ANEXO: PROGRAMAS EM MATLAB

Apresenta-se, a seguir, listagens de implementação, em linguagem MATLAB, de alguns programas computacionais para análise de Chapas, Placas e Cascas.

Textos precedidos pelo símbolo % são comentários, ignorados pelos programas, ou linhas que não se aplicam ao exemplo em questão.

Os exemplos podem, também, ser adaptados, com os devidos cuidados, para uso dos programas livres SCILAB e OCTAVE, similares ao MATLAB.

A.1 CHAPAS: ELEMENTO FINITO TRIANGULAR FUNÇÕES LINEARES

Programa para análise estática linear de uma chapa em Estado Plano de Tensões, discretizada com elementos finitos triangulares, funções de interpolação lineares, resultando tensões constantes no interior dos elementos.

Trata-se de uma chapa retangular de espessura unitária, altura 2 m e comprimento 4 m, módulo de elasticidade 100000 Pa, coeficiente de Poisson 0,3. A chapa está em balanço, fixa na extremidade esquerda e livre na direita, onde é solicitada por cargas axiais que perfazem uma força normal 1000 N. Ela é discretizada em 10 elementos triangulares.

```
%nome do arquivo MEF_chapa_elemento_triangular.m
%programa de EF para chapas planas com elementos triangulares
%funções de forma lineares - tensões constantes
% Prof Reyolando Brasil - abril/2016
%
clc
clear
%entrada de dados
%
%dimensões do problema
%
nno=10;%número de nós
nglpn=2;%número de graus de liberdade por nó
nds=nno*nglpn; %número de deslocamentos do sistema
nel=8;%número de elementos
nnel=3; %número de nós por elemento
ndpel=nnel*nglpn;%número de deslocamentos por elemento
%
%coordenadas X Y dos nós
%
gcoord=[0.0 0.0;0.0 1.0;1.0 0.0;1.0 1.0;2.0 0.0;
   2.0 1.0;3.0 0.0;3.0 1.0;4.0 0.0;4.0 1.0];
      %
```

```
%conectividade por elemento (numeração dos nós de cada elemento)
%
nodel=[1 3 4;1 4 2;3 5 6;3 6 4;5 7 8;5 8 6;7 9 10;7 10 8];
%
% matriz de número de graus de liberdade por nó
%
LN=zeros(nno,nglpn);
%
%condições de contorno (n. de nó restrito e direções restritas ou livres)
%
LN(1,:)=[1 1];
LN(2,:)=[1 0];
%
%dados físicos dos elementos (esp., mód. elasticidade e coef.de Poisson)
%
t=1; EM=100000.0; nu=0.3;
%
%determinação das matrizes
%
%matriz LN
%
ngl=0;ngr=nds;
for i=1:nno
   for j=1:nglpn
      if (LN(i,j)==0)
         ngl=ngl+1;
         LN(i,j)=ngl;
      else
         LN(i,j)=ngr;
         ngr=ngr-1;
      end
   end
```

```
end
nr=nds-ngl;
%
%inicialização de matrizes e vetores
%
K=zeros(nds,nds);
p=zeros(nds,1);
P=zeros(nds,1);
E=zeros(3,3);
Tens=zeros(nel,3);
q=zeros(6,1);
%
% vetor de carregamento P
%
P(LN(9,1))=500;
P(LN(10,1))=500;
%
%introduzir aqui recalques de apoio
%
%matriz constitutiva Estado Plano de Tensões
%
E=EM/(1-nu^2)*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
%
%matriz constitutiva Estado Plano de Deformações
%
%E=EM*(1-nu)/(1+nu)/(1-2*nu)*[1 nu/(1-nu) 0;
%      nu/(1-nu) 1 0;0 0 (1-2*nu)/2/(1-nu)];
%
%matrizes de rigidez
%
for iel=1:nel
   for j=1:nnel
```

```
        nd(j)=nodel(iel,j);
    end
    xa=gcoord(nd(1),1);ya=gcoord(nd(1),2);
    xb=gcoord(nd(2),1);yb=gcoord(nd(2),2);
    xc=gcoord(nd(3),1);yc=gcoord(nd(3),2);
%
%área do triângulo
%
    am=[1 xa ya;1 xb yb;1 xc yc];
    A=0.5*det(am);
%
% matriz B=LN
%
    B=1/2/A*[yb-yc 0 yc-ya 0 ya-yb 0;0 xc-xb 0 xa-xc 0 xb-xa;...
        xc-xb yb-yc xa-xc yc-ya xb-xa ya-yb];
%
% matriz de rigidez do elemento k
%
    k=A*t*B'*E*B;
%
%soma na matriz de rigidez do sistema
%
    kl=0;
    for n=1:nnel
        kl=kl+1;
        d(kl)=LN(nd(n),1);
        kl=kl+1;
        d(kl)=LN(nd(n),2);
    end
    for i=1:ndpel
        for j=1:ndpel
            K(d(i),d(j))=K(d(i),d(j))+k(i,j);
```

```
      end
   end
end
%
% solução do sistema: cálculo dos deslocamentos
%
disp('deslocamentos')
p(1:ngl)=K(1:ngl,1:ngl)\(P(1:ngl)-K(1:ngl,ngl+1:nds)*p(ngl+1:nds))
%
%cálculo das reações de apoio
%
disp('Esforços Nodais inclusive reações de apoio')
P(ngl+1:nds)=K(ngl+1:nds,1:ngl)*p(1:ngl)+K(ngl+1:ngl+1,ngl+1:ngl+1)...
   *p(ngl+1:nds)
%
%tensões
%
disp('tensões nos elementos')
disp('sigma_x, sigma_y, tau_xy')
for iel=1:nel
   for j=1:nnel
      nd(j)=nodel(iel,j);
   end
   xa=gcoord(nd(1),1);ya=gcoord(nd(1),2);
   xb=gcoord(nd(2),1);yb=gcoord(nd(2),2);
   xc=gcoord(nd(3),1);yc=gcoord(nd(3),2);
%
%área do triângulo
%
   am=[1 xa ya;1 xb yb;1 xc yc];
   A=0.5*det(am);
%
```

```
% matriz B=LN
%
   B=1/2/A*[yb-yc 0 yc-ya 0 ya-yb 0;0 xc-xb 0 xa-xc 0 xb-xa;...
      xc-xb yb-yc xa-xc yc-ya xb-xa ya-yb];
%
   kl=0;
   for n=1:nnel
      kl=kl+1;
      d(kl)=LN(nd(n),1);
      kl=kl+1;
      d(kl)=LN(nd(n),2);
   end
   for i=1:ndpel
      q(i)=p(d(i));
   end
   tau=E*B*q;
   Tens(iel,:)=tau';
end
```

A.2 CHAPAS: ELEMENTO FINITO RETANGULAR (ARGYRIS), ESTÁTICA

Programa para análise estática linear de uma chapa em Estado Plano de Tensões, discretizada com elementos finitos retangulares, funções de interpolação quadráticas, resultando tensões variáveis no interior dos elementos.

Trata-se de uma chapa retangular horizontal de espessura unitária, altura 2 m e comprimento 4 m, módulo de elasticidade 100000, coeficiente de Poisson 0,3. A chapa está em balanço, fixa na extremidade esquerda e livre na direita, onde é solicitada por carga vertical de 1000.

O programa está equipado com um gerador de malha simples para facilitar a entrada de dados. A chapa é dividida em 16 elementos na direção do comprimento e em 4 elementos na direção da altura, perfazendo 64 elementos finitos quadrados de 0,5 m x 0,5 m.

```
%nome do arquivo MEF_chapa_elemento_retangular_ger.m
%programa de EF para chapas planas com elementos retangulares
%Baseado em Argyris 1954
% Prof Reyolando Brasil - abril/2018
%
clc
clear
%entrada de dados
%
%gerador de malha elemento retangular
%Prof. Reyolando Brasil - abril 2018
%dimensoes do retangulo
compr=8;
alt=2;
%divisoes
ndx=16;ndy=4;
nno=(ndx+1)*(ndy+1);%numero de nos
dx=compr/ndx;dy=alt/ndy;
%geracao das coordenadas
gcoord=zeros(nno,2);
```

```
x=0;y=0;k=0;
for i=1:ndx+1
   for j=1:ndy+1
      k=k+1;
      gcoord(k,1)=x;gcoord(k,2)=y;
      y=y+dy;
   end
   y=0;
   x=x+dx;
end
%
%geracao da conectividade dos elementos
%
nel=ndx*ndy;%numero de elementos
nodel=zeros(nel,4);
kel=0;kaux=0;
for i=1:ndx
   for j=1:ndy
      kaux=kaux+1;
      kel=kel+1;
      nodel(kel,1)=kaux+ndy+2;nodel(kel,2)=kaux+1;
      nodel(kel,3)=kaux;nodel(kel,4)=kaux+ndy+1;
   end
   kaux=kaux+1;
end
%dimensao do problema
%
nglpn=2;%numero de graus de liberdade por no
nds=nno*nglpn; %numero de deslocamentos do sistema
nnel=4; %numero de nos por elemento
ndpel=nnel*nglpn;%numero de deslocamentos por elemento
%
```

```
% matriz de numero de graus de liberdade por no
%
LN=zeros(nno,nglpn);
%
%condicoes de contorno (n. de no restrito e direções restritas ou livres)
%engaste de viga em balanco
for i=k-ndy:k
   LN(i,:)=[-1 -1];
end
%
%dados fisicos dos elementos (esp., modulo elasticidade e coef.de Poisson)
%
t=1; EM=1000000.0; nu=0.3;
EL=EM/(1-nu*nu);G=EM/2/(1+nu);
E=EL*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
%
%determinação das matrizes
%
%matriz LN
%
ngl=0;
for i=1:nno
   for j=1:nglpn
      if  LN(i,j)==0
         ngl=ngl+1;
         LN(i,j)=ngl;
      end
   end
end
ngr=ngl;
for i=1:nno
   for j=1:nglpn
```

```
        if LN(i,j)<0
            ngr=ngr+1;
            LN(i,j)=ngr;
        end
    end
end
%
%inicializacao de matrizes e vetores
%
K=zeros(nds,nds);
p=zeros(nds,1);
P=zeros(nds,1);
Tens=zeros(nel,3);
q=zeros(8,1);
%
% vetor de carregamento P
%carga vertical na extremidade livre de viga em balanco
V=-1000;
for i=1:ndy+1
    P(LN(i,2))=V/(ndy+1);
end
%
%introduzir aqui recalques de apoio
%
%matrizes de rigidez dos elementos
%
for iel=1:nel
    for j=1:nnel
        nd(j)=nodel(iel,j);
    end
    xa=gcoord(nd(1),1);xb=gcoord(nd(2),1);
    yb=gcoord(nd(2),2);yc=gcoord(nd(3),2);
```

```
%
%dimensoes do retangulo
%
   a=(xa-xb)/2;b=(yb-yc)/2;
%
%constantes
   c1=EL*t*b/3/a;c2=c1/2;c3=EL*t*nu/4;
   c4=G*t*a/3/b;c5=c4/2;c6=G*t/4;
%
   kd(1,1)=c1;kd(1,2)=c3;kd(1,3)=-c1;kd(1,4)=c3;kd(1,5)=-c2;k-
d(1,6)=-c3;kd(1,7)=c2;kd(1,8)=-c3;
   kd(2,2)=c1;kd(2,3)=-c3;kd(2,4)=c2;kd(2,5)=-c3;kd(2,6)=-c2;kd(2,7)=c3;kd(2,8)=-c1;
   kd(3,3)=c1;kd(3,4)=-c3;kd(3,5)=c2;kd(3,6)=c3;kd(3,7)=-c2;kd(3,8)=c3;
   kd(4,4)=c1;kd(4,5)=-c3;kd(4,6)=-c1;kd(4,7)=c3;kd(4,8)=-c2;
   kd(5,5)=c1;kd(5,6)=c3;kd(5,7)=-c1;kd(5,8)=c3;
   kd(6,6)=c1;kd(6,7)=-c3;kd(6,8)=c2;
   kd(7,7)=c1;kd(7,8)=-c3;
   kd(8,8)=c1;
%
   ks(1,1)=c4;ks(1,2)=c6;ks(1,3)=c5;ks(1,4)==-6c;ks1(5,)-
-c5;ks(1,6)=-c6;ks(1,7)=-c4;ks(1,8)=c6;
   ks(2,2)=c4;ks(2,3)=c6;ks(2,4)=-c4;ks(2,5)=-c6;ks(2,6)=-c5;ks(2,7)=-c6;ks(2,8)=c5;
   ks(3,3)=c4;ks(3,4)=-c6;ks(3,5)=-c4;ks(3,6)=-c6;ks(3,7)=-c5;ks(3,8)=c6;
   ks(4,4)=c4;ks(4,5)=c6;ks(4,6)=c5;ks(4,7)=c6;ks(4,8)=-c5;
   ks(5,5)=c4;ks(5,6)=c6;ks(5,7)=c5;ks(5,8)=-c6;
   ks(6,6)=c4;ks(6,7)=c6;ks(6,8)=-c4;
   ks(7,7)=c4;ks(7,8)=-c6;
   ks(8,8)=c4;
%
   k=kd+ks;
%
%simetria
```

```
%
    for i=2:8
        for j=1:i-1
            k(i,j)=k(j,i);
        end
    end
%
%soma na matriz de rigidez do sistema
%
    kl=0;
    for n=1:nnel
        kl=kl+1;
        d(kl)=LN(nd(n),1);
        kl=kl+1;
        d(kl)=LN(nd(n),2);
    end
    for i=1:ndpel
        for j=1:ndpel
            K(d(i),d(j))=K(d(i),d(j))+k(i,j);
        end
    end
end
%
% solucao do sistema: calculo dos deslocamentos
%
disp('deslocamentos')
p(1:ngl)=K(1:ngl,1:ngl)\(P(1:ngl)-K(1:ngl,ngl+1:nds)*p(ngl+1:nds));
disp(p)
%
%calculo das reações de apoio
%
disp('Esforços Nodais inclusive reacoes de apoio')
```

```
P(ngl+1:nds)=K(ngl+1:nds,1:ngl)*p(1:ngl)+K(ngl+1:ngl+1,ngl+1:ngl+1)...
   *p(ngl+1:nds);
disp(P)
%
%
%Tensões
%
disp('Tensoes no pto central dos elementos')
disp('sigma_x,    sigma_y,    tau_xy')
for iel=1:nel
   for j=1:nnel
      nd(j)=nodel(iel,j);
   end
   xa=gcoord(nd(1),1);xb=gcoord(nd(2),1);
   yb=gcoord(nd(2),2);yc=gcoord(nd(3),2);
%
%  dimensões do retangulo
%
   a=(xa-xb)/2;b=(yb-yc)/2;
%
%  constantes
%
   ca=1/4/a;cb=1/4/b;
%  matriz B=L*N calculada no centro do elemento x=y=0
%
   B=[ca 0 -ca 0 -ca 0 ca 0;0 cb 0 cb 0 -cb 0 -cb;cb ca cb -ca -cb -ca -cb ca];
%
   kl=0;
   for n=1:nnel
      kl=kl+1;
      d(kl)=LN(nd(n),1);
      kl=kl+1;
```

```
        d(kl)=LN(nd(n),2);
    end
    for i=1:ndpel
        q(i)=p(d(i));
    end
    tau=E*B*q;
    Tens(iel,:)=tau';
end
disp(Tens)
```

A.3 CHAPAS: ELEMENTO FINITO RETANGULAR, FREQUÊNCIAS E MODOS

Programa para determinação de frequências e modos de vibração de uma chapa discretizada com elementos finitos retangulares, com matriz de massas não consistente.

Considere-se a mesma chapa quadrada da Fig. 1.24, de espessura, comprimento e largura unitários. Também se adota módulo de elasticidade 15, coeficiente de Poisson 0,25 e densidade 1.

A chapa será dividida em somente um elemento retangular, como mostrado na Figura 1.24.

Pede-se determinar as duas frequências de vibração livre e respectivos modos de vibração.

```
%nome do arquivo MEF_chapa_elemento_retangular_FREQ.m
%programa de EF para chapas planas com elementos retangulares
%frequencias e modos de vibracao
%Baseado em Argyris 1954
% Prof Reyolando Brasil - janeiro 2019
%
clc
clear
%entrada de dados
%
%dimensoes do problema
%
nno=4;%numero de nós
nglpn=2;%numero de graus de liberdade por no
nds=nno*nglpn; %numero de deslocamentos do sistema
nncr=4;%numero de nós com restricoes
nel=1;%numero de elementos
nnel=4; %numero de nos por elemento
ndpel=nnel*nglpn;%numero de deslocamentos por elemento
%
```

```
%coordenadas X Y dos nos
%
gcoord=[0 0;1 0;1 1;0 1];
%
%conectividade por elemento (n. dos nos de cada elemento)
%
nodel=[1 2 3 4];
%
% matriz de numero de graus de liberdade por no
%
LN=zeros(nno,nglpn);
%
%condicoes de contorno (n. de no restrito e direcoes restritas ou livres)
%
LN(1,:)=[-1 -1];
LN(2,:)=[-1 -1];
LN(3,:)=[0 -1];
LN(4,:)=[0 -1];
%
%dados fisicos dos elementos: esp., modulo elasticidade, coef.de
%Poisson,densidade
%
t=1; EM=15;nu=0.25;rho=1;
EL=EM/(1-nu*nu);G=EM/2/(1+nu);
E=EL*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
%
%determinacao das matrizes
%
%matriz LN
%
ngl=0;
for i=1:nno
```

```
    for j=1:nglpn
        if  LN(i,j)==0
            ngl=ngl+1;
            LN(i,j)=ngl;
        end
    end
end
ngr=ngl;
for i=1:nno
    for j=1:nglpn
        if LN(i,j)<0
            ngr=ngr+1;
            LN(i,j)=ngr;
        end
    end
end
%
%inicializacao de matrizes e vetores
%
K=zeros(nds,nds);
M=zeros(nds,nds);
%
%matrizes de rigidez e massa dos elementos
%
for iel=1:nel
    for j=1:nnel
        nd(j)=nodel(iel,j);
    end
    xa=gcoord(nd(1),1);xb=gcoord(nd(2),1);
    yb=gcoord(nd(2),2);yc=gcoord(nd(3),2);
%
%dimensoes do retangulo
```

```
%
   a=(xa-xb)/2;b=(yb-yc)/2;
%
%constantes
   c1=EL*t*b/3/a;c2=c1/2;c3=EL*t*nu/4;
   c4=G*t*a/3/b;c5=c4/2;c6=G*t/4;
%
   kd(1,1)=c1;kd(1,2)=c3;kd(1,3)=-c1;kd(1,4)=c3;kd(1,5)=-c2;k-
d(1,6)=-c3;kd(1,7)=c2;kd(1,8)=-c3;
   kd(2,2)=c1;kd(2,3)=-c3;kd(2,4)=c2;kd(2,5)=-c3;kd(2,6)=-c2;kd(2,7)=c3;kd(2,8)=-c1;
   kd(3,3)=c1;kd(3,4)=-c3;kd(3,5)=c2;kd(3,6)=c3;kd(3,7)=-c2;kd(3,8)=c3;
   kd(4,4)=c1;kd(4,5)=-c3;kd(4,6)=-c1;kd(4,7)=c3;kd(4,8)=-c2;
   kd(5,5)=c1;kd(5,6)=c3;kd(5,7)=-c1;kd(5,8)=c3;
   kd(6,6)=c1;kd(6,7)=-c3;kd(6,8)=c2;
   kd(7,7)=c1;kd(7,8)=-c3;
   kd(8,8)=c1;
%
   ks(1,1)=c4;ks(1,2)=c6;ks(1,3)=c5;ks(1,4)==-6c;ks1(5,)-
-c5;ks(1,6)=-c6;ks(1,7)=-c4;ks(1,8)=c6;
   ks(2,2)=c4;ks(2,3)=c6;ks(2,4)=-c4;ks(2,5)=-c6;ks(2,6)=-c5;ks(2,7)=-c6;ks(2,8)=c5;
   ks(3,3)=c4;ks(3,4)=-c6;ks(3,5)=-c4;ks(3,6)=-c6;ks(3,7)=-c5;ks(3,8)=c6;
   ks(4,4)=c4;ks(4,5)=c6;ks(4,6)=c5;ks(4,7)=c6;ks(4,8)=-c5;
   ks(5,5)=c4;ks(5,6)=c6;ks(5,7)=c5;ks(5,8)=-c6;
   ks(6,6)=c4;ks(6,7)=c6;ks(6,8)=-c4;
   ks(7,7)=c4;ks(7,8)=-c6;
   ks(8,8)=c4;
%
   k=kd+ks;
%
%simetria
%
   for i=2:8
```

```
        for j=1:i-1
            k(i,j)=k(j,i);
        end
    end
%
%matriz de massa simplificada do elemento m
m=rho*a*b/4*eye(8)
%
%soma na matriz de rigidez e de massa do sistema
%
    kl=0;
    for n=1:nnel
        kl=kl+1;
        d(kl)=LN(nd(n),1);
        kl=kl+1;
        d(kl)=LN(nd(n),2);
    end
    for i=1:ndpel
        for j=1:ndpel
            K(d(i),d(j))=K(d(i),d(j))+k(i,j);
            M(d(i),d(j))=M(d(i),d(j))+m(i,j);
        end
    end
end
%
KLL=K(1:ngl,1:ngl)
MLL=M(1:ngl,1:ngl)
%
% solução do problema de autovalores e autovetores
%
[v,d]=eig(KLL,MLL);
%
```

```
disp('Frequencias, em rad/s')
for i=1:ngl
    sqrt(d(i,i))
end
%
% modos de vibracao
%
for i=1:ngl
    const=v(1,i);
    for j=1:ngl
        v(j,i)=v(j,i)/const;
    end
end
disp('modos de vibracao')
```

A.4 PLACAS RETANGULARES: DIFERENÇAS FINITAS, ESTÁTICA

Programa para análise estática linear de uma placa fina, discretizada pelo Método das Diferenças Finitas.

Trata-se de uma placa retangular de espessura 2 cm, comprimento 3 m, largura 2 m, módulo de elasticidade 200 GPa, coeficiente de Poisson 0,3. A placa engastada nas faces paralelas ao comprimento e simplesmente apoiada nas paralelas à largura. O carregamento é uniformemente distribuído, 2 kN/m². A malha adotada tem 4 divisões em ambas as direções.

```
% placas retangular 3x2m
% diferenças finitas
% Prof. Reyolando Brasil 22 abril 2017
%
clc
clear
%
%propriedade físicas
%
a=3;b=2;%dimensões do retângulo
EM=200e6;t=0.02;nu=0.3;p=2;
D=EM*t^3/(12*(1-nu^2));pdD=p/D;
%
%geração da malha
%
nd=4;
np1=nd+1;
np3=nd+3;%número de nós da malha expandida em cada direção
neq=(nd-1)^2;%número de equações a resolver
hx=a/nd;hy=b/nd;hx4=hx*hx*hx*hx;hy4=hy*hy*hy*hy;hx2hy2=hx*hx*hy*hy;
%
%inicialização
%
```

```
A=zeros(np3*np3,np3*np3);%número de nós da malha = np3*np3
B=zeros(np3*np3,1);
kk=zeros(neq,1);
aa=zeros(neq,neq);
bb=zeros(neq,1);
%
%geração das equações de diferenças finitas para os nós internos
%
m=0;
for i=3:np1 % número da linha
   k=(i-1)*np3+2;
   for j=3:np1 %número da coluna
      k=k+1;m=m+1;kk(m,1)=k;%número do nó
      A(k,k-2)=1/hx4;
      A(k,k-1)=-4/hx4-4/hx2hy2;
      A(k,k+1)=-4/hx4-4/hx2hy2;
      A(k,k+2)=1/hx4;
      A(k,k)=6/hx4+6/hy4+8/hx2hy2;
      A(k+2*np3,k)=1/hy4;
      A(k+np3,k)=-4/hy4-4/hx2hy2;
      A(k-np3,k)=-4/hy4-4/hx2hy2;
      A(k-2*np3,k)=1/hy4;
      A(k+np3,k+1)=2/hx2hy2;
      A(k+np3,k-1)=2/hx2hy2;
      A(k-np3,k+1)=2/hx2hy2;
      A(k-np3,k-1)=2/hx2hy2;
      B(k,1)=pdD;
   end
end
%
%condições de contorno: isup, iinf, iesq, idir
%
```

```
%código: -1=apoiado; 1=engastado
%
isup=1;iinf=-1;iesq=-1;idir=-1;
%
%bordo superior
%
k=2*np3+2;
for j=3:np1
   k=k+1;
   A(k,k)=A(k,k)+isup*A(k-2*np3,k);
end
%
%bordo inferior
%
k=nd*np3+2;
for j=3:np1;
   k=k+1;
   A(k,k)=A(k,k)+iinf*A(k+2*np3,k);
end
%
%bordo esquerdo
%
k=2*np3+3;
for i=3:np1
    A(k,k)=A(k,k)+iesq*A(k,k-2);
    k=k+np3;
end
%bordo direito
%
k=3*np3-2;
for i=3:np1
    A(k,k)=A(k,k)+idir*A(k,k+2);
```

```
    k=k+np3;
end
%
%montagem sistema de equações
%
for i=1:neq
    for j=1:neq
        aa(i,j)=A(kk(i),kk(j));
    end
    bb(i)=B(kk(i));
end
%
%solução do sitema
%
w=aa\bb;
wmax=1000*max(w)%deslocamento máximo em mm
```

A.5 PLACAS RETANGULARES: DIFERENÇAS FINITAS, FREQUÊNCIAS

Apresenta-se, a seguir, listagem de uma implementação, em linguagem MATLAB, da determinação das frequências de vibração livre não amortecida de uma placa fina, discretizada pelo Método das Diferenças Finitas.

Trata-se de uma placa retangular de espessura 10 cm, comprimento 4 m, largura 2 m, módulo de elasticidade 25 GPa, coeficiente de Poisson 0,25, densidade de 2500 kg/m³. A placa é simplesmente apoiada em todo seu perímetro. A malha adotada tem 16 divisões em ambas as direções.

```
% placa_dif_fin_vibra_livre
% placas
% diferenças finitas
%vibracoes livres
% Prof. Reyolando Brasil 24 janeiro 2019
%
clc
clear
%
%propriedades físicas
%
a=4;b=2;%dimensões do retângulo
EM=250e8;t=0.1;nu=0.25;rho=2500;
D=EM*t^3/(12*(1-nu^2));
%
%geração da malha
%
nd=16;%número de divisões da malha
np1=nd+1;
np3=nd+3;%número de nós da malha expandida em cada direção
neq=(nd-1)^2;%número de equações a resolver
hx=a/nd;hy=b/nd;hx4=hx*hx*hx*hx;hy4=hy*hy*hy*hy;hx2hy2=hx*hx*hy*hy;
%
```

```
%inicialização
%
A=zeros(np3*np3,np3*np3);%número de nós da malha = np3*np3
kk=zeros(neq,1);
aa=zeros(neq,neq);
bb=zeros(neq,neq);
%
%geração das equações de diferenças finitas para os nós internos
%
m=0;
for i=3:np1 % número da linha
   k=(i-1)*np3+2;
   for j=3:np1 %número da coluna
      k=k+1;m=m+1;kk(m,1)=k;%número do nó
      A(k,k-2)=1/hx4;
      A(k,k-1)=-4/hx4-4/hx2hy2;
      A(k,k+1)=-4/hx4-4/hx2hy2;
      A(k,k+2)=1/hx4;
      A(k,k)=6/hx4+6/hy4+8/hx2hy2;
      A(k+2*np3,k)=1/hy4;
      A(k+np3,k)=-4/hy4-4/hx2hy2;
      A(k-np3,k)=-4/hy4-4/hx2hy2;
      A(k-2*np3,k)=1/hy4;
      A(k+np3,k+1)=2/hx2hy2;
      A(k+np3,k-1)=2/hx2hy2;
      A(k-np3,k+1)=2/hx2hy2;
      A(k-np3,k-1)=2/hx2hy2;
   end
end
%
%condições de contorno: isup, iinf, iesq, idir
%
```

```
%código: -1=apoiado; 1=engastado
%
isup=-1;iinf=-1;iesq=-1;idir=-1;
%
%bordo superior
%
k=2*np3+2;
for j=3:np1
    k=k+1;
    A(k,k)=A(k,k)+isup*A(k-2*np3,k);
end
%
%bordo inferior
%
k=nd*np3+2;
for j=3:np1;
    k=k+1;
    A(k,k)=A(k,k)+iinf*A(k+2*np3,k);
end
%
%bordo esquerdo
%
k=2*np3+3;
for i=3:np1
    A(k,k)=A(k,k)+iesq*A(k,k-2);
    k=k+np3;
end
%bordo direito
%
k=3*np3-2;
for i=3:np1
    A(k,k)=A(k,k)+idir*A(k,k+2);
```

```
    k=k+np3;
end
%
%montagem sistema de equações
%
for i=1:neq
    for j=1:neq
        aa(i,j)=A(kk(i),kk(j));
    end
end
%
%
%solução do problema de autovalores e autovetores
%
lambda=eig(aa);
disp('Frequencias, em rad/s')
format long
omega=sqrt(lambda*D/rho)
```