R3*FJ*GZ &
              *CESP/1.2D0
   RKA(17,15)=-R3*FI*(-R3*QJ-S3X*FJ)*GZ*CESP/1.2D0
   RKA(17,16)=-(R3*QI+S3X*FI)*EP2*S3Y*FJ+S3Y*FI*EP3*(-R3*QJ-S3X*FJ)
   RKA(17,17)=(R3*QI+S3X*FI)*EP1*(R3*QJ+S3X*FJ)+S3Y**2*FI*EP3*FJ+R3**2*FI*FJ*GZ &
              *CESP/1.2D0
   RKA(18,1)=RKO*GZ*CESP*S1Y*FI*R3*FJ/0.2D1
   RKA(18,2)=RKO*GZ*CESP*R3*FI*(-R1*QJ/0.2D1-S1X*FJ/0.2D1)
   RKA(18,6)=RKO*GZ*CESP*R1*FI*R3*FJ
   RKA(18,7)=RKO*GZ*CESP*S2Y*FI*R3*FJ/0.2D1
   RKA(18,8)=RKO*GZ*CESP*R3*FI*(-R2*QJ/0.2D1-S2X*FJ/0.2D1)
   RKA(18,12)=RKO*GZ*CESP*R2*FI*R3*FJ
   RKA(18,13)=RKO*GZ*CESP*S3Y*FI*R3*FJ/0.2D1
   RKA(18,14)=RKO*GZ*CESP*R3*FI*(-R3*QJ/0.2D1-S3X*FJ/0.2D1)
   RKA(18,18)=RKO*GZ*CESP*R3**2*FI*FJ
END SUBROUTINE MakeKA

SUBROUTINE Matriz_global(M)
   USE calc
   INTEGER, INTENT(IN) :: M
   REAL(KIND(0.0D0)) :: AUX1(18*(M+3),18*(M+3)), SUM1
   REAL(KIND(0.0D0)) :: AUX2(18*(M+3),18*(M+3)), SUM2
   AUX1=0.0D0
   AUX2=0.0D0
   DO KI=1,6*(M+3)
```

```
      DO KJ=1,6*(M+3)
       DO I=1,3
         DO J=1,3
          SUM1=0.0D0
          SUM2=0.0D0
          DO K=1,3
             SUM1=SUM1+COSDIR(K,I)*RKEL(3*(KI-1)+K,3*(KJ-1)+J)
             IF (IDIN.EQ.1) THEN
                SUM2=SUM2+COSDIR(K,I)*RMEL(3*(KI-1)+K,3*(KJ-1)+J)
             END IF
          END DO
          AUX1(3*(KI-1)+I,3*(KJ-1)+J)=SUM1
          IF (IDIN.EQ.1) THEN
             AUX2(3*(KI-1)+I,3*(KJ-1)+J)=SUM2
          END IF
         END DO
       END DO
      END DO
    END DO
    DO KI=1,6*(M+3)
      DO KJ=1,6*(M+3)
       DO I=1,3
         DO J=1,3
          SUM1=0.0D0
          SUM2=0.0D0
          DO K=1,3
             SUM1=SUM1+AUX1(3*(KI-1)+I,3*(KJ-1)+K)*COSDIR(K,J)
             IF (IDIN.EQ.1) THEN
                SUM2=SUM2+AUX2(3*(KI-1)+I,3*(KJ-1)+K)*COSDIR(K,J)
             END IF
          END DO
          RKEG(3*(KI-1)+I,3*(KJ-1)+J)=SUM1
          IF (IDIN.EQ.1) THEN
             RMEG(3*(KI-1)+I,3*(KJ-1)+J)=SUM2
          END IF
         END DO
       END DO
      END DO
    END DO
  END SUBROUTINE Matriz_global

  SUBROUTINE Ensamblar(NTR,NEL)
    USE geral
    USE calc
    INTEGER, INTENT(IN) :: NTR,NEL
    INTEGER :: LM(18)
    INTEGER :: IAUX, IAUX1,IAUX2
    IF (NTR.EQ.1) THEN
       IAUX=0
       ELSE
       IAUX=6*NNOS*(3*(NTR-1)+SUM(MPT(1:NTR-1)))
    END IF
    DO L=1,6
       LM(L)=6*(INCID(NEL,1)-1)+L
       LM(6+L)=6*(INCID(NEL,2)-1)+L
       LM(12+L)=6*(INCID(NEL,3)-1)+L
    END DO
    DO IJ=1,MPT(NTR)+3
      IAUX1=IAUX+6*NNOS*(IJ-1)
      DO IK=1,MPT(NTR)+3
       IAUX2=IAUX+6*NNOS*(IK-1)
       DO J=1,18
         DO K=1,18
          RKG(IAUX1+LM(J),IAUX2+LM(K))=RKG(IAUX1+LM(J),IAUX2+LM(K))&
          +RKEG(18*(IJ-1)+J,18*(IK-1)+K)
          IF (IDIN.EQ.1) THEN
             RMG(IAUX1+LM(J),IAUX2+LM(K))=RMG(IAUX1+LM(J),IAUX2+LM(K))&
             +RMEG(18*(IJ-1)+J,18*(IK-1)+K)
          END IF
         END DO
       END DO
      END DO
    END DO
    DEALLOCATE (RKEL,RKEG)
    IF (IDIN.EQ.1) THEN
       DEALLOCATE (RMEL,RMEG)
    END IF
```

```
END SUBROUTINE Ensamblar

SUBROUTINE CargaUniforme
   USE geral
   REAL(KIND(0.0D0)) :: FACT, RL
   DO IT=1,NTRAMOS
     IF (IT.EQ.1) THEN
      IAUX=0
      ELSE
      IAUX=6*NNOS*(3*(IT-1)+SUM(MPT(1:IT-1)))
     END IF
     DO I=1,MPT(IT)+3
       RL=COORD(IT+1,1,1)-COORD(IT,1,1)
      RM=REAL(MPT(IT),KIND(0.0D0))
      IF (I.EQ.1 .OR. I.EQ.MPT(IT)+3) THEN
        FACT=RL/(24.*RM)
      ELSE IF (I.EQ.2 .OR. I.EQ.MPT(IT)+2) THEN
        FACT=RL/(2.*RM)
      ELSE IF (I.EQ.3 .OR. I.EQ.MPT(IT)+1) THEN
        FACT=23.*RL/(24.*RM)
      ELSE
        FACT=RL/RM
      END IF
      DO J=1,NNOS
        DO K=1,3
         RFG(IAUX+6*NNOS*(I-1)+6*(J-1)+K)=CARUNIF(IT,J,K)*FACT
        END DO
      END DO
     END DO
   END DO
END SUBROUTINE CargaUniforme

SUBROUTINE CargaPontual
   USE geral
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
      IAUX=6*NNOS
      ELSE
      IAUX=6*NNOS*(SUM(MPT(1:IS-1))+3*IS-5)
     END IF
     DO J=1,NNOS
      DO K=1,6
        IF (CARPONT(IS,J,K).NE.0.) THEN
         RFG(IAUX-6*NNOS+6*(J-1)+K)=RFG(IAUX-6*NNOS+6*(J-1)+K)+CARPONT(IS,J,K)/6.
         RFG(IAUX+6*(J-1)+K)=RFG(IAUX+6*(J-1)+K)+2.*CARPONT(IS,J,K)/3.
         RFG(IAUX+6*NNOS+6*(J-1)+K)=RFG(IAUX+6*NNOS+6*(J-1)+K)+CARPONT(IS,J,K)/6.
        END IF
      END DO
     END DO
   END DO
END SUBROUTINE CargaPontual

!SUBSTITUÇÃO PELOS DESLOCAMENTOS FISICOS E VETOR DE CARGA MODIFICADO
SUBROUTINE DeslocFisicos(IORMA)                              !IORMA=Ordem da matriz
   USE geral
   INTEGER, INTENT(IN) :: IORMA
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           DO IA=1,IORMA
            RKG(I1-18*NNOS+I2,IA)=RKG(I1-18*NNOS+I2,IA)-RKG(I1-12*NNOS+I2,IA)/4.
            RKG(I1-6*NNOS+I2,IA)=RKG(I1-6*NNOS+I2,IA)-RKG(I1-12*NNOS+I2,IA)/4.
            RKG(I1-12*NNOS+I2,IA)=3.*RKG(I1-12*NNOS+I2,IA)/2.
            IF (CALC_DIN.EQ."s") THEN
               RMG(I1-18*NNOS+I2,IA)=RMG(I1-18*NNOS+I2,IA)-RMG(I1-12*NNOS &
                                    +I2,IA)/4.
               RMG(I1-6*NNOS+I2,IA)=RMG(I1-6*NNOS+I2,IA)-RMG(I1-12*NNOS+I2,IA)/4.
               RMG(I1-12*NNOS+I2,IA)=3.*RMG(I1-12*NNOS+I2,IA)/2.
```

```
              END IF
             END DO
             RFG(I1-18*NNOS+I2)=RFG(I1-18*NNOS+I2)-RFG(I1-12*NNOS+I2)/4.
             RFG(I1-6*NNOS+I2)=RFG(I1-6*NNOS+I2)-RFG(I1-12*NNOS+I2)/4.
             RFG(I1-12*NNOS+I2)=3.*RFG(I1-12*NNOS+I2)/2.
             DO IA=1,IORMA
              RKG(IA,I1-18*NNOS+I2)=RKG(IA,I1-18*NNOS+I2)-RKG(IA,I1-12*NNOS+I2)/4.
              RKG(IA,I1-6*NNOS+I2)=RKG(IA,I1-6*NNOS+I2)-RKG(IA,I1-12*NNOS+I2)/4.
              RKG(IA,I1-12*NNOS+I2)=3.*RKG(IA,I1-12*NNOS+I2)/2.
              IF (CALC_DIN.EQ."s") THEN
                 RMG(IA,I1-18*NNOS+I2)=RMG(IA,I1-18*NNOS+I2)-RMG(IA,I1-12*NNOS &
                                       +I2)/4.
                 RMG(IA,I1-6*NNOS+I2)=RMG(IA,I1-6*NNOS+I2)-RMG(IA,I1-12*NNOS+I2)/4.
                 RMG(IA,I1-12*NNOS+I2)=3.*RMG(IA,I1-12*NNOS+I2)/2.
              END IF
             END DO
            END IF
            IF (IS.NE.NSEC) THEN
              DO IA=1,IORMA
               RKG(I1+I2,IA)=RKG(I1+I2,IA)-RKG(I1+6*NNOS+I2,IA)/4.
               RKG(I1+12*NNOS+I2,IA)=RKG(I1+12*NNOS+I2,IA)-RKG(I1+6*NNOS+I2,IA)/4.
               RKG(I1+6*NNOS+I2,IA)=3.*RKG(I1+6*NNOS+I2,IA)/2.
                  IF (CALC_DIN.EQ."s") THEN
                  RMG(I1+I2,IA)=RMG(I1+I2,IA)-RMG(I1+6*NNOS+I2,IA)/4.
                  RMG(I1+12*NNOS+I2,IA)=RMG(I1+12*NNOS+I2,IA)-RMG(I1+6*NNOS+I2,IA)/4.
                  RMG(I1+6*NNOS+I2,IA)=3.*RMG(I1+6*NNOS+I2,IA)/2.
               END IF
              END DO
              RFG(I1+I2)=RFG(I1+I2)-RFG(I1+6*NNOS+I2)/4.
              RFG(I1+12*NNOS+I2)=RFG(I1+12*NNOS+I2)-RFG(I1+6*NNOS+I2)/4.
              RFG(I1+6*NNOS+I2)=3.*RFG(I1+6*NNOS+I2)/2.
              DO IA=1,IORMA
               RKG(IA,I1+I2)=RKG(IA,I1+I2)-RKG(IA,I1+6*NNOS+I2)/4.
               RKG(IA,I1+12*NNOS+I2)=RKG(IA,I1+12*NNOS+I2)-RKG(IA,I1+6*NNOS+I2)/4.
               RKG(IA,I1+6*NNOS+I2)=3.*RKG(IA,I1+6*NNOS+I2)/2.
               IF (CALC_DIN.EQ."s") THEN
                  RMG(IA,I1+I2)=RMG(IA,I1+I2)-RMG(IA,I1+6*NNOS+I2)/4.
                  RMG(IA,I1+12*NNOS+I2)=RMG(IA,I1+12*NNOS+I2)-RMG(IA,I1+6*NNOS+I2)/4.
                  RMG(IA,I1+6*NNOS+I2)=3.*RMG(IA,I1+6*NNOS+I2)/2.
               END IF
              END DO
            END IF
           END IF
         END DO
        END DO
      END DO
END SUBROUTINE DeslocFisicos

SUBROUTINE Restricoes(IORMA)                                  !IORMA=Ordem da Matriz
   USE geral
   INTEGER, INTENT(IN) :: IORMA
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           DO IA=1,IORMA
            RKG(I1-12*NNOS+I2,IA)=0.0D0
            RKG(IA,I1-12*NNOS+I2)=0.0D0
            IF (CALC_DIN.EQ."s") THEN
               RMG(I1-12*NNOS+I2,IA)=0.0D0
               RMG(IA,I1-12*NNOS+I2)=0.0D0
            END IF
           END DO
           RFG(I1-12*NNOS+I2)=0.0D0
           RKG(I1-12*NNOS+I2,I1-12*NNOS+I2)=1.0D0
         END IF
         IF (IS.NE.NSEC) THEN
           DO IA=1,IORMA
            RKG(I1+6*NNOS+I2,IA)=0.0D0
```

```
            RKG(IA,I1+6*NNOS+I2)=0.0D0
            IF (CALC_DIN.EQ."s") THEN
               RMG(I1+6*NNOS+I2,IA)=0.0D0
               RMG(IA,I1+6*NNOS+I2)=0.0D0
            END IF
           END DO
           RFG(I1+6*NNOS+I2)=0.0D0
           RKG(I1+6*NNOS+I2,I1+6*NNOS+I2)=1.0D0
         END IF
        END IF
      END DO
     END DO
   END DO
END SUBROUTINE Restricoes

SUBROUTINE Sol(A,C,X,NEQ)
   IMPLICIT REAL(KIND(0.0D0)) (A-H,O-Z)
   INTENT(OUT) :: X
   DIMENSION A(NEQ,NEQ),C(NEQ),X(NEQ)
   DO K=2,NEQ
      DO I=K,NEQ
         R=A(I,K-1)/A(K-1,K-1)
         C(I)=C(I)-R*C(K-1)
         DO J=K,NEQ
            A(I,J)=A(I,J)-R*A(K-1,J)
         END DO
      END DO
   END DO
   X(NEQ)=C(NEQ)/A(NEQ,NEQ)
   DO K=NEQ-1,1,-1
      X(K)=C(K)
      DO J=K+1,NEQ
         X(K)=X(K)-A(K,J)*X(J)
      END DO
      X(K)=X(K)/A(K,K)
   END DO
END SUBROUTINE Sol

!RETORNA A UG PARAMETROS
SUBROUTINE UGparametros
   USE geral
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           UG(I1-12*NNOS+I2)=3.*UG(I1-12*NNOS+I2)/2.-(UG(I1-18*NNOS+I2) &
                             +UG(I1-6*NNOS+I2))/4.
         END IF
         IF (IS.NE.NSEC) THEN
           UG(I1+6*NNOS+I2)=3.*UG(I1+6*NNOS+I2)/2.-(UG(I1+I2)+UG(I1+12*NNOS &
                           +I2))/4.
         END IF
        END IF
      END DO
     END DO
   END DO
END SUBROUTINE UGparametros

!RETORNA A RMOD PARAMETROS
SUBROUTINE RMODOparametros
   USE geral
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
```

```
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           DO J=1,NFREQ
            RMODO(I1-12*NNOS+I2,J)=3.0D0*RMODO(I1-12*NNOS+I2,J)/2.0D0-(RMODO(I1 &
                                  -18*NNOS+I2,J)+RMODO(I1-6*NNOS+I2,J))/4.0D0
           END DO
         END IF
         IF (IS.NE.NSEC) THEN
           DO J=1,NFREQ
            RMODO(I1+6*NNOS+I2,J)=3.0D0*RMODO(I1+6*NNOS+I2,J)/2.0D0-(RMODO(I1 &
                                 +I2,J)+RMODO(I1+12*NNOS+I2,J))/4.0D0
           END DO
         END IF
        END IF
      END DO
     END DO
   END DO
END SUBROUTINE RMODOparametros

!ROTINA PARA CALCULAR SUBMATRIZ MA 18x18
SUBROUTINE MakeMA(CESP,RHO,INROT)
   USE pmatrizloc
   REAL(KIND(0.0D0)), INTENT(IN) :: CESP,RHO
   INTEGER, INTENT(IN) :: INROT
   RMA(1,1) = R1 ** 2 * FI * RHO * CESP * FJ
   RMA(1,7) = R1 * FI * RHO * CESP * R2 * FJ
   RMA(1,13) = R1 * FI * RHO * CESP * R3 * FJ
   RMA(2,2) = R1 ** 2 * FI * RHO * CESP * FJ
   RMA(2,8) = R1 * FI * RHO * CESP * R2 * FJ
   RMA(2,14) = R1 * FI * RHO * CESP * R3 * FJ
   RMA(3,3) = R1 ** 2 * FI * RHO * CESP * FJ
   RMA(3,9) = R1 * FI * RHO * CESP * R2 * FJ
   RMA(3,15) = R1 * FI * RHO * CESP * R3 * FJ
   RMA(7,1) = R1 * FI * RHO * CESP * R2 * FJ
   RMA(7,7) = R2 ** 2 * FI * RHO * CESP * FJ
   RMA(7,13) = R2 * FI * RHO * CESP * R3 * FJ
   RMA(8,2) = R1 * FI * RHO * CESP * R2 * FJ
   RMA(8,8) = R2 ** 2 * FI * RHO * CESP * FJ
   RMA(8,14) = R2 * FI * RHO * CESP * R3 * FJ
   RMA(9,3) = R1 * FI * RHO * CESP * R2 * FJ
   RMA(9,9) = R2 ** 2 * FI * RHO * CESP * FJ
   RMA(9,15) = R2 * FI * RHO * CESP * R3 * FJ
   RMA(13,1) = R1 * FI * RHO * CESP * R3 * FJ
   RMA(13,7) = R2 * FI * RHO * CESP * R3 * FJ
   RMA(13,13) = R3 ** 2 * FI * RHO * CESP * FJ
   RMA(14,2) = R1 * FI * RHO * CESP * R3 * FJ
   RMA(14,8) = R2 * FI * RHO * CESP * R3 * FJ
   RMA(14,14) = R3 ** 2 * FI * RHO * CESP * FJ
   RMA(15,3) = R1 * FI * RHO * CESP * R3 * FJ
   RMA(15,9) = R2 * FI * RHO * CESP * R3 * FJ
   RMA(15,15) = R3 ** 2 * FI * RHO * CESP * FJ
   IF (INROT.EQ.1) THEN
      RMA(4,4) = R1 ** 2 * FI * RHO * CESP ** 3 * FJ / 0.12D2
      RMA(4,10) = R1 * FI * RHO * CESP ** 3 * R2 * FJ / 0.12D2
      RMA(4,16) = R1 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(5,5) = R1 ** 2 * FI * RHO * CESP ** 3 * FJ / 0.12D2
      RMA(5,11) = R1 * FI * RHO * CESP ** 3 * R2 * FJ / 0.12D2
      RMA(5,17) = R1 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(10,4) = R1 * FI * RHO * CESP ** 3 * R2 * FJ / 0.12D2
      RMA(10,10) = R2 ** 2 * FI * RHO * CESP ** 3 * FJ / 0.12D2
      RMA(10,16) = R2 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(11,5) = R1 * FI * RHO * CESP ** 3 * R2 * FJ / 0.12D2
      RMA(11,11) = R2 ** 2 * FI * RHO * CESP ** 3 * FJ / 0.12D2
      RMA(11,17) = R2 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(16,4) = R1 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(16,10) = R2 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(16,16) = R3 ** 2 * FI * RHO * CESP ** 3 * FJ / 0.12D2
      RMA(17,5) = R1 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(17,11) = R2 * FI * RHO * CESP ** 3 * R3 * FJ / 0.12D2
      RMA(17,17) = R3 ** 2 * FI * RHO * CESP ** 3 * FJ / 0.12D2
   END IF
END SUBROUTINE MakeMA

!ROTINA PARA CALCULAR SUBMATRIZ GA 18x18
SUBROUTINE MakeGA(RNX,RNY,RNXY)
```

```
   USE pmatrizloc
   REAL(KIND(0.0D0)), INTENT(IN) :: RNX,RNY,RNXY
   RGA(3,3) = ((R1 * QI + S1X * FI) * RNX + S1Y * FI * RNXY) * (R1 *&
              QJ + S1X * FJ) + ((R1 * QI + S1X * FI) * RNXY + S1Y * FI * RNY) *&
              S1Y * FJ
   RGA(3,9) = ((R1 * QI + S1X * FI) * RNX + S1Y * FI * RNXY) * (R2 *&
              QJ + S2X * FJ) + ((R1 * QI + S1X * FI) * RNXY + S1Y * FI * RNY) *&
              S2Y * FJ
   RGA(3,15) = ((R1 * QI + S1X * FI) * RNX + S1Y * FI * RNXY) * (R3 *&
               QJ + S3X * FJ) + ((R1 * QI + S1X * FI) * RNXY + S1Y * FI * RNY) *&
               S3Y * FJ
   RGA(9,3) = ((R2 * QI + S2X * FI) * RNX + S2Y * FI * RNXY) * (R1 *&
              QJ + S1X * FJ) + ((R2 * QI + S2X * FI) * RNXY + S2Y * FI * RNY) *&
              S1Y * FJ
   RGA(9,9) = ((R2 * QI + S2X * FI) * RNX + S2Y * FI * RNXY) * (R2 *&
              QJ + S2X * FJ) + ((R2 * QI + S2X * FI) * RNXY + S2Y * FI * RNY) *&
              S2Y * FJ
   RGA(9,15) = ((R2 * QI + S2X * FI) * RNX + S2Y * FI * RNXY) * (R3 *&
               QJ + S3X * FJ) + ((R2 * QI + S2X * FI) * RNXY + S2Y * FI * RNY) *&
               S3Y * FJ
   RGA(15,3) = ((R3 * QI + S3X * FI) * RNX + S3Y * FI * RNXY) * (R1 *&
               QJ + S1X * FJ) + ((R3 * QI + S3X * FI) * RNXY + S3Y * FI * RNY) *&
               S1Y * FJ
   RGA(15,9) = ((R3 * QI + S3X * FI) * RNX + S3Y * FI * RNXY) * (R2 *&
               QJ + S2X * FJ) + ((R3 * QI + S3X * FI) * RNXY + S3Y * FI * RNY) *&
               S2Y * FJ
   RGA(15,15) = ((R3 * QI + S3X * FI) * RNX + S3Y * FI * RNXY) * (R3&
                * QJ + S3X * FJ) + ((R3 * QI + S3X * FI) * RNXY + S3Y * FI * RNY)&
                * S3Y * FJ
END SUBROUTINE MakeGA

SUBROUTINE Param_elem(NTR,NEL)
   USE geral
   USE calc
   INTEGER, INTENT(IN) :: NTR,NEL
   INTEGER :: LM(18)
   INTEGER :: IAUX, IAUX1,IAUX2
   M=MPT(NTR)
   ALLOCATE (UGEL(18*(M+3)),UGEG(18*(M+3)))
   UGEL=0.0D0;UGEG=0.0D0
   IF (NTR.EQ.1) THEN
      IAUX=0
      ELSE
      IAUX=6*NNOS*(3*(NTR-1)+SUM(MPT(1:NTR-1)))
   END IF
   DO L=1,6
      LM(L)=6*(INCID(NEL,1)-1)+L
      LM(6+L)=6*(INCID(NEL,2)-1)+L
      LM(12+L)=6*(INCID(NEL,3)-1)+L
   END DO
   DO IJ=1,MPT(NTR)+3
     IAUX1=IAUX+6*NNOS*(IJ-1)
      DO J=1,18
         UGEG(18*(IJ-1)+J)=UG(IAUX1+LM(J))
      END DO
   END DO
END SUBROUTINE Param_elem

SUBROUTINE Transf_local(M)
   USE calc
   INTEGER, INTENT(IN) :: M
   REAL(KIND(0.0D0)) :: SUM1
   DO KI=1,6*(M+3)
      DO I=1,3
        SUM1=0.0D0
        DO J=1,3
         SUM1=SUM1+COSDIR(I,J)*UGEG(3*(KI-1)+J)
        END DO
        UGEL(3*(KI-1)+I)=SUM1
      END DO
   END DO
END SUBROUTINE Transf_local

SUBROUTINE Matriz_Geom(NTR,NEL)
   USE geral
   USE calc
   USE pmatrizloc
```

```
REAL(KIND(0.0D0)), DIMENSION(5) :: FX1,W1,FX2,W2
REAL(KIND(0.0D0)) :: AGA(18,18),Z,RN,RV,XIX,CESP
REAL(KIND(0.0D0)), ALLOCATABLE :: RMNX(:,:), RMNY(:,:), RMNXY(:,:)
REAL(KIND(0.0D0)) :: RNX, RNY, RNXY
IF (E1.EQ.0) THEN
   E1=ELAST/(1.-POISS**2)
   E2=ELAST*POISS/(1.-POISS**2)
   E3=ELAST/(1.+POISS)/2.0D0
END IF
CESP=ESPES(NEL)
M=MPT(NTR)
ALLOCATE (RGEL(18*(M+3),18*(M+3)),RGEG(18*(M+3),18*(M+3)))
RGEL=0.0D0;RGEG=0.0D0
NN=4                                                    !Pontos de Gauss em x
NE=4                                                    !Pontos de Gauss em y
CALL Qgauss(NN,NE,FX1,W1,FX2,W2)
ALLOCATE (RMNX(M*NN,NE),RMNY(M*NN,NE),RMNXY(M*NN,NE))
RMNX=0.0D0;RMNY=0.0D0;RMNXY=0.0D0
DO IM=1,M
  DO ING=1,NN
   RN=(FX1(ING)+1.0D0)/2.0D0+REAL((IM-1),KIND(0.0D0))
   RV=RN*(-Y(2)+Y(3)-Y(4))/REAL(M,KIND(0.0D0))+Y(4)
   DO IEG=1,NE
      XI=FX2(IEG)
      XIX=((1.0D0-XI)*(-Y(2))-(1.0D0+XI)*(Y(3)-Y(4)))/LONG/RV
      R1=0.5D0*XI*(XI-1.)
      R2=1.0D0-XI**2
      R3=0.5D0*XI*(XI+1.0D0)
      S1Y=(XI-0.5D0)*2.0D0/RV
      S2Y=-4.0D0*XI/RV
      S3Y=(XI+0.5D0)*2.D0/RV
      S1X=(XI-0.5D0)*XIX
      S2X=-2.0D0*XI*XIX
      S3X=(XI+0.5D0)*XIX
      CALL Esf_inic(M,RN,RNX,RNY,RNXY)
      RMNX(NN*(IM-1)+ING,IEG)=CESP*RNX
      RMNY(NN*(IM-1)+ING,IEG)=CESP*RNY
      RMNXY(NN*(IM-1)+ING,IEG)=CESP*RNXY
   END DO
  END DO
END DO
DO II=1,(M+3)
  DO JJ=II,(M+3)
   I=II-2
   J=JJ-2
   NT=0
   IF (I.EQ.-1 .AND. J.EQ.-1 .OR. I.EQ.-1 .AND. J.EQ.0 .OR. I.EQ.&
      -1 .AND. J.EQ.1 .OR. I.EQ.M+1 .AND. J.EQ.M+1 .OR. I.EQ.M .AND.&
      J.EQ.M+1 .OR. I.EQ.M-1 .AND. J.EQ.M+1) NT=1
   IF (I.EQ.0 .AND. J.EQ.0 .OR. I.EQ.0 .AND. J.EQ.1 .OR. I.EQ.M &
      .AND. J.EQ.M .OR. I.EQ.M-1 .AND. J.EQ.M) NT=2
   IF (I.EQ.1 .AND. J.EQ.1 .OR. I.EQ.M-1 .AND. J.EQ.M-1) NT=3
   IF (J-I.LT.4 .AND. NT.EQ.0) NT=I-J+4
   IF (NT.NE.0) THEN
     AGA=0.0D0
     DO IQ=1,NT
      DO IG=1,NN
        Z=FX1(IG)
        CALL Pinic(I,IQ,NT,M,IP)
        RN=(Z+1.0D0)/2.0D0+REAL(IP,KIND(0.0D0))
        CALL Detfun(I,J,IQ,NT,RN,LONG,M,2)   !2 SE VAI CALCULAR PI E PJ
        RV=RN*(-Y(2)+Y(3)-Y(4))/REAL(M,KIND(0.0D0))+Y(4)
        DO JG=1,NE
         XI=FX2(JG)
         XIX=((1.0D0-XI)*(-Y(2))-(1.0D0+XI)*(Y(3)-Y(4)))/LONG/RV
         R1=0.5D0*XI*(XI-1.0D0)
         R2=1.0D0-XI**2
         R3=0.5D0*XI*(XI+1.0D0)
         S1Y=(XI-0.5D0)*2.0D0/RV
         S2Y=-4.0D0*XI/RV
         S3Y=(XI+0.5D0)*2.0D0/RV
         S1X=(XI-0.5D0)*XIX
         S2X=-2.0D0*XI*XIX
         S3X=(XI+0.5D0)*XIX
         RNX=RMNX(NN*IP+IG,JG)
         RNY=RMNY(NN*IP+IG,JG)
         RNXY=RMNXY(NN*IP+IG,JG)
```

```
            CALL MakeGA(RNX,RNY,RNXY)
            DO IS=1,18
              DO JS=1,18
               AGA(IS,JS)=AGA(IS,JS)+W1(IG)*W2(JG)*RGA(IS,JS)*RV
              END DO
            END DO
           END DO
          END DO
          DO IA=1,18
           DO IB=1,18
             RGEL(18*I+18+IA,18*J+18+IB)=LONG*AGA(IA,IB)/4./REAL(M,KIND(0.0D0))
             RGEL(18*J+18+IB,18*I+18+IA)=RGEL(18*I+18+IA,18*J+18+IB)
           END DO
          END DO
        END IF
       END DO
   END DO
   DEALLOCATE(RMNX,RMNY,RMNXY)
   CALL Geom_global(M)
END SUBROUTINE Matriz_Geom

SUBROUTINE Esf_inic(M,RN,RNX,RNY,RNXY)
   USE calc
   USE pmatrizloc
   INTEGER, INTENT(IN) :: M
   REAL(KIND(0.0D0)), INTENT(IN) :: RN
   REAL(KIND(0.0D0)), INTENT(OUT) :: RNX,RNY,RNXY
   INTEGER :: I
   REAL(KIND(0.0D0)) :: RI,RM,FIC(4),QIC(4),S1,S2,S3
   RNX=0.0D0
   RNY=0.0D0
   RNXY=0.0D0
   I=INT(RN)
   RI=REAL(I,KIND(0.0D0))
   RM=REAL(M,KIND(0.0D0))
   FIC(1)=(RI+1-RN)**3/6.0D0
   FIC(2)=(1.0D0+3.0D0*(RI+1-RN)+3.0D0*(RI+1-RN)**2-3.0D0*(RI+1-RN)**3)/6.0D0
   FIC(3)=(1.0D0+3.0D0*(RN-RI)+3.0D0*(RN-RI)**2-3.0D0*(RN-RI)**3)/6.0D0
   FIC(4)=(RN-RI)**3/6.0D0
   QIC(1)=-(RI+1-RN)**2*RM/(2.0D0*LONG)
   QIC(2)=-(1.0D0+2.0D0*(RI+1-RN)-3.0D0*(RI+1-RN)**2)*RM/(2.0D0*LONG)
   QIC(3)=(1.0D0+2.0D0*(RN-RI)-3.0D0*(RN-RI)**2)*RM/(2.0D0*LONG)
   QIC(4)=(RN-RI)**2*RM/(2.0D0*LONG)
   DO J=1,4
      S1=E1*(R1*QIC(J)+S1X*FIC(J))*UGEL(18*(I+J-1)+1)+E2*S1Y*FIC(J)*UGEL(18*(I &
         +J-1)+2)
      S2=E1*(R2*QIC(J)+S2X*FIC(J))*UGEL(18*(I+J-1)+7)+E2*S2Y*FIC(J)*UGEL(18*(I &
         +J-1)+8)
      S3=E1*(R3*QIC(J)+S3X*FIC(J))*UGEL(18*(I+J-1)+13)+E2*S3Y*FIC(J)*UGEL(18*(I &
         +J-1)+14)
      RNX=RNX+S1+S2+S3
      S1=E2*(R1*QIC(J)+S1X*FIC(J))*UGEL(18*(I+J-1)+1)+E1*S1Y*FIC(J)*UGEL(18*(I &
         +J-1)+2)
      S2=E2*(R2*QIC(J)+S2X*FIC(J))*UGEL(18*(I+J-1)+7)+E1*S2Y*FIC(J)*UGEL(18*(I &
         +J-1)+8)
      S3=E2*(R3*QIC(J)+S3X*FIC(J))*UGEL(18*(I+J-1)+13)+E1*S3Y*FIC(J)*UGEL(18*(I &
         +J-1)+14)
      RNY=RNY+S1+S2+S3
      S1=E3*S1Y*FIC(J)*UGEL(18*(I+J-1)+1)+E3*(R1*QIC(J)+S1X*FIC(J))*UGEL(18*(I &
         +J-1)+2)
      S2=E3*S2Y*FIC(J)*UGEL(18*(I+J-1)+7)+E3*(R2*QIC(J)+S2X*FIC(J))*UGEL(18*(I &
         +J-1)+8)
      S3=E3*S3Y*FIC(J)*UGEL(18*(I+J-1)+13)+E3*(R3*QIC(J)+S3X*FIC(J))*UGEL(18*(I &
         +J-1)+14)
      RNXY=RNXY+S1+S2+S3
   END DO
END SUBROUTINE Esf_inic

SUBROUTINE Geom_global(M)
   USE calc
   INTEGER, INTENT(IN) :: M
   REAL(KIND(0.0D0)) :: AUX1(18*(M+3),18*(M+3)), SUM1
   REAL(KIND(0.0D0)) :: AUX2(18*(M+3),18*(M+3)), SUM2
   AUX1=0.0D0
   AUX2=0.0D0
   DO KI=1,6*(M+3)
```

```
      DO KJ=1,6*(M+3)
       DO I=1,3
         DO J=1,3
          SUM1=0.0D0
          SUM2=0.0D0
          DO K=1,3
             SUM1=SUM1+COSDIR(K,I)*RGEL(3*(KI-1)+K,3*(KJ-1)+J)
          END DO
          AUX1(3*(KI-1)+I,3*(KJ-1)+J)=SUM1
         END DO
       END DO
      END DO
   END DO
   DO KI=1,6*(M+3)
     DO KJ=1,6*(M+3)
      DO I=1,3
        DO J=1,3
         SUM1=0.0D0
         SUM2=0.0D0
         DO K=1,3
            SUM1=SUM1+AUX1(3*(KI-1)+I,3*(KJ-1)+K)*COSDIR(K,J)
         END DO
         RGEG(3*(KI-1)+I,3*(KJ-1)+J)=SUM1
        END DO
       END DO
     END DO
   END DO
END SUBROUTINE Geom_global

SUBROUTINE Ensam_geom(NTR,NEL)
   USE geral
   USE calc
   INTEGER, INTENT(IN) :: NTR,NEL
   INTEGER :: LM(18)
   INTEGER :: IAUX, IAUX1,IAUX2
   IF (NTR.EQ.1) THEN
      IAUX=0
      ELSE
      IAUX=6*NNOS*(3*(NTR-1)+SUM(MPT(1:NTR-1)))
   END IF
   DO L=1,6
      LM(L)=6*(INCID(NEL,1)-1)+L
      LM(6+L)=6*(INCID(NEL,2)-1)+L
      LM(12+L)=6*(INCID(NEL,3)-1)+L
   END DO
   DO IJ=1,MPT(NTR)+3
     IAUX1=IAUX+6*NNOS*(IJ-1)
     DO IK=1,MPT(NTR)+3
      IAUX2=IAUX+6*NNOS*(IK-1)
      DO J=1,18
        DO K=1,18
         RGG(IAUX1+LM(J),IAUX2+LM(K))=RGG(IAUX1+LM(J),IAUX2+LM(K))&
                                      +RGEG(18*(IJ-1)+J,18*(IK-1)+K)
        END DO
       END DO
     END DO
   END DO
   DEALLOCATE (UGEL,UGEG,RGEL,RGEG)
END SUBROUTINE Ensam_geom

SUBROUTINE DeslFis_geom(IORMA)
   USE geral
   INTEGER, INTENT(IN) :: IORMA
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           DO IA=1,IORMA
            RGG(I1-18*NNOS+I2,IA)=RGG(I1-18*NNOS+I2,IA)-RGG(I1-12*NNOS+I2,IA)/4.
```

```
            RGG(I1-6*NNOS+I2,IA)=RGG(I1-6*NNOS+I2,IA)-RGG(I1-12*NNOS+I2,IA)/4.
            RGG(I1-12*NNOS+I2,IA)=3.*RGG(I1-12*NNOS+I2,IA)/2.
           END DO
           DO IA=1,IORMA
            RGG(IA,I1-18*NNOS+I2)=RGG(IA,I1-18*NNOS+I2)-RGG(IA,I1-12*NNOS+I2)/4.
            RGG(IA,I1-6*NNOS+I2)=RGG(IA,I1-6*NNOS+I2)-RGG(IA,I1-12*NNOS+I2)/4.
            RGG(IA,I1-12*NNOS+I2)=3.*RGG(IA,I1-12*NNOS+I2)/2.
           END DO
         END IF
         IF (IS.NE.NSEC) THEN
           DO IA=1,IORMA
            RGG(I1+I2,IA)=RGG(I1+I2,IA)-RGG(I1+6*NNOS+I2,IA)/4.
            RGG(I1+12*NNOS+I2,IA)=RGG(I1+12*NNOS+I2,IA)-RGG(I1+6*NNOS+I2,IA)/4.
            RGG(I1+6*NNOS+I2,IA)=3.*RGG(I1+6*NNOS+I2,IA)/2.
           END DO
           DO IA=1,IORMA
            RGG(IA,I1+I2)=RGG(IA,I1+I2)-RGG(IA,I1+6*NNOS+I2)/4.
            RGG(IA,I1+12*NNOS+I2)=RGG(IA,I1+12*NNOS+I2)-RGG(IA,I1+6*NNOS+I2)/4.
            RGG(IA,I1+6*NNOS+I2)=3.*RGG(IA,I1+6*NNOS+I2)/2.
           END DO
         END IF
        END IF
       END DO
      END DO
   END DO
END SUBROUTINE DeslFis_geom

SUBROUTINE Restric_geom(IORMA)
   USE geral
   INTEGER, INTENT(IN) :: IORMA
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           DO IA=1,IORMA
            RGG(I1-12*NNOS+I2,IA)=0.0D0
            RGG(IA,I1-12*NNOS+I2)=0.0D0
           END DO
         END IF
         IF (IS.NE.NSEC) THEN
           DO IA=1,IORMA
            RGG(I1+6*NNOS+I2,IA)=0.0D0
            RGG(IA,I1+6*NNOS+I2)=0.0D0
           END DO
         END IF
        END IF
      END DO
     END DO
   END DO
END SUBROUTINE Restric_geom

!RETORNA A RMODOI PARAMETROS
SUBROUTINE RMODOIparam_geom
   USE geral
   INTEGER :: I1,I2
   DO IS=1,NSEC
     IF (IS.EQ.1) THEN
       I1=0
      ELSE
      I1=6*NNOS*(3*(IS-1)+SUM(MPT(1:(IS-1))))
     END IF
     DO IN=1,NNOS
      DO I=1,6
        I2=6*(IN-1)+I
        IF (IRESTR(IS,IN,I).EQ.1) THEN
         IF (IS.NE.1) THEN
           DO J=1,NMODOS
            RMODOI(I1-12*NNOS+I2,J)=3.0D0*RMODOI(I1-12*NNOS+I2,J)/2.0D0-(RMODOI &
                                    (I1-18*NNOS+I2,J)+RMODOI(I1-6*NNOS+I2,J))/4.0D0
```

```
          END DO
        END IF
        IF (IS.NE.NSEC) THEN
          DO J=1,NMODOS
           RMODOI(I1+6*NNOS+I2,J)=3.0D0*RMODOI(I1+6*NNOS+I2,J)/2.0D0-(RMODOI &
                                  (I1+I2,J)+RMODOI(I1+12*NNOS+I2,J))/4.0D0
          END DO
        END IF
       END IF
     END DO
    END DO
  END DO
END SUBROUTINE RMODOIparam_geom
```

## B.2. Definições

```
MODULE geral
   INTEGER :: NTRAMOS,NNOS,NELEM,NSEC,NFREQ,NMODOS
   CHARACTER(1) :: CALC_EST,CALC_DIN,INER_ROT,CALC_INS
   REAL(KIND(0.0D0)) :: ELAST,POISS,RHO
   REAL(KIND(0.0D0)),ALLOCATABLE :: COORD(:,:,:),ESPES(:),CARUNIF(:,:,:),&
   CARPONT(:,:,:)
   INTEGER, ALLOCATABLE :: MPT(:),NDIV(:),INCID(:,:),IRESTR(:,:,:)
   REAL(KIND(0.0D0)),ALLOCATABLE ::
RKG(:,:),RMG(:,:),RGG(:,:),RKGI(:,:),RFG(:),UG(:)
   REAL(KIND(0.0D0)),ALLOCATABLE :: BETA(:),RMODO(:,:),BETAI(:),RMODOI(:,:)
   COMPLEX(KIND(0.0D0)),ALLOCATABLE :: ALPHA(:),AUTOV(:,:),ALPHAI(:),AUTOVI(:,:)
END MODULE geral

MODULE calc
   INTEGER :: IDIN
   REAL(KIND(0.0D0)) :: VLX(3),VLY(3),VLZ(3),LONG,Y(4)
   REAL(KIND(0.0D0)) :: COSDIR(3,3)
   REAL(KIND(0.0D0)), ALLOCATABLE :: RKEL(:,:), RKEG(:,:), RMEL(:,:), RMEG(:,:),&
   RGEL(:,:), RGEG(:,:), UGEL(:), UGEG(:)
END MODULE calc

MODULE pmatrizloc
   REAL(KIND(0.0D0)):: E1,E2,E3,EP1,EP2,EP3,GZ,RKO
   REAL(KIND(0.0D0)):: RKA(18,18),RMA(18,18),RGA(18,18),FI,QI,FJ,QJ
   REAL(KIND(0.0D0)):: R1,R2,R3,S1Y,S2Y,S3Y,S1X,S2X,S3X
END MODULE pmatrizloc
```

## B.3. Rotinas Gerais

```
!CALCULA O MODULO DE UM VETOR DE TRÊS DIMENSÕES
SUBROUTINE ModuloVetor(VETOR,RMOD)
   REAL(KIND(0.0D0)), INTENT(IN) :: VETOR(3)
   REAL(KIND(0.0D0)), INTENT(OUT) :: RMOD
   REAL(KIND(0.0D0)) :: AUX(3),S
   AUX=VETOR**2
   S=SUM(AUX)
   RMOD=SQRT(S)
END SUBROUTINE ModuloVetor

!ROTINA PARA CALCULAR PONTOS E PESOS NA QUADRATURA DE GAUSS
SUBROUTINE Qgauss(NN,NE,F1,W1,F2,W2)
   INTEGER, INTENT(IN) :: NN,NE
   REAL(KIND(0.0D0)), INTENT(OUT) :: F1(5),W1(5),F2(5),W2(5)
   IF (NN.EQ.1) THEN
      F1(1)=0
      W1(1)=2
   END IF
   IF (NN.EQ.2) THEN
      F1(1)=-1./SQRT(3.)
      F1(2)=1./SQRT(3.)
      W1(1)=1
      W1(2)=1
   END IF
   IF (NN.EQ.3) THEN
      F1(1)=-SQRT(3./5.)
      F1(2)=0
```

```
      F1(3)=SQRT(3./5.)
      W1(1)=5./9.
      W1(2)=8./9.
      W1(3)=5./9.
   END IF
   IF (NN.EQ.4) THEN
      F1(1)=-SQRT((3.+2.*SQRT(6./5.))/7.)
      F1(2)=-SQRT((3.-2.*SQRT(6./5.))/7.)
      F1(3)=SQRT((3.-2.*SQRT(6./5.))/7.)
      F1(4)=SQRT((3.+2.*SQRT(6./5.))/7.)
      W1(1)=0.5-SQRT(5./6.)/6.
      W1(2)=0.5+SQRT(5./6.)/6.
      W1(3)=0.5+SQRT(5./6.)/6.
      W1(4)=0.5-SQRT(5./6.)/6.
   END IF
   IF (NN.EQ.5) THEN
      F1(1)=-SQRT(5.+2.*SQRT(10./7.))/3.
      F1(2)=-SQRT(5.-2.*SQRT(10./7.))/3.
      F1(3)=0.
      F1(4)=SQRT(5.-2.*SQRT(10./7.))/3.
      F1(5)=SQRT(5.+2.*SQRT(10./7.))/3.
      W1(1)=(322.-13.*SQRT(70.))/900.
      W1(2)=(322.+13.*SQRT(70.))/900.
      W1(3)=512./900.
      W1(4)=(322.+13.*SQRT(70.))/900.
      W1(5)=(322.-13.*SQRT(70.))/900.
   END IF
   IF (NE.EQ.1) THEN
      F2(1)=0
      W2(1)=2
   END IF
   IF (NE.EQ.2) THEN
      F2(1)=-1./SQRT(3.)
      F2(2)=1./SQRT(3.)
      W2(1)=1
      W2(2)=1
   END IF
   IF (NE.EQ.3) THEN
      F2(1)=-SQRT(3./5.)
      F2(2)=0
      F2(3)=SQRT(3./5.)
      W2(1)=5./9.
      W2(2)=8./9.
      W2(3)=5./9.
   END IF
   IF (NE.EQ.4) THEN
      F2(1)=-SQRT((3.+2.*SQRT(6./5.))/7.)
      F2(2)=-SQRT((3.-2.*SQRT(6./5.))/7.)
      F2(3)=SQRT((3.-2.*SQRT(6./5.))/7.)
      F2(4)=SQRT((3.+2.*SQRT(6./5.))/7.)
      W2(1)=0.5-SQRT(5./6.)/6.
      W2(2)=0.5+SQRT(5./6.)/6.
      W2(3)=0.5+SQRT(5./6.)/6.
      W2(4)=0.5-SQRT(5./6.)/6.
   END IF
   IF (NE.EQ.5) THEN
      F2(1)=-SQRT(5.+2.*SQRT(10./7.))/3.
      F2(2)=-SQRT(5.-2.*SQRT(10./7.))/3.
      F2(3)=0.
      F2(4)=SQRT(5.-2.*SQRT(10./7.))/3.
      F2(5)=SQRT(5.+2.*SQRT(10./7.))/3.
      W2(1)=(322.-13.*SQRT(70.))/900.
      W2(2)=(322.+13.*SQRT(70.))/900.
      W2(3)=512./900.
      W2(4)=(322.+13.*SQRT(70.))/900.
      W2(5)=(322.-13.*SQRT(70.))/900.
   END IF
END SUBROUTINE Qgauss
```

## B.4. Entrada de Dados

```
SUBROUTINE Leitura_dados
   CHARACTER(1)::I
   WRITE(*,*) "Introduzir os dados inicias no arquivo DADOS1 antes de continuar"
   WRITE(*,*) "   1. Criar arquivo DADOS2 con informacao adicional"
   WRITE(*,*) "   2. Dados adicionais inseridos, Continuar"
   WRITE(*,*) "   3. Salir"
   READ(*,*) I
   SELECT CASE (I)
      CASE ("1")
         CALL Ler_dados1
         CALL Criar_dados2
         STOP
      CASE ("2")
         CALL Ler_dados1
         CALL Ler_dados2
      CASE ("3")
         STOP
      CASE DEFAULT
         STOP
   END SELECT
END SUBROUTINE Leitura_dados

SUBROUTINE Ler_dados1
   USE geral
   OPEN(11,FILE='DADOS1.ASC',STATUS='UNKNOWN')
   READ(11,"(4/,30X,I10)") NTRAMOS
   READ(11,"(30X,I10)") NNOS,NELEM
   READ(11,"(/,30X,A)") CALC_EST
   READ(11,"(2/,30X,A)") CALC_DIN
   READ(11,"(2/,30X,A)") INER_ROT
   READ(11,"(/,30X,I10)") NFREQ
   READ(11,"(2/,30X,A)") CALC_INS
   READ(11,"(/,30X,I10)") NMODOS
   NSEC=NTRAMOS+1
   CLOSE(11)
END SUBROUTINE Ler_dados1

SUBROUTINE Criar_dados2
   USE geral
   OPEN(12,FILE='DADOS2.ASC',STATUS='UNKNOWN')
   WRITE(12,*) "PROGRAMA VIGA GENERALIZADA COM FUNCOES SPLINE"
   WRITE(12,"(2/,' PROPRIEDADES DO MATERIAL')")
   WRITE(12,"(/,' Modulo de elasticidad em [KN/m^2]    :',E10.2)") 0.21E+09
   WRITE(12,"(' Coeficiente de Poisson               :',F8.4)") 0.3
   WRITE(12,"(' Densidade especifica [KN*seg^2/m^4]  :',F8.4)") 7.95
   WRITE(12,"(2/,' INFORMAÇÃO DOS NOS')")
   DO I=1,NSEC
      WRITE(12,"(/,' COORDENADAS SECAO',I3)") I
      WRITE(12,"(' x:',F7.2)") 6.*(REAL(I)-1)
      DO J=1,NNOS
         WRITE(12,"(' No',I3,'    y:',F7.2,'               z:',F7.2)")&
         J,0.,0.
      END DO
   END DO
   WRITE(12,"(/,' NUMERO DE DIVISOES POR TRAMO')")
   WRITE(12,"(' Para o calculo',7X,'Para os resultados')")
   DO I=1,NTRAMOS
      WRITE(12,"(' Tramo',I3,':'I3,9X,'Tramo',I3,':'I3)") I,5,I,8
   END DO
   WRITE(12,"(2/,' INFORMAÇÃO DOS ELEMENTOS',/)")
   DO I=1,NELEM
      WRITE(12,"(' Elemento',I3,'    No inic.:',I3,'      No inter:',I3,&
      '      No final:',I3,'         Espes.:',F7.2)") I,2*I-1,2*I,2*I+1,0.05
   END DO
   WRITE(12,"(2/,' CARREGAMENTO UNIFORME POR LINHA NODAL')")
   DO I=1,NTRAMOS
      WRITE(12,"(/' TRAMO',I3)") I
      DO J=1,NNOS
         WRITE(12,"(' No',I3,'   qx:',F7.2,&
         '               qy:',F7.2,'              qz:',F7.2)")&
         J,0.,0.,0.
      END DO
   END DO
```

```
      WRITE(12,"(2/,' CARREGAMENTO PONTUAL NOS NÓS')")
   DO I=1,NSEC
      WRITE(12,"(/,' SECAO',I3)") I
      DO J=1,NNOS
         WRITE(12,"(' No',I3,'   Fx:',F7.2,'       Fy:',F7.2,'       Fz:',F7.2,&
         '       Mx:',F7.2,'       My:',F7.2,'       Mz:',F7.2)")&
         J,0.,0.,0.,0.,0.,0.
      END DO
   END DO
      WRITE(12,"(2/,' RESTRIÇÕES')")
   DO I=1,NSEC
      WRITE(12,"(/' SEÇÃO',I3)") I
      DO J=1,NNOS
         WRITE(12,"(' No',I3,'   Rx:',I3,'      Ry:',I3,&
         '      Rz:',I3,'      Mx:',I3,'      My:',I3,'      Mz:',I3)")&
         J,0,0,0,0,0,0
      END DO
   END DO
   CLOSE(12)
END SUBROUTINE Criar_dados2

SUBROUTINE Ler_dados2
   USE geral
   REAL(KIND(0.0D0)) :: X
   ALLOCATE (COORD(NSEC,NNOS,3),MPT(NTRAMOS),NDIV(NTRAMOS),INCID(NELEM,3) &
            ,ESPES(NELEM),CARUNIF(NTRAMOS,NNOS,3),CARPONT(NSEC,NNOS,6) &
            ,IRESTR(NSEC,NNOS,6))
   MPT=0;NDIV=0;INCID=0;ESPES=0.;IRESTR=0
   COORD=0.0D0;CARUNIF=0.0D0;CARPONT=0.0D0
   OPEN(12,FILE='DADOS2.ASC',STATUS='UNKNOWN')
   READ(12,"(5/,(39X,F22.0))") ELAST,POISS,RHO
   READ(12,"(2/)")
   DO I=1,NSEC
      READ(12,"(2/,3X,F22.0)") X
      DO J=1,NNOS
         COORD(I,J,1)=X
         READ(12,"(12X,F22.0,2X,F22.0,2X,F22.0)") (COORD(I,J,K),K=2,3)
      END DO
   END DO
   READ(12,"(2/)")
   DO I=1,NTRAMOS
      READ(12,"(10X,I12,9X,I12)") MPT(I), NDIV(I)
   END DO
   READ(12,"(3/)")
   DO I=1,NELEM
      READ(12,"(25X,I9,9X,I9,9X,I12,7X,F22.0)") (INCID(I,J),J=1,3),ESPES(I)
   END DO
   READ(12,"(2/)")
   DO I=1,NTRAMOS
      READ(12,"(/)")
      DO J=1,NNOS
         READ(12,"(12X,F22.0,3X,F22.0,3X,F22.0)") (CARUNIF(I,J,K),K=1,3)
      END DO
   END DO
   READ(12,"(2/)")
   DO I=1,NSEC
      READ(12,"(/)")
      DO J=1,NNOS
         READ(12,"(12X,F14.0,3X,F14.0,3X,F14.0,3X,F14.0,3X,F14.0,3X,F14.0)") &
         (CARPONT(I,J,K),K=1,6)
      END DO
   END DO
   READ(12,"(2/)")
   DO I=1,NSEC
      READ(12,"(/)")
      DO J=1,NNOS
         READ(12,"(12X,I9,3X,I9,3X,I9,3X,I9,3X,I9,3X,I9)") (IRESTR(I,J,K),K=1,6)
      END DO
   END DO
   CLOSE(12)
END SUBROUTINE Ler_dados2
```

## B.5. Saída de Dados

```
SUBROUTINE Resul_est
   USE geral
   INTEGER :: NTR, IN, ID
   REAL(KIND(0.0D0)) :: X, A1(6), DESL
   OPEN(14,FILE='RESEST.ASC',STATUS='UNKNOWN')
   WRITE(14,*) 'Deslocamentos e Rotações'
   DO IN=1,NNOS
      WRITE(14,"(' Nó N° ',I3)") IN
      WRITE(14,"(5X,'x',17X,'Dx',16X,'Dy',16X,'Dz',16X,'Rx',16X,'Ry',16X,'Rz')")
      DO IC=1,6
         CALL Desloc(1,IN,IC,1,1,0.0D0,DESL)
         A1(IC)=DESL
      END DO
      WRITE(14,"(F12.4,2X,6E18.8)") 0.0D0,(A1(I),I=1,6)
      DO NTR=1,NTRAMOS
         DO IX=1,NDIV(NTR)
            X=COORD(NTR,1,1)+REAL(IX,KIND(0.0D0))*(COORD(NTR+1,1,1) &
              -COORD(NTR,1,1))/REAL(NDIV(NTR),KIND(0.0D0))
            DO ID=1,6
               CALL Desloc(NTR,IN,ID,1,1,X,DESL)
               A1(ID)=DESL
            END DO
            WRITE(14,"(F12.4,2X,6E18.8)") X,(A1(I),I=1,6)
         END DO
      END DO
   END DO
   CLOSE(14)
END SUBROUTINE Resul_est

SUBROUTINE Resul_din
   USE geral
   REAL(KIND(0.0D0)) :: X, A1(3), DESL
   INTEGER :: IR,NTR
   OPEN(15,FILE='RESDIN.ASC',STATUS='UNKNOWN')
   DO I=1,NFREQ
      IF (REAL(ALPHA(I)).LT.0.0D0) THEN
         IR=1
      END IF
   END DO
   IF (IR.EQ.0) THEN
     WRITE(15,*) 'Frequencias Naturais da Estrutura'
     WRITE(15,*) SQRT(REAL(ALPHA(1:NFREQ)/BETA(1:NFREQ)))/2.0D0/3.141593D0
     DO IM=1,NFREQ
      WRITE(15,"(' MODO N° ',I3)") IM
      DO IN=1,NNOS
         WRITE(15,"(' Nó N° ',I3)") IN
         WRITE(15,"(5X,'x',17X,'Dx',16X,'Dy',16X,'Dz')")
         DO IC=1,3
            CALL Desloc(1,IN,IC,2,IM,0.0D0,DESL)
            A1(IC)=DESL
         END DO
         WRITE(15,"(F12.4,2X,3E18.8)") 0.0D0,(A1(I),I=1,3)
         DO NTR=1,NTRAMOS
            DO IX=1,NDIV(NTR)
               X=COORD(NTR,1,1)+REAL(IX,KIND(0.0D0))*(COORD(NTR+1,1,1) &
                 -COORD(NTR,1,1))/REAL(NDIV(NTR),KIND(0.0D0))
               DO ID=1,3
                  CALL Desloc(NTR,IN,ID,2,IM,X,DESL)
                  A1(ID)=DESL
               END DO
               WRITE(15,"(F12.4,2X,3E18.8)") X,(A1(I),I=1,3)
            END DO
         END DO
      END DO
     END DO
   ELSE
     WRITE(15,*) 'Autovalores da equação caracteristica negativos'
   END IF
   CLOSE(15)
END SUBROUTINE Resul_din

SUBROUTINE Resul_ins
   USE geral
```

```
   REAL(KIND(0.0D0)) :: X, A1(3), DESL
   INTEGER :: NTR
   OPEN(16,FILE='RESINS.ASC',STATUS='UNKNOWN')
     WRITE(16,*) 'Cargas Criticas da Estrutura'
     WRITE(16,*) -1.0D0*REAL(ALPHAI(1:NMODOS)/BETAI(1:NMODOS))
     DO IM=1,NMODOS
      WRITE(16,"(' MODO N° ',I3)") IM
      DO IN=1,NNOS
         WRITE(16,"(' Nó N° ',I3)") IN
         WRITE(16,"(5X,'x',17X,'Dx',16X,'Dy',16X,'Dz')")
         DO IC=1,3
            CALL Desloc(1,IN,IC,3,IM,0.0D0,DESL)
            A1(IC)=DESL
         END DO
         WRITE(16,"(F12.4,2X,3E18.8)") 0.0D0,(A1(I),I=1,3)
         DO NTR=1,NTRAMOS
            DO IX=1,NDIV(NTR)
               X=COORD(NTR,1,1)+REAL(IX,KIND(0.0D0))*(COORD(NTR+1,1,1) &
                 -COORD(NTR,1,1))/REAL(NDIV(NTR),KIND(0.0D0))
               DO ID=1,3
                  CALL Desloc(NTR,IN,ID,3,IM,X,DESL)
                  A1(ID)=DESL
               END DO
               WRITE(16,"(F12.4,2X,3E18.8)") X,(A1(I),I=1,3)
            END DO
         END DO
      END DO
     END DO
   CLOSE(16)
END SUBROUTINE Resul_ins

SUBROUTINE Desloc(NTR,NNO,ID,IDAN,IMOD,X,DESL)          !IDAN (EST=1 DIN=2 INS=3)
   USE geral
   INTEGER, INTENT(IN) :: NTR, NNO, ID, IDAN, IMOD
   REAL(KIND(0.0D0)), INTENT(IN) :: X
   REAL(KIND(0.0D0)), INTENT(OUT) :: DESL
   INTEGER :: IAUX, I
   REAL(KIND(0.0D0)) :: RL, RM, RN, RI, FI(4), Q(4)
   DESL=0.
   IF (NTR.EQ.1) THEN
      IAUX=0
      ELSE
      IAUX=6*NNOS*(3*(NTR-1)+SUM(MPT(1:NTR-1)))
   END IF
   RL=COORD(NTR+1,1,1)-COORD(NTR,1,1)
   RM=REAL(MPT(NTR),KIND(0.0D0))
   RN=RM*X/RL
   I=INT(RN)
   RI=REAL(I,KIND(0.0D0))
   FI(1)=(RI+1-RN)**3/6.
   FI(2)=(1.+3.*(RI+1-RN)+3.*(RI+1-RN)**2-3.*(RI+1-RN)**3)/6.
   FI(3)=(1.+3.*(RN-RI)+3.*(RN-RI)**2-3.*(RN-RI)**3)/6.
   FI(4)=(RN-RI)**3/6.
   DO J=1,4
     SELECT CASE (IDAN)
      CASE (1)
        IF (.NOT.(X.EQ.RL .AND. J.EQ.4)) THEN
         Q(J)=UG(IAUX+6*NNOS*(I+J-1)+6*(NNO-1)+ID)
         DESL=DESL+FI(J)*Q(J)
        END IF
      CASE (2)
        IF (.NOT.(X.EQ.RL .AND. J.EQ.4)) THEN
         Q(J)=RMODO(IAUX+6*NNOS*(I+J-1)+6*(NNO-1)+ID,IMOD)
         DESL=DESL+FI(J)*Q(J)
        END IF
      CASE (3)
        IF (.NOT.(X.EQ.RL .AND. J.EQ.4)) THEN
         Q(J)=RMODOI(IAUX+6*NNOS*(I+J-1)+6*(NNO-1)+ID,IMOD)
         DESL=DESL+FI(J)*Q(J)
        END IF
     END SELECT
   END DO
END SUBROUTINE Desloc
```