Elementos Finitos

Resortes

Fig. 2.1. The Spring Element

$$k = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$$

k: Matriz de rigidez del elemento

$$[K]\{U\} = \{F\}$$

K: Matriz de rigidez global

U: Dezplazamiento global de los nodos

F: Vector de fuerza global de los nodos

$$\{f\} = [k]\{u\}$$

u: Vector de dezplazamiento del elemento

f: Vector de fuerza del elemento

Problem 2.1:

Consider the spring system composed of two springs as shown in Fig. 2.4. Given $k_1 = 200\,\mathrm{kN/m}$, $k_2 = 250\,\mathrm{kN/m}$, and $P = 10\,\mathrm{kN}$, determine:

1. the global stiffness matrix for the system.
2. the displacements at node 2.
3. the reactions at nodes 1 and 3.
4. the force in each spring.

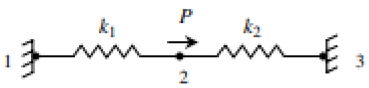


Fig. 2.4. Two-Element Spring System for Problem 2.1

1.

```
k=[200 250];
conex = [1 2;2 3];
nodos= 3;
f=[2 10]; %nodo y fuerza

elem = size(k,2);
K=zeros(nodos);

%Ensamble
for i=1:elem
    ke=k(i)*[1 -1;-1 1];
    m=conex(i,1);
    n=conex(i,2);
    K(m,m) = K(m,m) + ke(1,1);
    K(m,n) = K(m,n) + ke(1,2);
    K(n,m) = K(n,m) + ke(2,1);
    K(n,n) = K(n,n) + ke(2,2);
end
K
```

$$K = \begin{matrix} 200 & -200 & 0 \\ -200 & 450 & -250 \\ 0 & -250 & 250 \end{matrix}$$

2., 3.

```
%Fuerzas nodo

F=zeros(nodos,1);
F(f(1))=f(2);

%Desplazamientos nodo
u=zeros(nodos,1);

a=2; %simplificación de matriz
u(a)=K(a,a)\F(a);
F=K*u;

t1=table([1:nodos]',F(:,1),u(:,1),'VariableNames',{'Nodos','Fuerza','D
ezplazamiento'})
```

Nodos	Fuerza	Dezplazamiento
1	-4.4444	0
2	10	0.022222
3	-5.5556	0

4.

```
Ue=zeros(elem,2);
Fe=zeros(elem,2);
T=cell(elem,1);

for i=1:elem
    ke=k(i)*[1 -1;-1 1];
    ue=[u(conex(i,1)); u(conex(i,2))];
    fe=ke*ue;
    Ue(i,1)=ue(1);
    Ue(i,2)=ue(2);
    Fe(i,1)=fe(1);
    Fe(i,2)=fe(2);
    if fe(1)>0
        T{i}='Compresión';
    else
        T{i}='Tensión';
    end
end

t2=table([1:elem]',conex(:,1),conex(:,2),abs(Fe(:,1)),T,'VariableNames
',{'Elemento','Nodo1','Nodo2','Fuerza','Tipo'})
```

Elemento	Nodo1	Nodo2	Fuerza	Tipo
1	1	2	4.4444	{'Tensión' }
2	2	3	5.5556	{'Compresión'}

Problem 2.2:

Consider the spring system composed of four springs as shown in Fig. 2.5. Given $k = 170\,\mathrm{kN/m}$ and $P = 25\,\mathrm{kN}$, determine:

1. the global stiffness matrix for the system.
2. the displacements at nodes 2, 3, and 4.
3. the reaction at node 1.
4. the force in each spring.

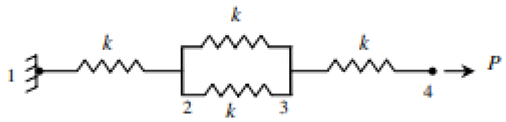


Fig. 2.5. Four-Element Spring System for Problem 2.2

1.

```
k=[170 170 170 170];
conex = [1 2;2 3;2 3;3 4];
nodos= 4;
f=[4 25]; %nodo y fuerza
a=2:4;
```

Se repite el código

$$K = \begin{matrix} 170 & -170 & 0 & 0 \\ -170 & 510 & -340 & 0 \\ 0 & -340 & 510 & -170 \\ 0 & 0 & -170 & 170 \end{matrix}$$

2., 3.

Nodos	Fuerza	Dezplazamiento
1	-25	0
2	0	0.14706
3	0	0.22059
4	25	0.36765

4.

Elemento	Nodo1	Nodo2	Fuerza	Tipo
1	1	2	25	{'Tensión'}
2	2	3	12.5	{'Tensión'}
3	2	3	12.5	{'Tensión'}
4	3	4	25	{'Tensión'}

Barras

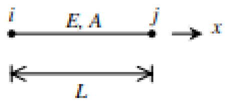


Fig. 3.1. The Linear Bar Element

$$k = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix}$$

k: Matriz de rigidez del elemento

$$[K]\{U\} = \{F\}$$

K: Matriz de rigidez global

U: Dezplazamiento global de los nodos

F: Vector de fuerza global de los nodos

$$\{f\} = [k]\{u\}$$

u: Vector de dezplazamiento del elemento

f: Vector de fuerza del elemento

Problem 3.1:

Consider the structure composed of three linear bars as shown in Fig. 3.5. Given $E = 70\,\mathrm{GPa}$, $A = 0.005\,\mathrm{m}^2$, $P_1 = 10\,\mathrm{kN}$, and $P_2 = 15\,\mathrm{kN}$, determine:

1. the global stiffness matrix for the structure.
2. the displacements at nodes 2, 3, and 4.
3. the reaction at node 1.
4. the stress in each bar.

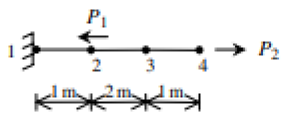


Fig. 3.5. Three-Bar Structure for Problem 3.1

1.

```
E=70e6;
L=[1 2 1];
A=0.005;
nodos= 4;
conex = [1 2;2 3;3 4];
f=[2 -10;4 15]; %nodo y fuerza

elem = size(L,2);
K=zeros(nodos);

for i=1:elem
    ke=E*A/L(i)*[1 -1;-1 1];
m=conex(i,1);
    n=conex(i,2);
    K(m,m) = K(m,m) + ke(1,1);
    K(m,n) = K(m,n) + ke(1,2);
    K(n,m) = K(n,m) + ke(2,1);
    K(n,n) = K(n,n) + ke(2,2);
end
K
```

K =

350000	-350000	0	0
-350000	525000	-175000	0
0	-175000	525000	-350000
0	0	-350000	350000

```
%Fuerzas nodo
F=zeros(nodos,1);
F(f(1,1))=f(1,2);
F(f(2,1))=f(2,2);

%Desplazamientos nodo
u=zeros(nodos,1);

a=2:4; %simplificación de matriz
u(a)=K(a,a)\F(a)
F=K*u
```

2.

```
u =

   1.0e-03 *

        0
   0.0143
   0.1000
   0.1429
```

3.

```
F =

   -5.0000
  -10.0000
   -0.0000
   15.0000
```

```
Ue=zeros(elem,2);
Fe=zeros(elem,2);
Se=zeros(elem,2);

for i=1:elem
    ke=E*A/L(i)*[1 -1;-1 1];
    ue=[u(conex(i,1)); u(conex(i,2))];
    fe=ke*ue;
    sigmae=fe/A;
    Ue(i,1)=ue(1);
    Ue(i,2)=ue(2);
    Fe(i,1)=fe(1);
    Fe(i,2)=fe(2);
    Se(i,1)=sigmae(1);
    Se(i,2)=sigmae(2);
end

t2=table([1:elem]',conex(:,1),conex(:,2),Fe(:,1),Se(:,1),'VariableName
s',{'Elemento','Nodo1','Nodo2','Fuerza','Esfuerzo'})
```

4.

Elemento	Nodo1	Nodo2	Fuerza	Esfuerzo
1	1	2	-5	-1000
2	2	3	-15	-3000
3	3	4	-15	-3000

Problem 3.3:

Consider the structure composed of a spring and a linear bar as shown in Fig. 3.6. Given $E = 200\,\mathrm{GPa}$, $A = 0.01\,\mathrm{m}^2$, $k = 1000\,\mathrm{kN/m}$, and $P = 25\,\mathrm{kN}$, determine:

1. the global stiffness matrix for the structure.
2. the displacement at node 2.
3. the reactions at nodes 1 and 3.
4. the stress in the bar.
5. the force in the spring.

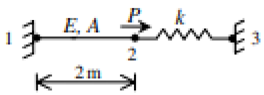


Fig. 3.6. Linear Bar with a Spring for Problem 3.3

1.

```
k=[200e6*0.01/2 1000];
f=[2 25];
nodos= 3;
conex = [1 2;2 3];
f=[2 25]; %nodo y fuerza

elem = size(k,2);
K=zeros(nodos);

for i=1:elem
    ke=k(i)*[1 -1;-1 1];
    m=conex(i,1);
    n=conex(i,2);
    K(m,m) = K(m,m) + ke(1,1);
    K(m,n) = K(m,n) + ke(1,2);
    K(n,m) = K(n,m) + ke(2,1);
    K(n,n) = K(n,n) + ke(2,2);
end
K
```

K =

```
     1000000    -1000000           0
    -1000000     1001000       -1000
           0       -1000        1000
```

2., 3.

```
%Fuerzas nodo
F=zeros(nodos,1);
F(f(1))=f(2);

%Desplazamientos nodo
u=zeros(nodos,1);

a=2; %simplificación de matriz
u(a)=K(a,a)\F(a);
F=K*u;

t1=table([1:nodos]',F(:,1),u(:,1),'VariableNames',{'Nodos','Fuerza','D
ezplazamiento'})
```

Nodos	Fuerza	Dezplazamiento
1	-24.975	0
2	25	2.4975e-05
3	-0.024975	0

```
Ue=zeros(elem,2);
Fe=zeros(elem,2);

for i=1:elem
    ke=k(i)*[1 -1;-1 1];
    ue=[u(conex(i,1)); u(conex(i,2))];
    fe=ke*ue;
    Ue(i,1)=ue(1);
    Ue(i,2)=ue(2);
    Fe(i,1)=fe(1);
    Fe(i,2)=fe(2);
end

sigma1=Fe(1,:)/0.01
f2=Fe(2,:)
```

4.

```
sigma1 =

   1.0e+03 *

  -2.4975    2.4975
```

5.

```
f2 =

    0.0250   -0.0250
```

Barras 3D

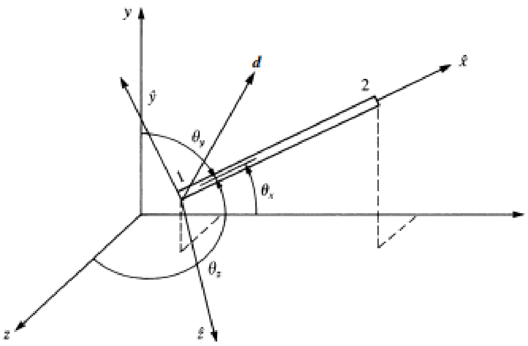


Figure 3–17 Bar in three-dimensional space

$$\hat{\mathbf{i}} \cdot \mathbf{i} = \frac{x_2 - x_1}{L} = C_x$$
$$\hat{\mathbf{i}} \cdot \mathbf{j} = \frac{y_2 - y_1}{L} = C_y$$
$$\hat{\mathbf{i}} \cdot \mathbf{k} = \frac{z_2 - z_1}{L} = C_z$$
$$L = [(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]^{1/2}$$
$$C_x = \cos\theta_x \qquad C_y = \cos\theta_y \qquad C_z = \cos\theta_z$$

$$\underline{k} = \frac{AE}{L}\left[\begin{array}{c|c} \underline{\lambda} & -\underline{\lambda} \\ \hline -\underline{\lambda} & \underline{\lambda} \end{array}\right]$$

$$\underline{\lambda} = \begin{bmatrix} C_x^2 & C_x C_y & C_x C_z \\ C_y C_x & C_y^2 & C_y C_z \\ C_z C_x & C_z C_y & C_z^2 \end{bmatrix}$$

For the space trusses shown in Figures P3–40 and P3–41, determine the nodal displacements and the stresses in each element. Let $E = 210$ GPa and $A = 10 \times 10^{-4}$ m^2 for all elements. Verify force equilibrium at node 1. The coordinates of each node, in meters, are shown in the figure. All supports are ball-and-socket joints.

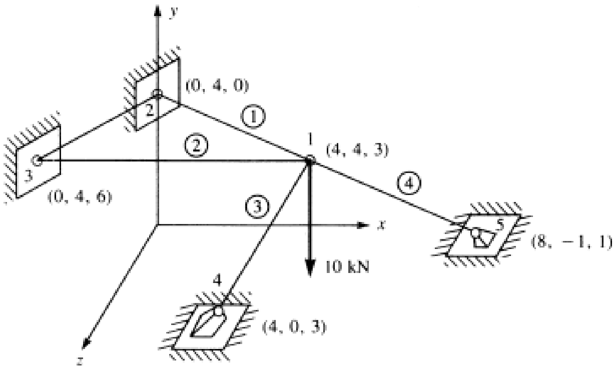


Figure P3–40

```
elem=4;
E(1:elem)=210*10^9;
area(1:elem)=10*10^-4; %misma área
nodos=[4 4 3 ; 0 4 0 ; 0 4 6 ; 4 0 3 ; 8 -1 1];
UnionNodos=[1 2 ;1 3 ; 1 4 ; 1 5];
%[dx1 dy1 dz1 dx2 dy2 dz2 dx3 dy3 dz3...](0 si está empotrado y 1 si
puede moverse)
Desplazamientos=[1 1 1 0 0 0 0 0 0 0 0 0 0 0 0];
%[Fx1 Fy1 Fz1 Fx2 Fy2 Fz2 Fx3 Fy3 Fz3...]
Fuerzas=[0 -10000 0 0 0 0 0 0 0 0 0 0 0 0 0];

L = zeros(1,elem);
Cx = zeros(1,elem);
Cy = zeros(1,elem);
Cz = zeros(1,elem);
LAMBDA = zeros(6,6);

for i = 1:elem
    indice = UnionNodos(i,:);
    P1 = [nodos(indice(1),1) nodos(indice(1),2) nodos(indice(1),3)];
    P2 = [nodos(indice(2),1) nodos(indice(2),2) nodos(indice(2),3)];
    L(i) = norm(P1-P2);
    Cx(i) = (P2(1) - P1(1))/ L(i);
    Cy(i) = (P2(2) - P1(2))/ L(i);
    Cz(i) = (P2(3) - P1(3))/ L(i);
    lambda = [Cx(i)^2 Cx(i)*Cy(i) Cx(i)*Cz(i) ; Cy(i)*Cx(i) Cy(i)^2
Cy(i)*Cz(i) ;...
           Cz(i)*Cx(i) Cz(i)*Cy(i) Cz(i)^2];
    LAMBDA(:,:,i) = [lambda -lambda ; -lambda lambda];
end

k = (E.*area)./L;
A = zeros(6,6);

%Ensamble de la matriz global
for i = 1:elem
    A(:,:,i) = k(i)*LAMBDA(:,:,i);
    %Dividimos la matriz A 6x6 en sub matrices de3x3
    j = UnionNodos(i,:);
    B(:,:,i) = mat2cell(A(:,:,i),[3 3],[3 3]);
```

```
    %Asignamos cada sub matriz segun indice
    C(j(1),j(1),i) = B(1,1,i);
    C(j(1),j(2),i) = B(1,2,i);
    C(j(2),j(1),i) = B(2,1,i);
    C(j(2),j(2),i) = B(2,2,i);
end
A
S = 3*size(nodos,1);
m = cell(S/3,S/3);

for i = 1:size(nodos,1)
    for j = 1:size(nodos,1)
        clear x
        x(:,:,:) = cell2mat(reshape(C(i,j,:),1,[],elem));
        m(i,j) = {sum(x,3)};

        %Si esta vacio se asigna 0
        if size(m{i,j}) == [0 0]
            m(i,j) = {zeros(3,3)};
        end
    end
end

MG = cell2mat(m)          %Convertimos la matriz global en un arreglo
numérico

%Reducir la matriz global
v = find(Desplazamientos==0);
MGR = MG;
MGR(v,:) = 0;
MGR(:,v) = 0;
indicefil = zeros(1,S);
indicecol = zeros(1,S);
for i = 1:S
    if MGR(i,:) == 0
        indicefil(i) = i;
    end
    if MGR(:,i) == 0
        indicecol(i) = i;
    end
end

MGR(indicefil~=0,:) = [];    %Eliminar filas y columnas de ceros para
tener la matriz global reducida
MGR(:,indicecol~=0) = []
Fuerzas(indicefil~=0) = [];  %Eliminar filas y columnas de ceros de
las fuerzas

%Desplazamientos de nodos
d = MGR\Fuerzas';
dfinal = zeros(S,1);
k = 1;

for i = 1:length(Desplazamientos)
    if Desplazamientos(i) == 0
        dfinal(i,1) = 0;
    else
        dfinal(i,1) = d(k);
        k = k+1;
    end
```

```
end

%Resultados

d2 = mat2cell(dfinal,3*ones(1,size(nodos,1)),1); %Dividimos dfinal en
paquetes de 3x1
Esfuerzos = zeros(1,elem);
Flocal = zeros(elem,6);
j = 1;
for i = 1:elem
    indice = UnionNodos(i,:);
    Esfuerzos(i) = (E(i)./L(i)) * [-Cx(i) -Cy(i) -Cz(i) Cx(i) Cy(i)
Cz(i)] * [d2{indice(1,1)} ; d2{indice(1,2)}];
    Flocal(i,:) = A(:,:,i)*[d2{indice(1,1)} ; d2{indice(1,2)}];
    j = j + 2;
end

Reacciones=reshape(MG*dfinal,[3,5]).'
dfinal = reshape(dfinal,[3,5]).'
Esfuerzos=Esfuerzos'
Flocal
```

Matriz global reducida

```
MGR =

   1.0e+07 *

    6.4891   -1.3913   -0.5565
   -1.3913    6.9892    0.6957
   -0.5565    0.6957    3.3023
```

Nodos

```
Reacciones =

   1.0e+04 *

   -0.0000   -1.0000    0.0000
    0.0271         0    0.0203
    0.1355         0   -0.1016
         0    0.7968         0
   -0.1626    0.2032    0.0813
```

```
dfinal =

   1.0e-03 *

   -0.0302   -0.1518    0.0269
         0         0         0
         0         0         0
         0         0         0
         0         0         0
```

Elementos

```
Flocal =

   1.0e+03 *

   -0.2709         0   -0.2032    0.2709         0    0.2032
   -1.3546         0    1.0160    1.3546         0   -1.0160
         0   -7.9681         0         0    7.9681         0
    1.6255   -2.0319   -0.8128   -1.6255    2.0319    0.8128
```

```
Esfuerzos =

   1.0e+06 *

   -0.3387
   -1.6933
   -7.9681
   -2.7261
```

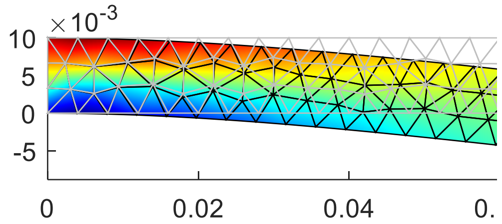
Carga aplicada en viga

```
% Malla
model = createpde();
l = 0.1;
h = 0.01;
R = [3; 4; 0; l; l; 0; 0; 0; h; h];
[dl, bt] = decsg(R); %Descompone en regiones
geometryFromEdges(model, dl);
generateMesh(model, 'GeometricOrder', 'linear', 'Hmax', 0.004);
[p, e, t] = meshToPet(model.Mesh);
edge = e(1, :);
t = t(1:3, :)';

nnodos = size(p, 2);
elem = size(t, 1);

% Parametros
E = 70e9;
nu = 0.33;
D = E / (1 - nu^2) * [ 1 nu 0;nu 1 0;0 0 (1 - nu)/2];

% Esfuerzos aplicados
q = [0; -10 / (0.1 * 0.01)];

K = zeros(2 * nnodos);
F = zeros(2 * nnodos, 1);

% Ensamble
for element = 1 : elem
    nodes = t(element, :);
    P = [ones(1, 3); p(:, nodes)];
    C = inv(P);
    area_of_element = abs(det(P))/2;
    diff_Phi = C(:, 2:3);

    B{element} = [];
    for i = 1 : 3
        b_e = [diff_Phi(i, 1) 0;0 diff_Phi(i, 2);diff_Phi(i, 2)
diff_Phi(i, 1)];
        B{element} = [B{element}, b_e];
    end

    Ke = B{element}' * D * B{element} * area_of_element;
    dofs = reshape([2 * nodes - 1; 2 * nodes], 1, 2 * numel(nodes));
    K(dofs, dofs) = K(dofs, dofs) + Ke;
end

% Condicion de frontera de Neumann
t_Neumann = [];
for e = 1 : elem
    nodes = t(e, :);
    I = p(2, nodes) == max(p(2, :));
    if(sum(I) == 2)
        t_Neumann = [t_Neumann; nodes(I)];
    end
end
```

```
for element = 1 : size(t_Neumann, 1)
    nodes = t_Neumann(element, :);
    dofs_Neumann = reshape([2 * nodes - 1; 2 * nodes], 1, 2 *
numel(nodes));
    P = p(:, nodes);
    length_of_element = norm(diff(P, 1, 2));
    H_mean = 1/2 * repmat(eye(2), 1, 2);
    Fe = H_mean' * q * length_of_element;
    F(dofs_Neumann) = F(dofs_Neumann) + Fe;
end

% Condición de frontera de Dirichlet
Dirichlet = edge(p(1, edge) == 0);
dofs_Dirichlet = [2 * Dirichlet - 1, 2 * Dirichlet];
K(dofs_Dirichlet, :) = 0;
K(dofs_Dirichlet, dofs_Dirichlet) = eye(numel(dofs_Dirichlet));
F(dofs_Dirichlet) = 0;

% Solucion
U = K \ F;
min(U(:))
displacements = [U(1 : 2 : end), U(2 : 2 : end)]';
magnification = 5e2;
p_new = p + magnification * displacements;

% Calculo de esfuerzos
S = zeros(1 * nnodos, 3);
node_occurences = zeros(1 * nnodos, 1);
for element = 1 : elem
    nodes = t(element, :);
    displacement_nodes_of_element = displacements(:, nodes);
    U_e = displacement_nodes_of_element(:);
    s = D * B{element} * U_e;
    S(nodes', :) = S(nodes', :) + repmat(s', 3, 1);
end

% Ocurrencia de nodos
for i = 1 : nnodos
    node_occurences(i) = numel(find(t == i));
end

S = S ./ repmat(node_occurences, 1, 3);

%Grafica
STRESS_COMPONENT = 1;
splot = S(:, STRESS_COMPONENT);

set(gcf, 'color', 'w')
colormap jet
hold on
trisurf(t, p_new(1, :), p_new(2, :), zeros(1, nnodos), splot,
'EdgeColor', 'k', 'FaceColor', 'interp');
trisurf(t, p(1, :), p(2, :), zeros(1, nnodos), 'EdgeColor',  0.75 * [1
1 1], 'FaceColor', 'none');
colorbar
view(2)
axis equal
axis tight
```

```
ans =

  -1.7681e-05
```

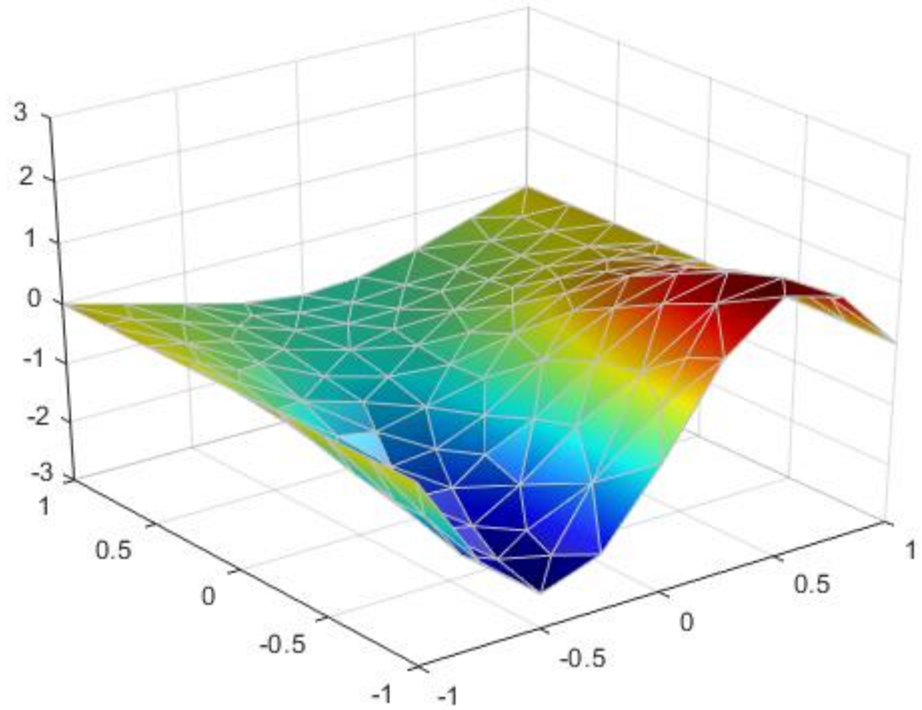
Desplazamiento mínimo



Ecuación de onda

```
% Malla
g = 'squareg';
[p, e, t] = initmesh(g, 'Hmax', 0.25);

e = e(1, :);
t = t(1 : 3, :)';
nnodes = size(p, 2);
elem = size(t, 1);

c = 1;
m = 1;

K = zeros(nnodes);
M = zeros(nnodes);

% Ensamble
for element = 1 : elem
    nodes = t(element, :);
    P = [ones(1, 3); p(:, nodes)];
    C = inv(P);
    area_of_element = abs(det(P)) / 2;
    grads_phis   = C(:, 2:3);
    xy_mean = mean(p(:, nodes), 2);
    Ke = grads_phis * c * grads_phis' * area_of_element;
    mean_of_phis   = [1/3; 1/3; 1/3];
    Me = m * (mean_of_phis * mean_of_phis') * area_of_element;
    K(nodes, nodes) = K(nodes, nodes) + Ke;
    M(nodes, nodes) = M(nodes, nodes) + Me;
end

% Condición de frontera de Dirichlet
Dirichlet = [1:4, 3, 12:18, 26:32];
K(Dirichlet, :) = 0;
M(Dirichlet, :) = 0;
M(Dirichlet, Dirichlet) = eye(numel(Dirichlet));

time = linspace(0.0, 10.0, 1e3);
A = [zeros(nnodes) eye(nnodes); -M \ K zeros(nnodes)];
u_0 = atan(cos(pi/2 * p(1, :)));
du_dt_0 = 3 * sin(pi * p(1, :)) .* exp(sin(pi / 2 * p(2, :)));
Z_initial = [u_0, du_dt_0];
dZ_dt = @(~, Z) A * Z;
[~, Z] = ode45(dZ_dt, time, Z_initial);
```

```
Z = Z(:, 1:nnodes);

%Grafica
set(gcf, 'color', 'w')
has_been_plot = false;
for z = Z'
    if ~has_been_plot
        has_been_plot = true;
        h = trisurf(t, p(1, :), p(2, :), z, 'EdgeColor', 0.75 * [1 1
1], 'FaceColor', 'interp');
        colormap(jet)
        light
        camlight left
        set(gca, 'Projection', 'perspective')
        axis([-1 1 -1 1 -3 3])
    else
       h.Vertices = [p(1, :)', p(2, :)', z];
       h.CData = z;
    end
    drawnow
end
```



Estructuras 2D

k: Matriz de rigidez del elemento

$$k = \frac{EA}{L}\begin{bmatrix} C^2 & CS & -C^2 & -CS \\ CS & S^2 & -CS & -S^2 \\ -C^2 & -CS & C^2 & CS \\ -CS & -S^2 & CS & S^2 \end{bmatrix}$$

$$[K]\{U\} = \{F\}$$

f: Fuerza en cada elemento

$$f = \frac{EA}{L}\begin{bmatrix} -C & -S & C & S \end{bmatrix}\{u\}$$

Consider the plane truss shown in Fig. 5.3. Given $E = 210$ GPa and $A = 1\times10^{-4}$ m^2, determine:

1. the global stiffness matrix for the structure.
2. the horizontal displacement at node 2.
3. the horizontal and vertical displacements at node 3.
4. the reactions at nodes 1 and 2.
5. the stress in each element.

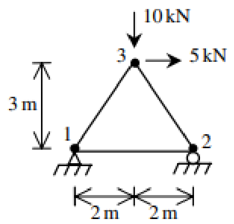


Fig. 5.3. Plane Truss with Three Elements for Example 5.1

```
nodos = [0 0;4 0;2 3];
conn = [1 2;2 3;3 1];

A=1e-4;
E=210e6;

p=[5 5 6 -10];

nn=size(nodos,1);
ndof=2*nn; %grados de libertad

f=zeros(ndof,1);
f(p(1))=p(2);
f(p(3))=p(4);

isol=[3 5 6]; %simplifica matriz en nodos libres

ne=size(conn,1);

K=zeros(ndof,ndof);
d=zeros(ndof,1);

for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    L = norm(nodos(n2,:)-nodos(n1,:));
    C = (nodos(n2,1)-nodos(n1,1))/L; %cos
    S = (nodos(n2,2)-nodos(n1,2))/L; %sin
```

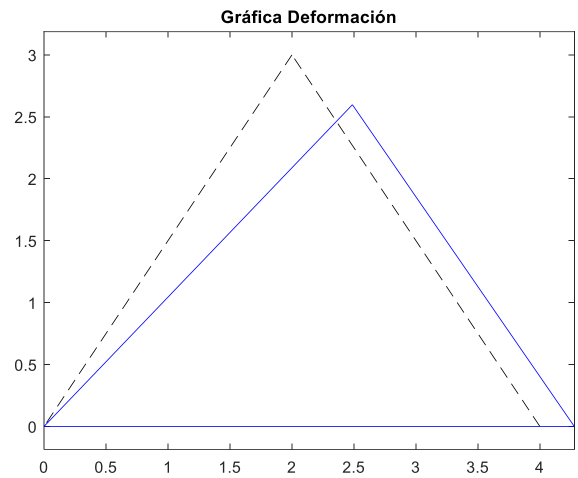
```
    ke = (A*E/L) * [C^2 C*S -C^2 -C*S; C*S S^2 -C*S -S^2;
                    -C^2 -C*S C^2 C*S; -C*S -S^2 C*S S^2];
    sctr = [2*n1-1 2*n1 2*n2-1 2*n2]; %posicion en matriz global
    K(sctr,sctr) = K(sctr,sctr) + ke;
end

d(isol) = K(isol,isol)\f(isol);

f = K*d;
sigma=zeros(ne,1);

for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    L = norm(nodos(n2,:)-nodos(n1,:));
    C = (nodos(n2,1)-nodos(n1,1))/L; %cos
    S = (nodos(n2,2)-nodos(n1,2))/L; %sin
    B = E/L*[-C -S C S];
    sctr = [2*n1-1 2*n1 2*n2-1 2*n2];
    sigma(e) = B*d(sctr);
end

clf
sclf=250; %escala deformacion

for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    x1 = nodos(n1,1); y1 = nodos(n1,2);
    x2 = nodos(n2,1); y2 = nodos(n2,2);
    u1 = d(2*n1-1); v1 = d(2*n1);
    u2 = d(2*n2-1); v2 = d(2*n2);
    plot([x1, x2],[y1, y2],'k--');
    plot([x1, x2]+sclf*[u1, u2],[y1, y2]+sclf*[v1,v2],'b');
    hold on
end
title('Gráfica Deformación')
axis equal

Fuerzas = reshape(f,[2,nn]).'
Desplazamientos = reshape(d,[2,nn]).'
sigma
```

```
K =

   1.0e+03 *

    7.0421    2.6882   -5.2500         0   -1.7921   -2.6882
    2.6882    4.0322         0         0   -2.6882   -4.0322
   -5.2500         0    7.0421   -2.6882   -1.7921    2.6882
         0         0   -2.6882    4.0322    2.6882   -4.0322
   -1.7921   -2.6882   -1.7921    2.6882    3.5842         0
   -2.6882   -4.0322    2.6882   -4.0322         0    8.0645
```

```
Fuerzas =

   -5.0000    1.2500
    0.0000    8.7500
    5.0000  -10.0000
```

```
Desplazamientos =

         0         0
    0.0011         0
    0.0020   -0.0016
```

```
sigma =

   1.0e+05 *

    0.5833
   -1.0516
   -0.1502
```



3.24 Determine the nodal displacements and the element forces for the truss shown in Figure P3–24. Assume all elements have the same AE.

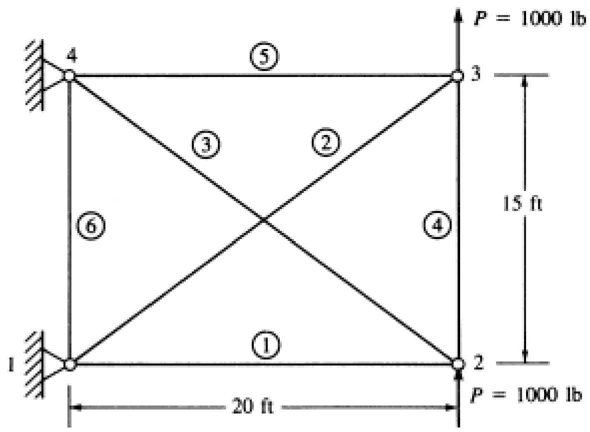


Figure P3–24

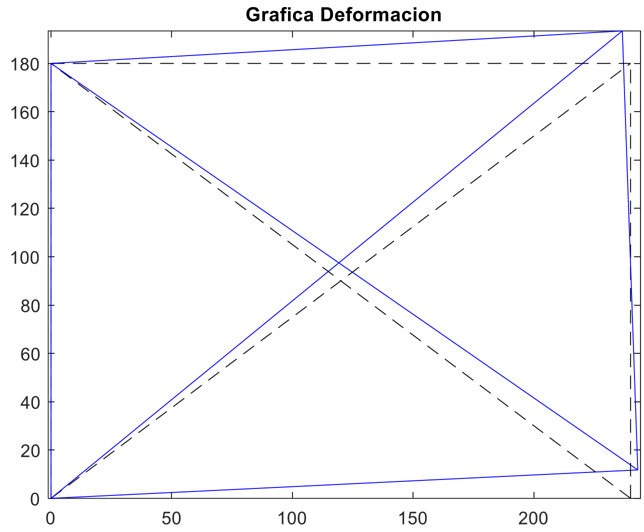
```
nodos = [0 0;20 0;20 15;0 15]*12;%pulgadas
conn = [1 2;1 3;2 4;2 3;4 3;1 4];

A=1;
E=10e6;
p=[6 1000];

nn=size(nodos,1);
ndof=2*nn; %grados de libertad
f=zeros(ndof,1);
f(p(1))=p(2);

isol=[3 4 5 6]; %simplifica matriz en nodos libres
ne=size(conn,1);
```

```
K=zeros(ndof,ndof);
d=zeros(ndof,1);

for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    L = norm(nodos(n2,:)-nodos(n1,:));
    C = (nodos(n2,1)-nodos(n1,1))/L; %cos
    S = (nodos(n2,2)-nodos(n1,2))/L; %sin
    ke = (A*E/L) * [C^2 C*S -C^2 -C*S; C*S S^2 -C*S -S^2;
                    -C^2 -C*S C^2 C*S; -C*S -S^2 C*S S^2];

    sctr = [2*n1-1 2*n1 2*n2-1 2*n2]; %posicion en matriz global
    K(sctr,sctr) = K(sctr,sctr) + ke;

end

d(isol) = K(isol,isol)\f(isol);
f = K*d;
sigma=zeros(ne,1);

for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    L = norm(nodos(n2,:)-nodos(n1,:));
    C = (nodos(n2,1)-nodos(n1,1))/L; %cos
    S = (nodos(n2,2)-nodos(n1,2))/L; %sin
    B = E/L*[-C -S C S];
    sctr = [2*n1-1 2*n1 2*n2-1 2*n2];
    sigma(e) = B*d(sctr);
end

clf
sclf=200; %escala deformacion

for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    x1 = nodos(n1,1); y1 = nodos(n1,2);
    x2 = nodos(n2,1); y2 = nodos(n2,2);
    u1 = d(2*n1-1); v1 = d(2*n1);
    u2 = d(2*n2-1); v2 = d(2*n2);
    plot([x1, x2],[y1, y2],'k--');
    plot([x1, x2]+sclf*[u1, u2],[y1, y2]+sclf*[v1,v2],'b');
    hold on
end
title('Grafica Deformacion')
axis equal

Fuerzas = reshape(f,[2,nn]).'
Desplazamientos = reshape(d,[2,nn]).'
sigma
```

```
Fuerzas =

   1.0e+03 *

  -1.3333   -0.5333
        0   -0.0000
   0.0000    1.0000
   1.3333   -0.4667
```

```
Desplazamientos =

        0         0
   0.0149    0.0588
  -0.0171    0.0672
        0         0
```

```
sigma =

  622.2222
  888.8889
 -777.7778
  466.6667
 -711.1111
        0
```



Problem 5.1:

Consider the plane truss shown in Fig. 5.5. Given $E = 210\,\mathrm{GPa}$ and $A = 0.005\,\mathrm{m}^2$, determine:

1. the global stiffness matrix for the structure.
2. the horizontal and vertical displacements at nodes 2, 3, 4, and 5.
3. the horizontal and vertical reactions at nodes 1 and 6.
4. the stress in each element.

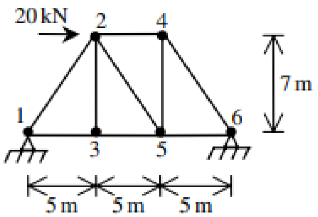


Fig. 5.5. Plane Truss for Problem 5.1

```
nodos = [0 0;5 7;5 0;10 7;10 0;15 0];
conn = [1 2;1 3;2 3;2 4;2 5;3 5;4 5;4 6;5 6];

A=5e-3;
E=210e6;
p=[3 20];

nn=size(nodos,1);
ndof=2*nn; %grados de libertad
f=zeros(ndof,1);
f(p(1))=p(2);

isol=[3 4 5 6 7 8 9 10]; %simplifica matriz en nodos libres
ne=size(conn,1);
```

```matlab
K=zeros(ndof,ndof);
d=zeros(ndof,1);
for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    L = norm(nodos(n2,:)-nodos(n1,:));
    C = (nodos(n2,1)-nodos(n1,1))/L; %cos
    S = (nodos(n2,2)-nodos(n1,2))/L; %sin
    ke = (A*E/L) * [C^2 C*S -C^2 -C*S; C*S S^2 -C*S -S^2;
                    -C^2 -C*S C^2 C*S; -C*S -S^2 C*S S^2];

    sctr = [2*n1-1 2*n1 2*n2-1 2*n2]; %posicion en matriz global
    K(sctr,sctr) = K(sctr,sctr) + ke;
end

d(isol) = K(isol,isol)\f(isol);

f = K*d;
sigma=zeros(ne,1);
for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    L = norm(nodos(n2,:)-nodos(n1,:));
    C = (nodos(n2,1)-nodos(n1,1))/L; %cos
    S = (nodos(n2,2)-nodos(n1,2))/L; %sin
    B = E/L*[-C -S C S];
    sctr = [2*n1-1 2*n1 2*n2-1 2*n2];
    sigma(e) = B*d(sctr);
end

clf
sclf=1000; %escala deformacion
for e=1:ne
    n1 = conn(e,1);
    n2 = conn(e,2);
    x1 = nodos(n1,1); y1 = nodos(n1,2);
    x2 = nodos(n2,1); y2 = nodos(n2,2);
    u1 = d(2*n1-1); v1 = d(2*n1);
    u2 = d(2*n2-1); v2 = d(2*n2);
    plot([x1, x2],[y1, y2],'k--');
    plot([x1, x2]+sclf*[u1, u2],[y1, y2]+sclf*[v1,v2],'b');
    hold on
end
title('Gráfica Deformación')
axis equal

Fuerzas = reshape(f,[2,nn]).'
Desplazamientos = reshape(d,[2,nn]).'
sigma
```

K =

1.0e+05 *

2.5124	0.5773	-0.4124	-0.5773	-2.1000	0	0	0	0	0	0	0
0.5773	0.8082	-0.5773	-0.8082	0	0	0	0	0	0	0	0
-0.4124	-0.5773	2.9247	0	0	0	-2.1000	0	-0.4124	0.5773	0	0
-0.5773	-0.8082	0	3.1165	0	-1.5000	0	0	0.5773	-0.8082	0	0
-2.1000	0	0	0	4.2000	0	0	0	-2.1000	0	0	0
0	0	0	-1.5000	0	1.5000	0	0	0	0	0	0
0	0	-2.1000	0	0	0	2.5124	-0.5773	0	0	-0.4124	0.5773
0	0	0	0	0	0	-0.5773	2.3082	0	-1.5000	0.5773	-0.8082
0	0	-0.4124	0.5773	-2.1000	0	0	0	4.6124	-0.5773	-2.1000	0
0	0	0.5773	-0.8082	0	0	0	-1.5000	-0.5773	2.3082	0	0
0	0	0	0	0	0	-0.4124	0.5773	-2.1000	0	2.5124	-0.5773
0	0	0	0	0	0	0.5773	-0.8082	0	0	-0.5773	0.8082

Fuerzas =

-8.8889	-9.3333
20.0000	0.0000
0	0
0.0000	0.0000
-0.0000	0
-11.1111	9.3333

Desplazamientos =

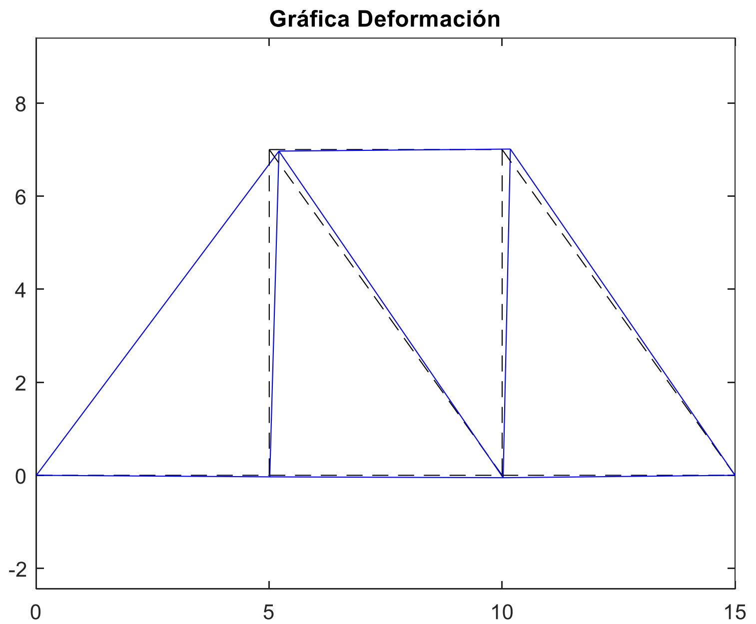
1.0e-03 *

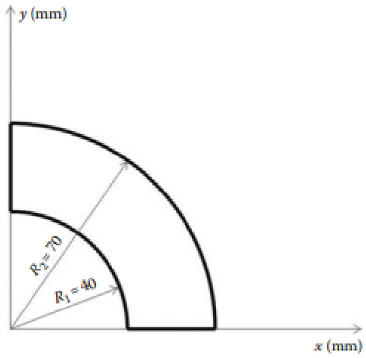
0	0
0.2083	-0.0333
0.0106	-0.0333
0.1766	0.0107
0.0212	-0.0516
0	0

sigma =

1.0e+03 *

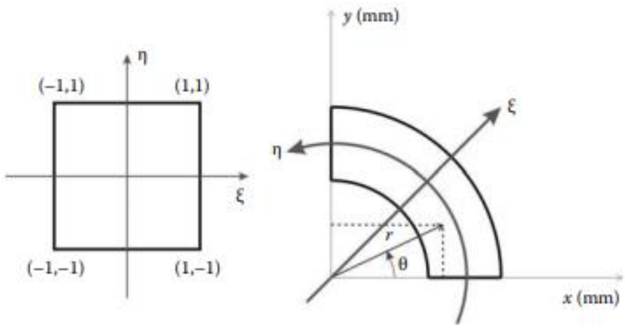
2.2940
0.4444
0
-1.3333
-2.2940
0.4444
1.8667
-2.2940
-0.8889





Integración numérica

Hallar el segundo momento de área usando la cuadratura de Gauss



Forma analítica

$$x = r\cos\theta$$

$$y = r\sin\theta$$

$$\theta = \frac{\pi}{4}\eta + \frac{\pi}{4}$$

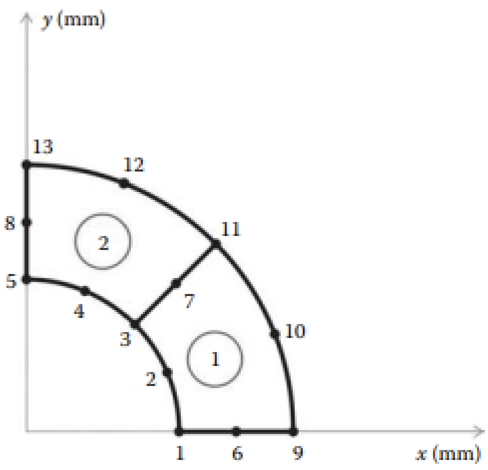
$$r = \frac{R_2 - R_1}{2}\xi + \frac{R_2 + R_1}{2} \qquad dA = r\,dr\,d\theta \qquad I_{xx} = \int_0^{\pi/2}\int_{R_1}^{R_2} (r\sin\theta)^2 r\,dr\,d\theta$$

$$I_{xx} = \int_{R_1}^{R_2} r^3 dr \int_0^{\pi/2} (\sin\theta)^2 d\theta = \left[\frac{r^4}{4}\right]_{R_1}^{R_2}\left[\frac{\theta}{2} - \frac{1}{4}\sin(2\theta)\right]_0^{\pi/2} = \frac{\pi(R_2^4 - R_1^4)}{16}$$

$$I_{xx} = 4{,}211{,}700 \ \ \text{mm}^4$$

Elementos finitos: Cuadratura de Gauss

$$I_{xx} = I_{xx}^{(1)} + I_{xx}^{(2)}$$

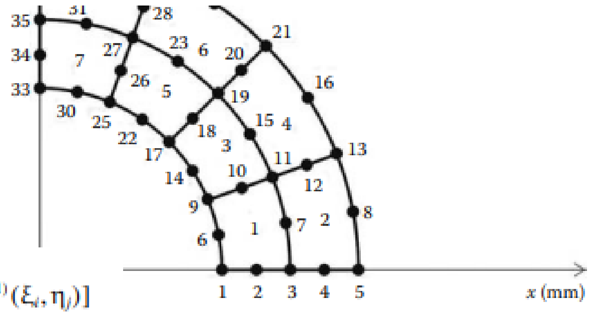


$$I_{xx}^{(1)} = \int_{A1} y^2 dx\, dy$$

$$I_{xx}^{(2)} = \int_{A2} y^2 dx\, dy$$

$$y = N_1(\xi, \eta) y_1 + N_2(\xi, \eta) y_2 + \cdots + N_8(\xi, \eta) y_8$$

$$dxdy = [J(\xi, \eta)] d\xi d\eta = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} d\xi d\eta$$



$$I_{xx}^{(1)} = \sum_{i=1}^{ngp} \sum_{j=1}^{ngp} \left(\sum_{k=1}^{8} N_k(\xi_i, \eta_j) y_k^{(1)} \right)^2 W_i W_j det[J^{(1)}(\xi_i, \eta_j)]$$

$$I_{xx}^{(2)} = \sum_{i=1}^{ngp} \sum_{j=1}^{ngp} \left(\sum_{k=1}^{8} N_k(\xi_i, \eta_j) y_k^{(2)} \right)^2 W_i W_j det[J^{(2)}(\xi_i, \eta_j)]$$

$$I_{xx}^{(1)} = \int_{-1}^{+1} \int_{-1}^{+1} \left(\sum_{k=1}^{8} N_k(\xi, \eta) y_k^{(1)} \right)^2 det[J^{(1)}(\xi, \eta)]\, d\xi\, d\eta$$

$$I_{xx}^{(2)} = \int_{-1}^{+1} \int_{-1}^{+1} \left(\sum_{k=1}^{8} N_k(\xi, \eta) y_k^{(2)} \right)^2 det[J^{(2)}(\xi, \eta)]\, d\xi\, d\eta$$

Cuadratura de Gauss

Segundo Momento de Área

```
global geom connec nel nne nnd RI RE

RI = 40; % Radio Interno
RE = 70; % Radio Externo

k_Malla2 % Datos de entrada

% Numero de puntos de Gauss
ngp = 3; % polinomios de grado 5

% Valores de x y pesos asociados
samp=zeros(ngp,2);

if ngp==1
    samp=[0. 2];
elseif ngp==2
    samp=[-1/sqrt(3) 1; 1/sqrt(3) 1];
```

```
elseif ngp==3
    samp= [-.2*sqrt(15) 5/9; 0 8/9; .2*sqrt(15) 5/9];
elseif ngp==4
    samp= [-0.861136311594053 0.347854845137454;
            -0.339981043584856 0.652145154862546;
            0.339981043584856 0.652145154862546;
            0.861136311594053 0.347854845137454];
end

Ixx = 0.;

for k=1:nel
    coord=zeros(nne,2);%coordenadas del elemento
    for i=1: nne
        coord(i,:)=geom(connec(k,i),:);
    end
    X = coord(:,1);
    Y = coord(:,2);

    for i=1:ngp
        xi = samp(i,1);
        WI = samp(i,2);
        for j =1:ngp
            eta = samp(j,1);
            WJ = samp(j,2);
            [der,fun] = fmquad(samp, i,j);
            JAC = der*coord; % jacobiano
            DET =det(JAC);
            Ixx =Ixx+ (dot(fun,Y))^2*WI*WJ*DET;
        end
    end
end
Ixx

function[der,fun] = fmquad(samp, ig,jg)
%vector de funcion de forma y derivadas
xi=samp(ig,1);
eta=samp(jg,1);
etam=(1.-eta);
etap=(1.+eta);
xim=(1.-xi);
xip=(1.+xi);

fun(1) = -0.25*xim*etam*(1.+ xi + eta);
fun(2) = 0.5*(1.- xi^2)*etam;
fun(3) = -0.25*xip*etam*(1. - xi + eta);
fun(4) = 0.5*xip*(1. - eta^2);
fun(5) = -0.25*xip*etap*(1. - xi - eta);
fun(6) = 0.5*(1. - xi^2)*etap;
fun(7) = -0.25*xim*etap*(1. + xi - eta);
fun(8) = 0.5*xim*(1. - eta^2);

der(1,1)=0.25*etam*(2.*xi + eta); der(1,2)=-1.*etam*xi;
der(1,3)=0.25*etam*(2.*xi-eta); der(1,4)=0.5*(1-eta^2);
der(1,5)=0.25*etap*(2.*xi+eta); der(1,6)=-1.*etap*xi;
der(1,7)=0.25*etap*(2.*xi-eta); der(1,8)=-0.5*(1.-eta^2);

der(2,1)=0.25*xim*(2.*eta+xi); der(2,2)=-0.5*(1. - xi^2);
der(2,3)=-0.25*xip*(xi-2.*eta); der(2,4)=-1.*xip*eta;
der(2,5)=0.25*xip*(xi+2.*eta); der(2,6)=0.5*(1.-xi^2);
```

```
der(2,7)=-0.25*xim*(xi-2.*eta); der(2,8)=-1.*xim*eta;
end
```

$$Ixx = Ixx + (dot(fun, Y))^2 * WI * WJ * DET$$

Datos de entrada:

2 Elementos

```
% Malla de 2 elementos
global geom connec nel nne nnd RI RE
nnd = 13; % # de nodos

%Coordenadas de nodos

geom = ...
[RI 0.;                                              % node 1
RI*cos(pi/8) RI*sin(pi/8);                           % node 2
RI*cos(pi/4) RI*sin(pi/4);                           % node 3
RI*cos(3*pi/8) RI*sin(3*pi/8);                       % node 4
RI*cos(pi/2) RI*sin(pi/2);                           % node 5
(RI+RE)/2 0.;                                        % node 6
((RI+RE)/2)*cos(pi/4) ((RI+RE)/2)*sin(pi/4);         % node 7
((RI+RE)/2)*cos(pi/2) ((RI+RE)/2)*sin(pi/2);         % node 8
RE 0.;                                               % node 9
RE*cos(pi/8) RE*sin(pi/8);                           % node 10
RE*cos(pi/4) RE*sin(pi/4);                           % node 11
RE*cos(3*pi/8) RE*sin(3*pi/8);                       % node 12
RE*cos(pi/2) RE*sin(pi/2)];                          % node 13

nel = 2; % # de elementos
nne = 8; % # de nodos por elemento

% Conexiones

connec = [1 6 9 10 11 7 3 2;      % Element 1
            3 7 11 12 13 8 5 4]; % Element 2
```

8 Elementos

```
% Malla de 8 elementos
global geom connec nel nne nnd RI RE
nnd = 37; % # de nodos

%Coordenadas de nodos

geom = ...
[RI 0.;                                                  % node 1
RI+(RE-RI)/4 0.;                                         % node 2
RI+(RE-RI)/2 0.;                                         % node 3
RI+3*(RE-RI)/4 0.;                                       % node 4
RE 0.;                                                   % node 5
RI*cos(pi/16) RI*sin(pi/16);                             % node 6
(RI+(RE-RI)/2)*cos(pi/16) (RI+(RE-RI)/2)*sin(pi/16);     % node 7
RE*cos(pi/16) RE*sin(pi/16);                             % node 8
RI*cos(pi/8) RI*sin(pi/8);                               % node 9
(RI+(RE-RI)/4)*cos(pi/8) (RI+(RE-RI)/4)*sin(pi/8);       % node 10
(RI+(RE-RI)/2)*cos(pi/8) (RI+(RE-RI)/2)*sin(pi/8);       % node 11
```

```
(RI+3*(RE-RI)/4)*cos(pi/8) (RI+3*(RE-RI)/4)*sin(pi/8);  % node 12
RE*cos(pi/8) RE*sin(pi/8);                              % node 13
RI*cos(3*pi/16) RI*sin(3*pi/16);                        % node 14
(RI+(RE-RI)/2)*cos(3*pi/16) (RI+(RE-RI)/2)*sin(3*pi/16);% node 15
RE*cos(3*pi/16) RE*sin(3*pi/16);                        % node 16
RI*cos(pi/4) RI*sin(pi/4);                              % node 17
(RI+(RE-RI)/4)*cos(pi/4) (RI+(RE-RI)/4)*sin(pi/4);      % node 18
(RI+(RE-RI)/2)*cos(pi/4) (RI+(RE-RI)/2)*sin(pi/4);      % node 19
(RI+3*(RE-RI)/4)*cos(pi/4) (RI+3*(RE-RI)/4)*sin(pi/4);  % node 20
RE*cos(pi/4) RE*sin(pi/4);                              % node 21
RI*cos(5*pi/16) RI*sin(5*pi/16);                        % node 22
(RI+(RE-RI)/2)*cos(5*pi/16) (RI+(RE-RI)/2)*sin(5*pi/16);% node 23
RE*cos(5*pi/16) RE*sin(5*pi/16);                        % node 24
RI*cos(6*pi/16) RI*sin(6*pi/16);                        % node 25
(RI+(RE-RI)/4)*cos(6*pi/16) (RI+(RE-RI)/4)*sin(6*pi/16);% node 26
(RI+(RE-RI)/2)*cos(6*pi/16) (RI+(RE-RI)/2)*sin(6*pi/16);% node 27
(RI+3*(RE-RI)/4)*cos(6*pi/16) (RI+3*(RE-RI)/4)*sin(6*pi/16);% node 28
RE*cos(6*pi/16) RE*sin(6*pi/16);                        % node 29
RI*cos(7*pi/16) RI*sin(7*pi/16);                        % node 30
(RI+(RE-RI)/2)*cos(7*pi/16) (RI+(RE-RI)/2)*sin(7*pi/16);% node 31
RE*cos(7*pi/16) RE*sin(7*pi/16);                        % node 32
RI*cos(pi/2) RI*sin(pi/2);                              % node 33
(RI+(RE-RI)/4)*cos(pi/2) (RI+(RE-RI)/4)*sin(pi/2);      % node 34
(RI+(RE-RI)/2)*cos(pi/2) (RI+(RE-RI)/2)*sin(pi/2);      % node 35
(RI+3*(RE-RI)/4)*cos(pi/2) (RI+3*(RE-RI)/4)*sin(pi/2);  % node 36
RE*cos(pi/2) RE*sin(pi/2)];                              % node 37

nel = 8; % # de elementos
nne = 8; % # de nodos por elemento

% Conexiones

connec = [1 2 3 7 11 10 9 6;     % Element 1
3 4 5 8 13 12 11 7;              % Element 2
9 10 11 15 19 18 17 14;          % Element 3
11 12 13 16 21 20 19 15;         % Element 4
17 18 19 23 27 26 25 22;         % Element 5
19 20 21 24 29 28 27 23;         % Element 6
25 26 27 31 35 34 33 30;         % Element 7
27 28 29 32 37 36 35 31];        % Element 8
```

Malla 2 elementos

```
Ixx =

   4.2051e+06
```

Malla 8 elementos

```
Ixx =

   4.2113e+06
```

PROBLEMA BIDIMENSIONAL LINEAL (2D)

- El campo de temperatura en el domino Ω, cuya frontera es $\Gamma_1 \cup \Gamma_2$ representado en la Figura 1, con condiciones de contorno diferentes para cada $\Gamma_i (i = 1{,}2)$, es descrito por la ecuación diferencial,

$$\begin{cases} \nabla . (k\nabla T) = 0 \ \ en \ \Omega \\ T = 0 \ \ en \ \Gamma_1, \\ \boldsymbol{n}.\nabla T = q \ \ en \ \Gamma_2 \end{cases}$$

donde k es la conductividad térmica

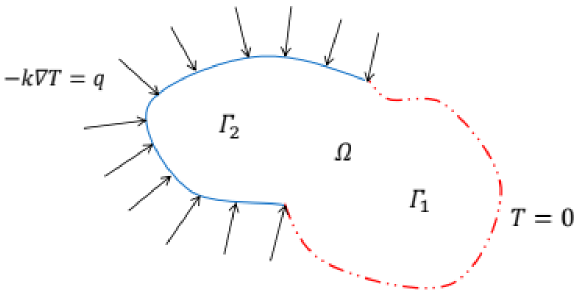


Figura 1: Problema Bidimensional

Hallar $T \in U$ tal que:

$$\int_{\Omega} w\nabla^2 T \, d\Omega = 0 \,, \qquad \forall w \in V;$$

$$U = \left\{ T \middle| \int_{\Omega} |\nabla T|^2 \, d\Omega < \infty; T(\Gamma_1) = 0 \right\}, \qquad V = \left\{ w \middle| \int_{\Omega} |\nabla w|^2 \, d\Omega < \infty; w(\Gamma_1) = 0 \right\}.$$

Residuos ponderados

$$\int_{\Omega} w \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right] dxdy = -\int_{\Omega} \left[\frac{\partial w}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial w}{\partial y}\frac{\partial T}{\partial y}\right] dxdy + \int_{\Omega} \frac{\partial}{\partial x}\left[w\frac{\partial T}{\partial x}\right] + \frac{\partial}{\partial y}\left[w\frac{\partial T}{\partial y}\right] dxdy$$

En forma vectorial;

$$\int_{\Omega} w\nabla^2 T d\Omega = -\int_{\Omega} \nabla w \,.\, \nabla T d\Omega +$$

Usando el teorema de la Divergencia,

$$\int_{\Omega} \nabla \,.\, \boldsymbol{f} \, d\Omega = \int_{\Gamma} \boldsymbol{n}$$

La formulación débil queda,

$$\int_{\Omega} w\nabla^2 T d\Omega = -\int_{\Omega} \nabla w \,.\, \nabla T d$$

$$\int_{\Gamma} w\boldsymbol{n}.\nabla T d\Gamma = \int_{\Gamma_1} \underbrace{w\boldsymbol{n}.\nabla T}_{0} \, d\Gamma_1 + \int_{\Gamma_2} \underbrace{w\boldsymbol{n}.\nabla T}_{q} \, d\Gamma_2$$

Condiciones de contorno

$$T_h = \sum_{j=1}^{N} C_j \phi_j \qquad \nabla T = \sum_{j=1}^{N} C_j \nabla \phi_j$$

$$-\int_{\Omega} \nabla w \,.\, \nabla T d\Omega + \int_{\Gamma_2} w \, q \, d\Gamma_2 = 0$$

$$R_i = -\int_{\Omega} \nabla\psi_i \,.\, \nabla T d\Omega + \int_{\Gamma_2} \psi_i \, q \, d\Gamma_2 = 0$$ **Expresión de residuo**

$$\int_{\Omega} \nabla\psi_i \,.\, \nabla\left(\sum_{j=1}^{N} C_j\phi_j\right) d\Omega = \int_{\Gamma_2} \psi_i \, q \, d\Gamma_2 \;\Rightarrow\; \sum_{j=1}^{N} C_j \underbrace{\left\{\int_{\Omega} \nabla\psi_i \,.\nabla\phi_j d\Omega\right\}}_{A_{ij}} = \underbrace{\int_{\Gamma_2} \psi_i \, q \, d\Gamma_2}_{b_i}$$

$$A_{ij}c_j = b_i \qquad A_{ij} = \int_{\Omega} \nabla\psi_i \,.\nabla\phi_j d\Omega \qquad b_i = \int_{\Gamma_2} \psi_i \, q \, d\Gamma_2 \quad \Rightarrow \quad \boxed{A_{ij} = \int_{\Omega} \nabla\phi_i \,.\nabla\phi_j d\Omega} \quad \boxed{b_i = \int_{\Gamma_2} \phi_i \, q \, d\Gamma_2}$$

$$\phi(x_j) = \delta_{ij} \Rightarrow \phi_i(x_j) = \begin{cases} 1 \;\; si \;\; i = j \\ 0 \;\; si \;\; i \neq j \end{cases}$$

$$A_{ij}^{(e)} = \int_{\Omega_{(e)}} \nabla\phi_i \,.\nabla\phi_j d\Omega \;\Rightarrow\; A_{ij}^{(e)} = \int_{\Omega_{(e)}} \left[\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x} + \frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y}\right] d\Omega$$

$$x = x(\eta,\xi) \quad y = y(\eta,\xi) \qquad \begin{cases} x(\xi,\eta) = \sum_{i=1}^{4} X_i\phi_i(\xi,\eta) \\ y(\xi,\eta) = \sum_{i=1}^{4} Y_i\phi_i(\xi,\eta) \end{cases}$$

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^{N} X_i \frac{\partial\phi_i}{\partial \xi} \;;\; \frac{\partial x}{\partial \eta} = \sum_{i=1}^{N} X_i \frac{\partial\phi_i}{\partial \eta} \;; \ldots$$

$$\begin{bmatrix} \dfrac{\partial\phi_i}{\partial\xi} \\ \dfrac{\partial\phi_i}{\partial\eta} \end{bmatrix} = \underbrace{\begin{bmatrix} \dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi} \\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta} \end{bmatrix}}_{\text{Jacobiano de Transformación}} \begin{bmatrix} \dfrac{\partial\phi_i}{\partial x} \\ \dfrac{\partial\phi_i}{\partial y} \end{bmatrix}$$

$$\int_{-1}^{1}\int_{-1}^{1} F(\xi,\eta)d\xi d\eta = \sum_{igp=1}^{NGPI}\sum_{igp=1}^{NGPJ} F(\xi_i,\eta_j)w_I w_J$$

$$A_{ij}^{(e)} = \int_{\Omega_{(e)}} \nabla\phi_i \,.\nabla\phi_j \, dxdy = \int_{-1}^{1}\int_{-1}^{1} \left(\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x} + \frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y}\right)|J|d\xi d\eta$$

$$A_{ij}^{(e)} = \int_{-1}^{1}\int_{-1}^{1} \left\{\left(\frac{\partial y}{\partial\eta}\frac{\partial\phi_i}{\partial\xi} - \frac{\partial y}{\partial\xi}\frac{\partial\phi_i}{\partial\eta}\right)\left(\frac{\partial y}{\partial\eta}\frac{\partial\phi_j}{\partial\xi} - \frac{\partial y}{\partial\xi}\frac{\partial\phi_j}{\partial\eta}\right) + \left(-\frac{\partial x}{\partial\eta}\frac{\partial\phi_i}{\partial\xi} + \frac{\partial x}{\partial\xi}\right.\right.$$

Cuadratura de Gauss

$$XY(i,j) = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \end{bmatrix}$$

$$PHI(I) = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix}, \qquad GradPHI(i,j) = \begin{bmatrix} \frac{\partial \phi_1}{\partial \xi} & \frac{\partial \phi_2}{\partial \xi} & \frac{\partial \phi_3}{\partial \xi} & \frac{\partial \phi_4}{\partial \xi} \\ \frac{\partial \phi_1}{\partial \eta} & \frac{\partial \phi_2}{\partial \eta} & \frac{\partial \phi_3}{\partial \eta} & \frac{\partial \phi_4}{\partial \eta} \end{bmatrix}$$

$$JAC = GradPHI\,.XY^T$$

$$GradPHIXY(i,j) = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} & \frac{\partial \phi_2}{\partial x} & \frac{\partial \phi_3}{\partial x} & \frac{\partial \phi_4}{\partial x} \\ \frac{\partial \phi_1}{\partial y} & \frac{\partial \phi_2}{\partial y} & \frac{\partial \phi_3}{\partial y} & \frac{\partial \phi_4}{\partial y} \end{bmatrix}$$

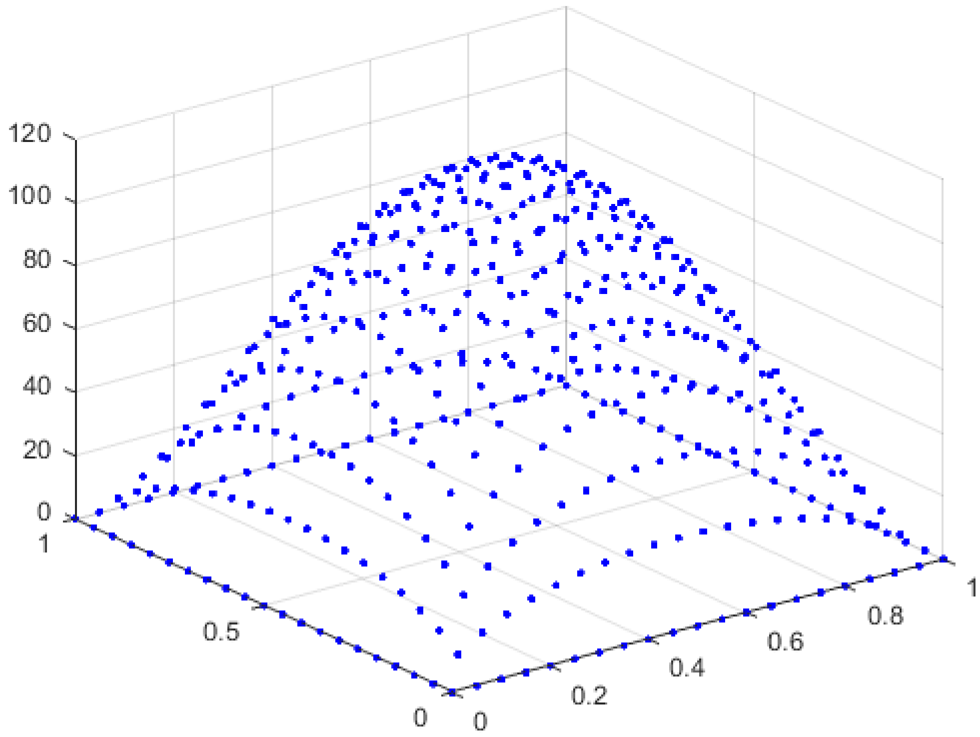
$$GradPHIXY = J^{-1}\,GradPHI$$

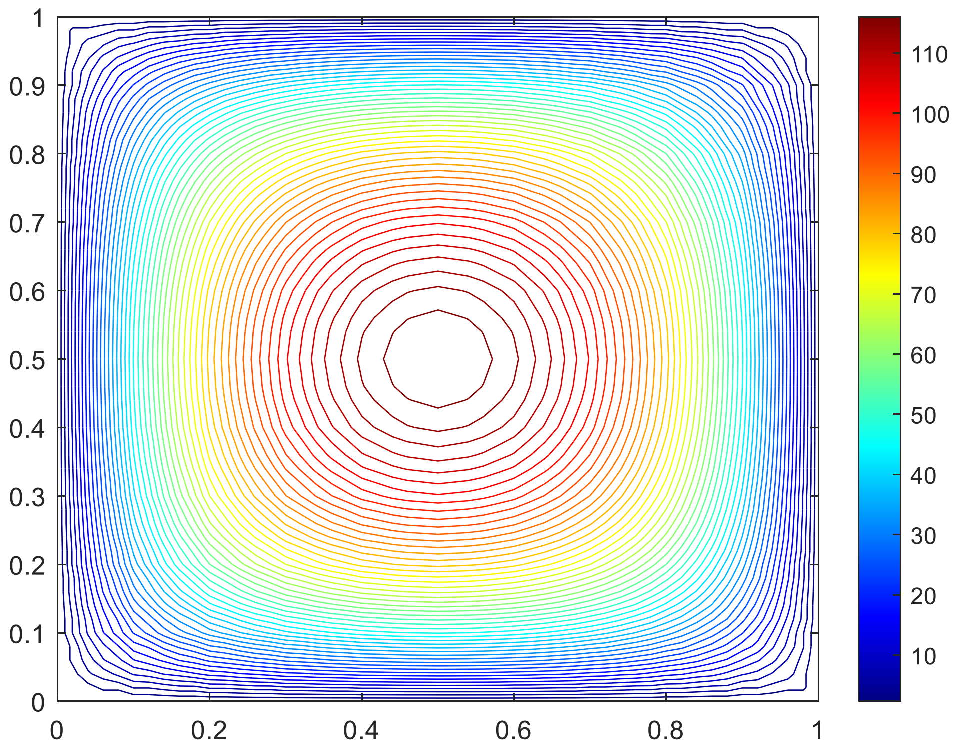
```
L  = 1; H  = 1; %Geometría
NEX = 20; NEY = 20; % Malla
NELE = NEX * NEY; % # elementos
NODES = (NEX+1) * (NEY+1); % # nodos

% Relación de la numeración local y global de los nodos
[DomNodeID] = NodeIndex(NELE,NEX,NEY);

% Cálculo de las coordenadas de los puntos nodales
tanALFA=1/0.0;
DX=L/NEX;
DY=H/NEY;
for ix=1:NEX
    for jy=1:NEY
        iele=(ix-1)*NEY+jy;
        xloc(1)=(ix-1)*DX+(jy-1)*DY/(tanALFA);
        xloc(2)=xloc(1)+DX;
        xloc(3)=xloc(2)+DY/(tanALFA);
        xloc(4)=xloc(1)+DY/(tanALFA);
        yloc(1)=(jy-1)*DY;
        yloc(2)=yloc(1);
        yloc(3)=yloc(2)+DY;
        yloc(4)=yloc(3);

        for ilnode=1:4
            ignode=DomNodeID(ilnode,iele);
            X(ignode)=xloc(ilnode);
            Y(ignode)=yloc(ilnode);
        end
    end
end

A = zeros(NODES, NODES);
b = zeros(NODES, 1);
for iele = 1:NELE
%Calcula la Matriz Elemental
[Aelem, belem] =  GetElemAb(iele,DomNodeID,X,Y);
  for ilnode = 1:4
      ignode = DomNodeID(ilnode,iele);
    for jlnode = 1:4
        jgnode = DomNodeID(jlnode,iele);
       A(ignode, jgnode) = A(ignode, jgnode) + Aelem(ilnode, jlnode);
    end
```

```matlab
        b(ignode) = b(ignode) + belem(ilnode);
    end
end

% Busqueda de los nodos en el  contorno del dominio ( T(0,y) = T(1,y)=
T(x,0) = T(x,1)= 0.0)
BdBox = [0 L 0 H];
eps = 0.01*sqrt((BdBox(2)- BdBox(1))*(BdBox(4)- BdBox(3))/size(X,1));
CC = find(abs(X(1,:)-BdBox(3))<eps | abs(X(1,:)-BdBox(4))<eps ...
    | abs(Y(1,:)-BdBox(1))< eps | abs(Y(1,:)-BdBox(2)) < eps);
A (CC,:) = []; b(CC) = [];
T = A\b;

figure(2)
plot3(X,Y,T','b.'); grid on;
tri = delaunay(X,Y);
figure
contourTri(tri,X,Y,T',50)
colormap jet
colorbar

function [Aelem, belem] = GetElemAb(iele,DomNodeID,X,Y)
Aelem = zeros(4,4);
belem = zeros(4,1);

%Cuadratura de Gauss
NGP=3;
WGP=[0.555,0.888,0.555];
XGP = [ -0.7745966, 0.0, 0.7745966];

% Valor elemental de las coordenadas de los nodos
for ilnode = 1:4
    ignode = DomNodeID(ilnode,iele);
    XY(1,ilnode) = X(ignode);
    XY(2,ilnode) = Y(ignode);
end
for igp = 1:NGP
XI = XGP(igp);
WI = WGP(igp);
  for jgp = 1:NGP
      NETA = XGP(jgp);
      WJ =WGP(jgp);
      W = WI*WJ;
      Phi = [  ((NETA-1)*(XI-1))/4; -((NETA-1)*(XI+1))/4;
             ((NETA+1)*(XI+1))/4; -((NETA+1)*(XI-1))/4];
      GradPhi=[ (NETA-1)/4  -(NETA-1)/4  (NETA+1)/4   -(NETA+1)/4;
                (XI-1)/4    -(XI+1)/4    (XI+1)/4     -(XI-1)/4];
      JAC = GradPhi * XY';
      JACinv = inv(JAC);
      JACdet = det(JAC);
      %calculando GradPhixy (derivada en relacion a x e y de las
funciones base)
      GradPhixy = JACinv*GradPhi;
      for ilnode = 1:4
          for jlnode = 1:4
          Ae =  W*JACdet*(GradPhixy(1,ilnode)*GradPhixy(1,jlnode) + ...
                          GradPhixy(2,ilnode)*GradPhixy(2,jlnode));
          Aelem(ilnode,jlnode) = Aelem(ilnode,jlnode) + Ae;
          end
          belem(ilnode) =  1.0;
```

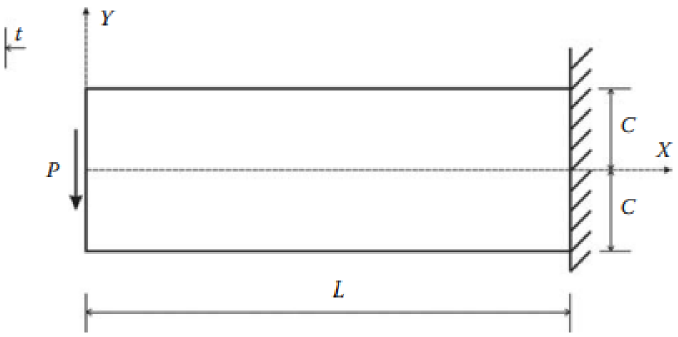
```
        end
    end
end
end

function [DomNodeID] = NodeIndex(NELE,NEX,NEY)
DomNodeID = zeros(4,NELE);
NodeCount = 1;
for iele = 1:NELE
    iWestEle = 0;
    iSouthEle = 0;
    % Elemento de la izquierda
    if (iele > NEY)
        iWestEle = iele - NEY;
    end
    % Elemento de arriba
    iSouth = mod(iele-1,NEY);
    if (iSouth == 0)
        iSouthEle = 0;
    else
        iSouthEle = iele-1;
    end
    if (iWestEle ~= 0) %nodos comunes con elemento izquierdo
        DomNodeID(1,iele) = DomNodeID(2,iWestEle);
        DomNodeID(4,iele) = DomNodeID(3,iWestEle);
    end
    if (iSouthEle ~= 0) %nodos comunes con elemento superior
        DomNodeID(1,iele) = DomNodeID(4,iSouthEle);
        DomNodeID(2,iele) = DomNodeID(3,iSouthEle);
    end
    for ilnode = 1:4
        if (DomNodeID(ilnode,iele) == 0)
            DomNodeID(ilnode,iele) = NodeCount;
            NodeCount = NodeCount + 1;
        end
    end
end
end

function contourTri(t,x,y,f,N)
Interval = 50; % nr of intervals for a uniform mesh
[X Y] = meshgrid(min(x(:)):(max(x(:))-
min(x(:)))/Interval:max(x(:)), ...
        min(y(:)):(max(y(:))-min(y(:)))/Interval:max(y(:)));
Z = X.*0-1.3e10;
% go through triangles and interpolate mesh points:
for i=1:size(t,1)
    x1 = x(t(i,1)); x2 = x(t(i,2)); x3 = x(t(i,3));
    y1 = y(t(i,1)); y2 = y(t(i,2)); y3 = y(t(i,3));
    z1 = f(t(i,1)); z2 = f(t(i,2)); z3 = f(t(i,3));
    inp  = inpolygon(X,Y,[x1 x2 x3 x1], [y1 y2 y3 y1]);
    ids  = find(inp(:));
    for j=1:length(ids)
        A = 0.5*det([1 1 1;x1 x2 x3;y1 y2 y3]);
        B = [x2*y3-x3*y2 y2-y3 x3-x2; x3*y1-x1*y3 y3-y1 x1-x3;
             x1*y2-x2*y1 y1-y2 x2-x1];
        Xi = 1/(2*A)*B*[1 X(ids(j)) Y(ids(j))]';
        xi1 = Xi(1); xi2 = Xi(2); xi3 = Xi(3);
        Z(ids(j)) = [z1 z2 z3]*[xi1 xi2 xi3]';
    end
end
```

```
Z((Z==-1.3e10)) = NaN;
contour(X,Y,Z,N)
end
```





Viga voladiza

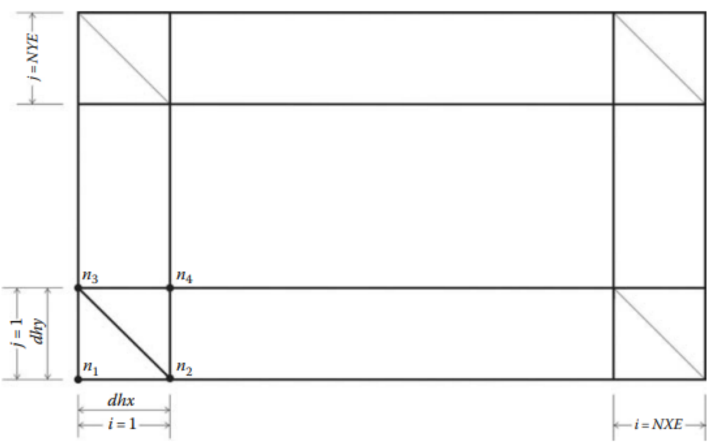


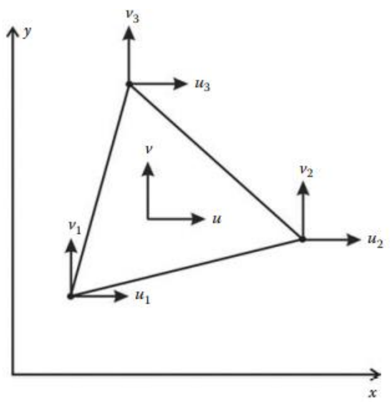
C = 10 mm, L = 60, t = 5

E = 200000 MPa

v = 0.3

P = 1000 N





$$N_1(x, y) = m_{11} + m_{12}x + m_{13}y$$
$$N_2(x, y) = m_{21} + m_{22}x + m_{23}y$$
$$N_3(x, y) = m_{31} + m_{32}x + m_{33}y$$

$$m_{11} = \frac{x_2y_3 - x_3y_2}{2A} \qquad m_{12} = \frac{y_2 - y_3}{2A} \qquad m_{13} = \frac{x_3 - x_2}{2A}$$
$$m_{21} = \frac{x_3y_1 - x_1y_3}{2A} \qquad m_{22} = \frac{y_3 - y_1}{2A} \qquad m_{23} = \frac{x_1 - x_3}{2A}$$
$$m_{31} = \frac{x_1y_2 - x_2y_1}{2A} \qquad m_{32} = \frac{y_1 - y_2}{2A} \qquad m_{33} = \frac{x_2 - x_1}{2A}$$

$$A = \frac{1}{2}\det\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}$$

$$u = N_1u_1 + N_2u_2 + N_3u_3$$
$$v = N_1v_1 + N_2v_2 + N_3v_3$$

Desplazamientos

$$\{U\} = [N]\{a\}$$

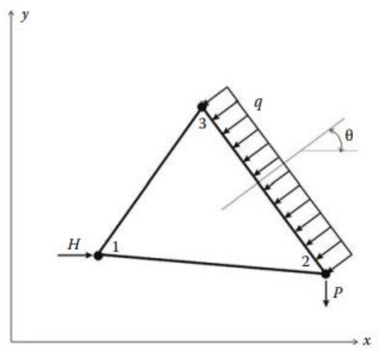
$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \left[\begin{array}{cc|cc|cc} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{array}\right] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}$$

Deformación $\{\epsilon\} = [B]\{a\}$

$$[B] = \begin{bmatrix} m_{12} & 0 & | & m_{22} & 0 & | & m_{32} & 0 \\ 0 & m_{13} & | & 0 & m_{23} & | & 0 & m_{33} \\ m_{13} & m_{12} & | & m_{23} & m_{22} & | & m_{33} & m_{32} \end{bmatrix}$$

$$[K_e] = [B]^T[D][B]tA_e \qquad [B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & | & \frac{\partial N_2}{\partial x} & 0 & | & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & | & 0 & \frac{\partial N_2}{\partial y} & | & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & | & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & | & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix}$$

Matriz de rigidez



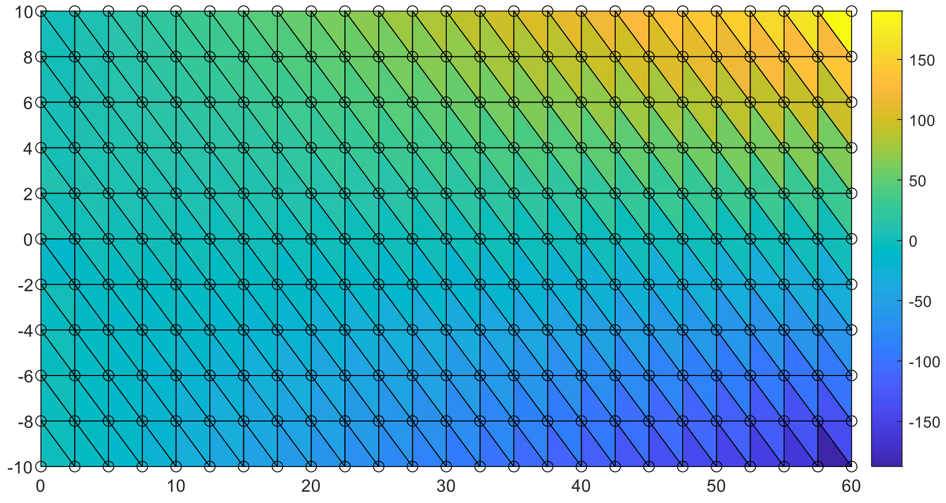
$$\sum_i [N_{(\{x\}=\{\bar{x}\})}]^T\{P\}_i = \begin{bmatrix} N_1 = 1 & 0 \\ 0 & N_1 = 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} H \\ 0 \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ N_2 = 1 & 0 \\ 0 & N_2 = 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ -P \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ H \\ 0 \\ 0 \\ -P \end{Bmatrix}$$

```
global nnd nel nne nodof eldof n
global geom connec dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin dhx dhy

format long g

Length = 60.;
Width =20.;
NXE = 24; % #elementos x
NYE = 10; % #elementos y
dhx = Length/NXE;
dhy = Width/NYE;
X_origin = 0 ;
Y_origin = Width/2 ;

nne = 3; % #nodos elemento
nodof = 2; % grados de libertad nodo
eldof = nne*nodof;

%Malla
nnd = 0;
k = 0;
```

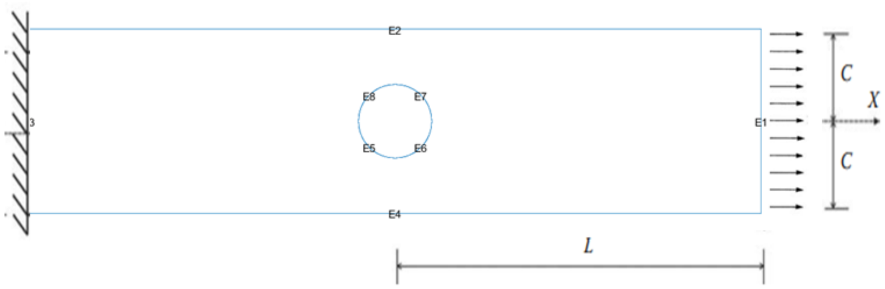
```
for i = 1:NXE
    for j=1:NYE
        k = k + 1;
        n1 = j + (i-1)*(NYE + 1);
        geom(n1,:) = [(i-1)*dhx - X_origin (j-1)*dhy - Y_origin ];
        n2 = j + i*(NYE+1);
        geom(n2,:) = [i*dhx - X_origin (j-1)*dhy - Y_origin ];
        n3 = n1 + 1;
        geom(n3,:) = [(i-1)*dhx - X_origin j*dhy - Y_origin ];
        n4 = n2 + 1;
        geom(n4,:) = [i*dhx- X_origin j*dhy - Y_origin ];
        nel = 2*k;
        m = nel -1;
        connec(m,:) = [n1 n2 n3];
        connec(nel,:) = [n2 n4 n3];
        nnd = n4;
    end
end

% Propiedades Material
E = 200000.; % Modulo de Young
vu = 0.3; % Coeficiente Poisson
thick = 5; % Grosor (mm)

c=E/(1.-vu*vu);
dee=c*[1 vu 0. ; vu 1 0. ; 0. 0. .5*(1.-vu)];

%Condiciones de frontera
nf = ones(nnd, nodof);
% Restringe los nodos donde x = Length
for i=1:nnd
    if geom(i,1) == Length;
        nf(i,:) = [0 0];
    end
end

n=0; for i=1:nnd
    for j=1:nodof
        if nf(i,j) ~= 0
            n=n+1;
            nf(i,j)=n;
        end
    end
end

% Cargas
Nodal_loads= zeros(nnd, 2);
% Aplicar la carga en los nodos donde X = Y =0.
Force = 1000.; % N
for i=1:nnd
    if geom(i,1) == 0. && geom(i,2) == 0.
        Nodal_loads(i,:) = [0. -Force];
    end
end

%Ensamble
fg=zeros(n,1);
for i=1: nnd
    if nf(i,1) ~= 0
        fg(nf(i,1))= Nodal_loads(i,1);
```

```
    end
    if nf(i,2) ~= 0
        fg(nf(i,2))= Nodal_loads(i,2);
    end
end

kk = zeros(n, n);
for i=1:nel
    global eldof
    [bee,g,A] = elem_T3(i);
    ke=thick*A*bee'*dee*bee;
    for i=1:eldof
        if g(i) ~= 0
            for j=1: eldof
                if g(j) ~= 0
                    kk(g(i),g(j))= kk(g(i),g(j)) + ke(i,j);
                end
            end
        end
    end
end
delta = kk\fg ; % Solucion

%Desplazamiento de nodos
for i=1: nnd
    if nf(i,1) == 0
        x_disp =0.;
    else
        x_disp = delta(nf(i,1));
    end
    %
    if nf(i,2) == 0
        y_disp = 0;
    else
        y_disp = delta(nf(i,2));
    end
    node_disp(i,:) =[x_disp y_disp];
end

k = 0;
vertical_disp=zeros(1,NXE+1);
for i=1:nnd;
    if geom(i,2)== 0.
        k=k+1;
        x_coord(k) = geom(i,1);
        vertical_disp(k)=node_disp(i,2);
    end
end
%
for i=1:nel
    [bee,g,A] = elem_T3(i); %Coord. del elemento y vector
    eld=zeros(eldof,1);
    for m=1:eldof
        if g(m)==0
            eld(m)=0.;
        else
            eld(m)=delta(g(m)); % Desplazamiento de elemento
        end
    end
    eps=bee*eld;
    EPS(i,:)=eps ;
```

```
    sigma=dee*eps;
    SIGMA(i,:)=sigma ; % Esfuerzos
end

%Gráfica
x_stress = SIGMA(:,1);
cmin = min(x_stress);
cmax = max(x_stress);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',x_stress, ...
    'Facecolor','flat','Marker','o');
colorbar;

function[bee,g,A] = elem_T3(i)
global nnd nel nne nodof eldof n
global geom connec dee nf load

x1 = geom(connec(i,1),1); y1 = geom(connec(i,1),2);
x2 = geom(connec(i,2),1); y2 = geom(connec(i,2),2);
x3 = geom(connec(i,3),1); y3 = geom(connec(i,3),2);
A = (0.5)*det([1 x1 y1; 1 x2 y2; 1 x3 y3]);

m11 = (x2*y3 - x3*y2)/(2*A);
m21 = (x3*y1 - x1*y3)/(2*A);
m31 = (x1*y2 - y1*x2)/(2*A);
m12 = (y2 - y3)/(2*A);
m22 = (y3 - y1)/(2*A);
m32 = (y1 - y2)/(2*A);
m13 = (x3 - x2)/(2*A);
m23 = (x1 - x3)/(2*A);
m33 = (x2 -x1)/(2*A);

bee = [m12 0 m22 0 m32 0;0 m13 0 m23 0 m33;m13 m12 m23 m22 m33 m32];%
l=0;
for k=1:nne
    for j=1:nodof
        l=l+1;
        g(l)=nf(connec(i,k),j);
    end
end
end
```



Placa con orificio en el medio



E = 200 000

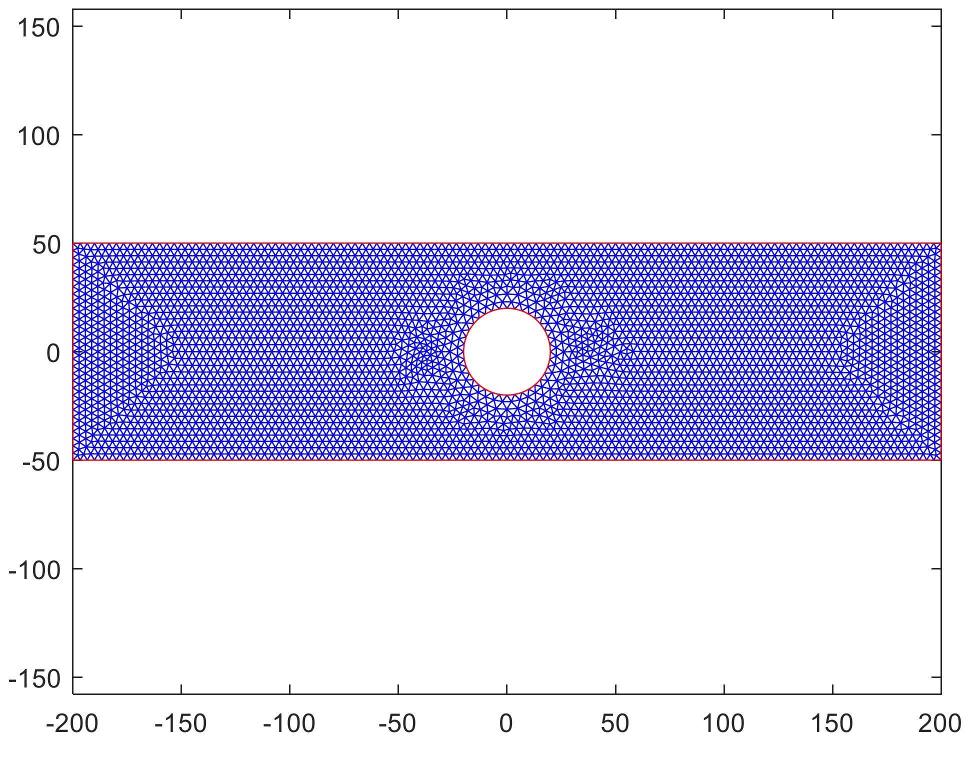
v = 0.25

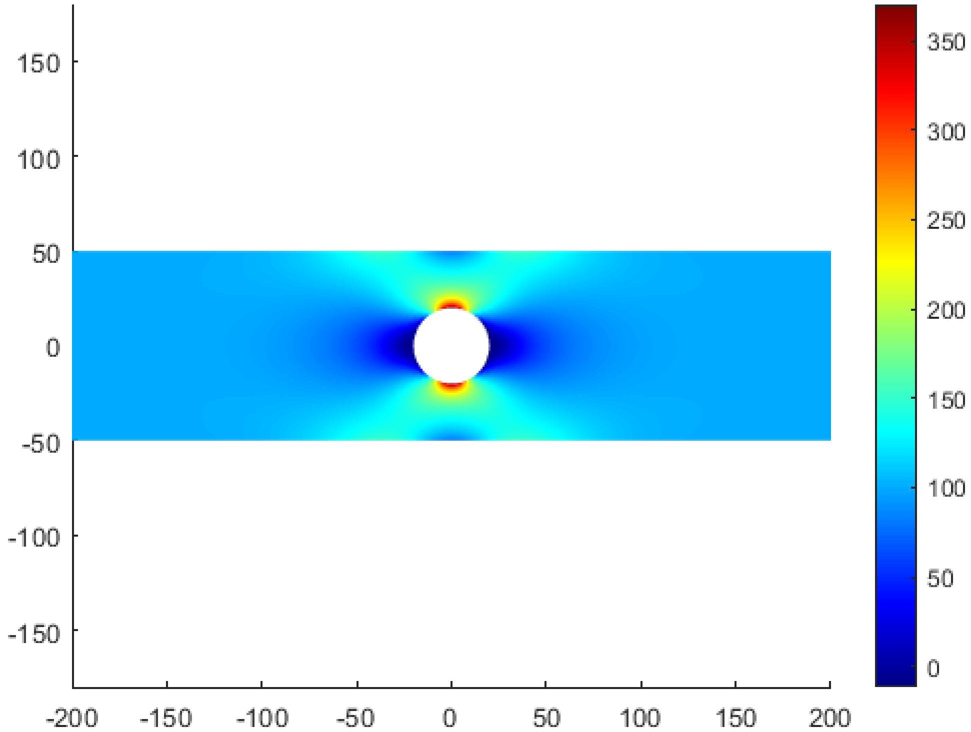
L = 200 , C =50 mm

R = 20 mm

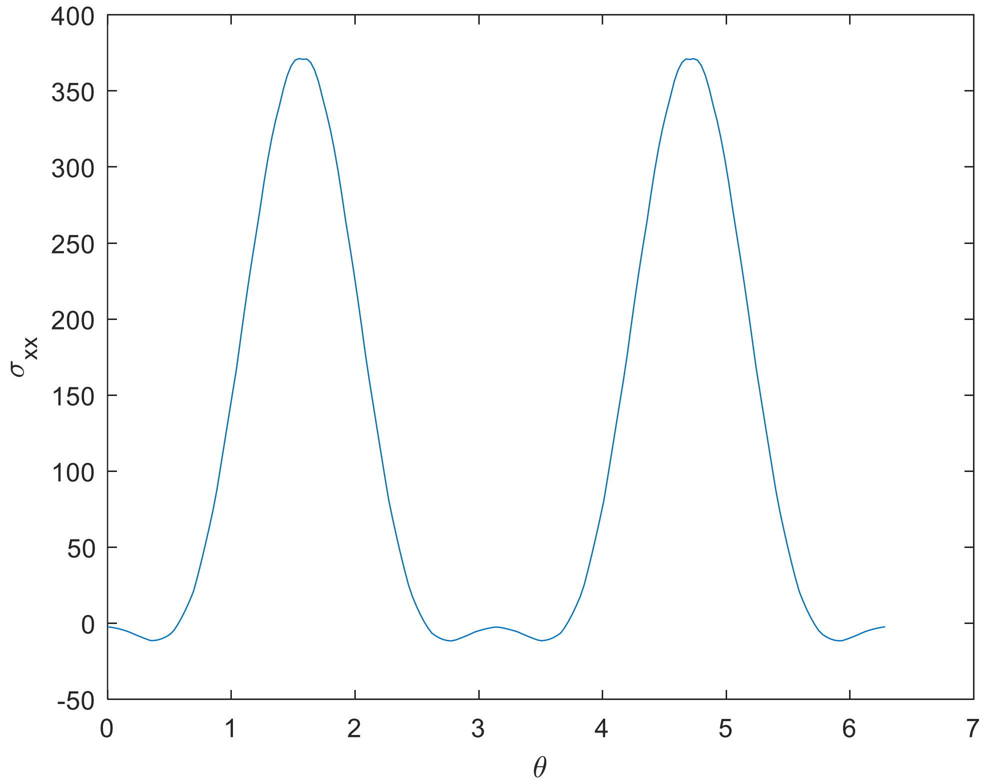
```
model = createpde('structural','static-planestress');

radius = 20.0;
width = 50.0;
Length = 200;

%Geometria
R1 = [3 4 -Length  Length ...
           Length -Length ...
          -width -width width width]';
C1 = [1 0 0 radius 0 0 0 0 0 0]';

gdm = [R1 C1];
ns = char('R1','C1');
g = decsg(gdm,'R1 - C1',ns');

geometryFromEdges(model,g);%incluir la geometria al modelo

%Parámetros
structuralProperties(model,'YoungsModulus',200E3,...
    'PoissonsRatio',0.25);

%Condiciones de borde
structuralBC(model,'Edge',3,'XDisplacement',0);
structuralBC(model,'Vertex',3,'YDisplacement',0);
structuralBoundaryLoad(model,'Edge',1,'SurfaceTraction',[100;0]);

%Malla
generateMesh(model,'Hmax',radius/6);
```

```
figure
pdemesh(model)

R = solve(model);

figure
pdeplot(model,'XYData',R.Stress.sxx,'ColorMap','jet')
axis equal

%Interpolación de esfuerzos
thetaHole = linspace(0,2*pi,200);
xr = radius*cos(thetaHole);
yr = radius*sin(thetaHole);
CircleCoordinates = [xr;yr];

stressHole = interpolateStress(R,CircleCoordinates);

figure
plot(thetaHole,stressHole.sxx)
xlabel('\theta')
ylabel('\sigma_{xx}')
```



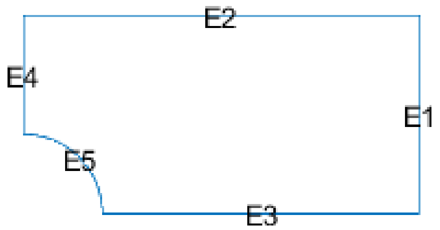


150
100
50
0
-50
-100
-150
-200
-150
-100
-50
0
50
100
150
200
350
300
250
200
150
100
50
0

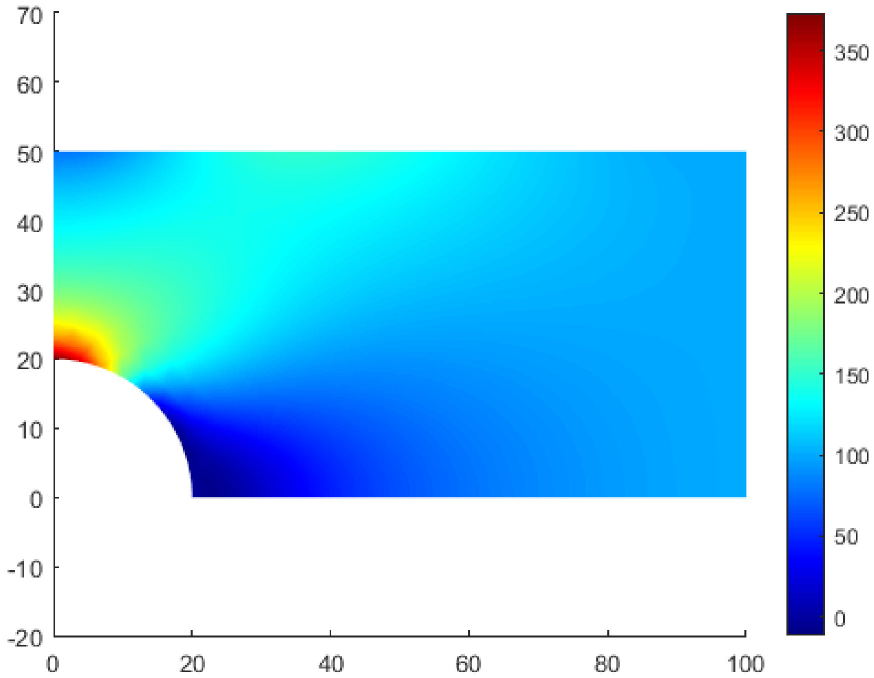


400
350
300
250
200
150
100
50
0
-50
0
1
2
3
4
5
6
7
σ_{xx}
θ

Análisis por simetría : Cuadrante

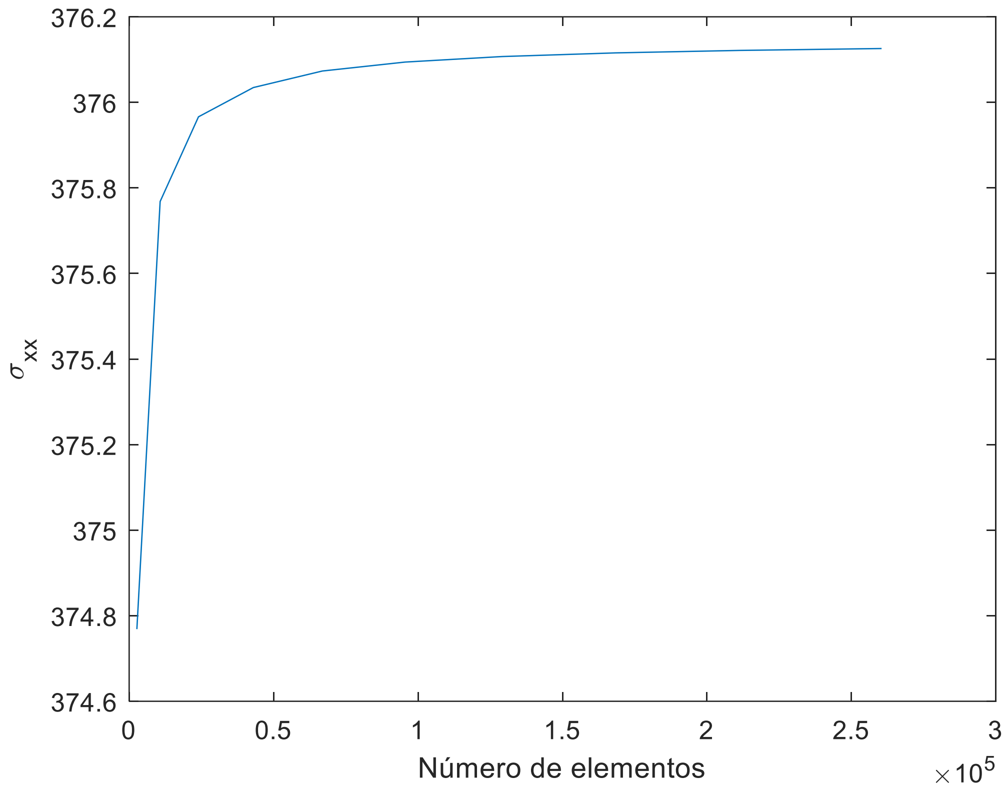



```
symModel = createpde('structural','static-planestress');

radius = 20.0;
width = 50.0;
Length = 200;
R1 = [3 4 0 Length/2 Length/2 ...
      0 0 0 width width]';
C1 = [1 0 0 radius 0 0 0 0 0 0]';
gm = [R1 C1];
sf = 'R1-C1';
ns = char('R1','C1');
g = decsg(gm,sf,ns');
geometryFromEdges(symModel,g);

structuralProperties(symModel,'YoungsModulus',200E3, ...
    'PoissonsRatio',0.25);
structuralBC(symModel,'Edge',[3 4],'Constraint','symmetric');
structuralBoundaryLoad(symModel,'Edge',1,'SurfaceTraction',[100;0]);

generateMesh(symModel,'Hmax',radius/6);
Rsym = solve(symModel);

figure
pdeplot(symModel,'XYData',Rsym.Stress.sxx,'ColorMap','jet');
axis equal
```

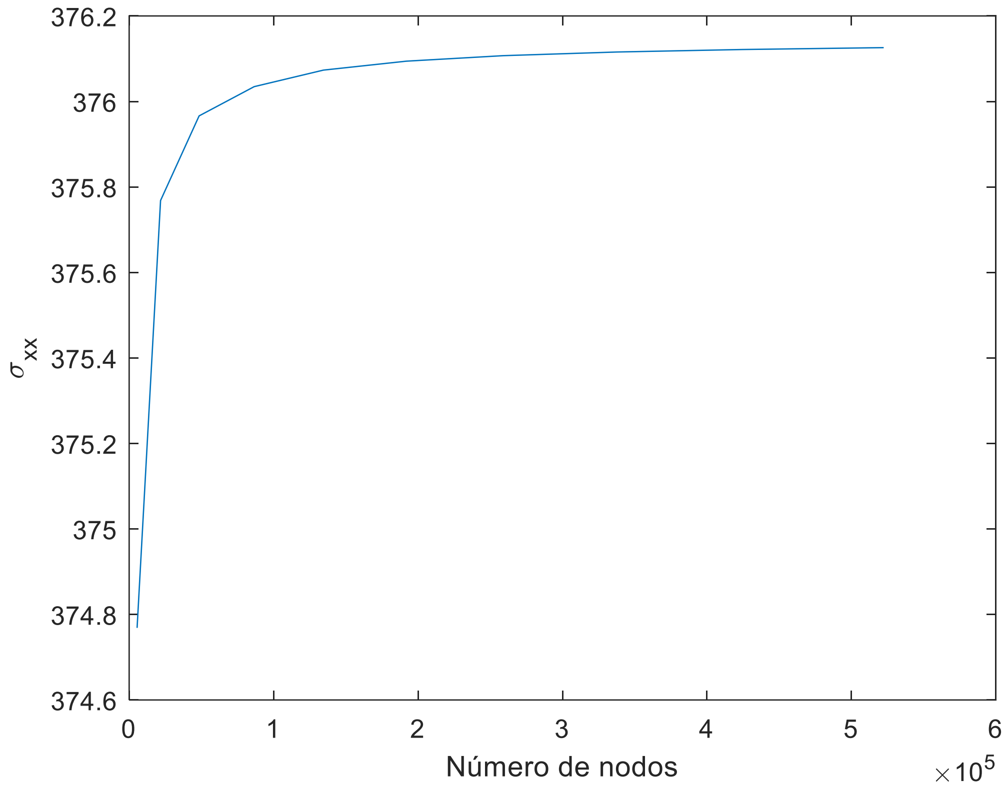


Refinamiento de Malla

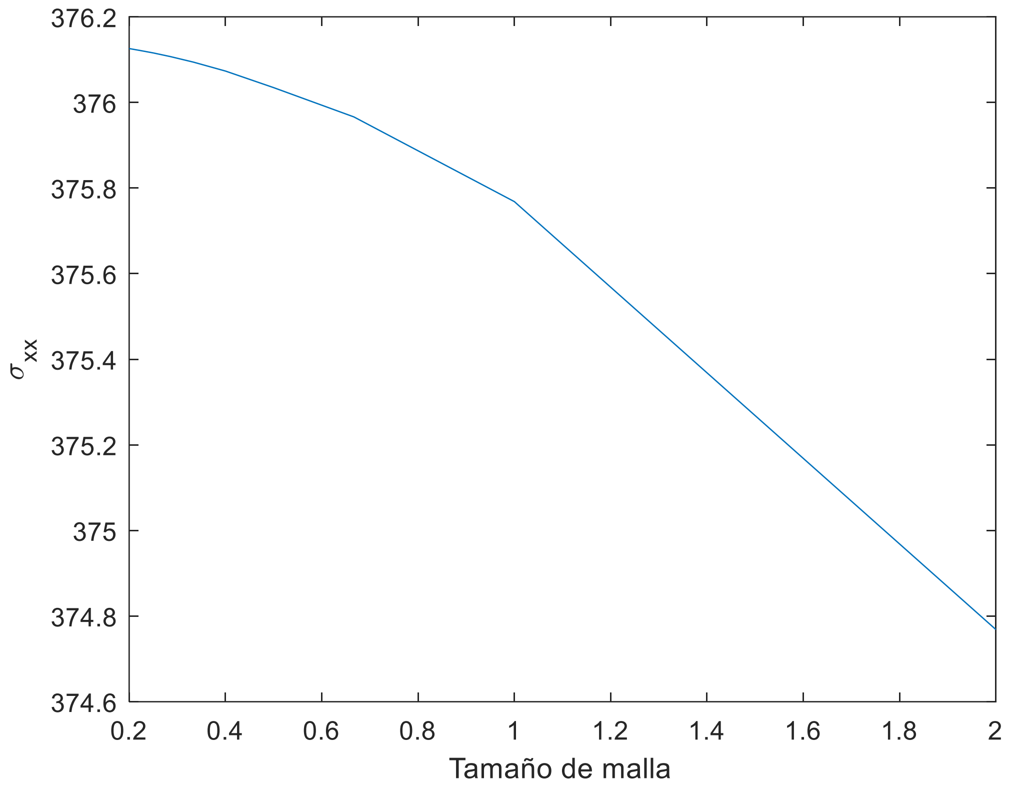
```
symModel = createpde('structural','static-planestress');

radius = 20.0;
width = 50.0;
Length = 200;

R1 = [3 4 0 Length/2 Length/2 ...
      0 0 0 width width]';
C1 = [1 0 0 radius 0 0 0 0 0 0]';
gm = [R1 C1];
sf = 'R1-C1';
ns = char('R1','C1');
g = decsg(gm,sf,ns');
geometryFromEdges(symModel,g);

structuralProperties(symModel,'YoungsModulus',200E3, ...
    'PoissonsRatio',0.25);

structuralBC(symModel,'Edge',[3 4],'Constraint','symmetric');
structuralBoundaryLoad(symModel,'Edge',1,'SurfaceTraction',[100;0]);

for i=10:10:100
    mesh = generateMesh(symModel,'Hmax',radius/i);
    Rsym = solve(symModel);
    [Stress] = Rsym.Stress.sxx;
    [Sigma_max,index_max] = max(Stress);
    sigma((i+2)/4)=Sigma_max;
    nnodos((i+2)/4)=size(mesh.Nodes,2);
    nelemen((i+2)/4)=size(mesh.Elements,2);
    Size((i+2)/4)=radius/i;
End

figure
plot(nelemen,sigma)
xlabel('Número de elementos')
ylabel('\sigma_{xx}')
figure
plot(nnodos,sigma)
xlabel('Número de nodos')
ylabel('\sigma_{xx}')
figure
plot(Size,sigma)
xlabel('Tamaño de malla')
ylabel('\sigma_{xx}')

table(Size',nnodos',nelemen',sigma',...
    'VariableNames',{'Tamaño de malla',...
    'Número de nodos','Número de elementos','?xx'})

mesh = generateMesh(symModel,'Hmax',0.3);
Rsym = solve(symModel);
[Stress] = Rsym.Stress.sxx;
[Sigma_max,index_max] = max(Stress)

table(0.3,size(mesh.Nodes,2),size(mesh.Elements,2),Sigma_max,...
    'VariableNames',{'Tamaño de malla',...
    'Número de nodos','Número de elementos','?xx'})
```



376.2
376
375.8
375.6
375.4
375.2
375
374.8
374.6
σ_{xx}
0
0.5
1
1.5
2
2.5
3
Número de elementos
$\times 10^5$



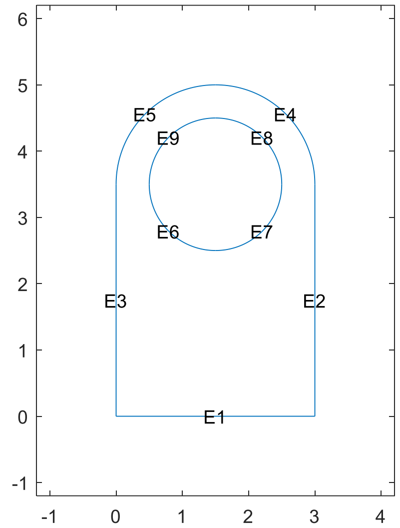
376.2
376
375.8
375.6
375.4
375.2
375
374.8
374.6
σ_{xx}
0
1
2
3
4
5
6
Número de nodos
$\times 10^5$



Tamaño de malla	Número de nodos	Número de elementos	σxx
2	5499	2676	374.77
1	21746	10727	375.77
0.66667	48428	23995	375.97
0.5	86554	42985	376.03
0.4	1.345e+05	66885	376.07
0.33333	1.9193e+05	95526	376.09
0.28571	2.5896e+05	1.2897e+05	376.11
0.25	3.3788e+05	1.6836e+05	376.12
0.22222	4.2415e+05	2.1142e+05	376.12
0.2	5.2234e+05	2.6044e+05	376.13

El valor el esfuerzo máximo converge a 376.1
Escogiendo una malla de 0.3

Tamaño de malla	Número de nodos	Número de elementos	σxx
0.3	2.3652e+05	1.1777e+05	376.1

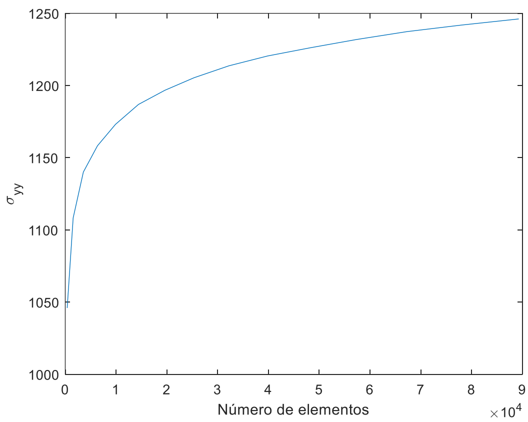


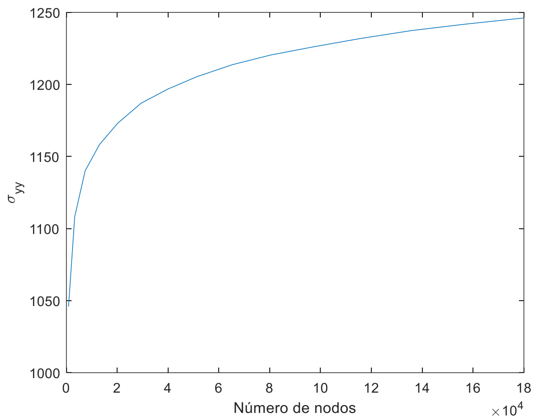
E = 30 E6

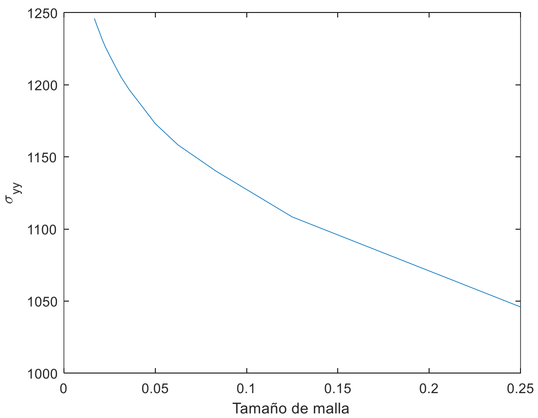
v = 0.3

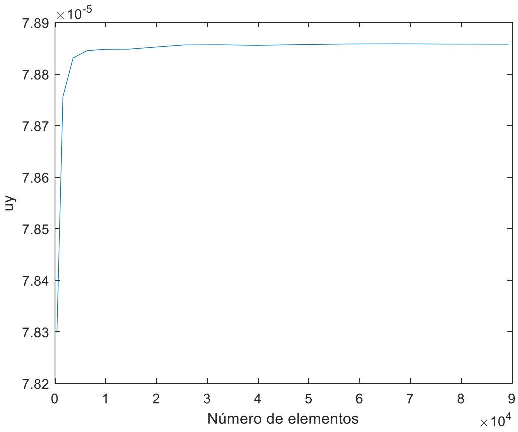
```
model = createpde('structural','static-planestress');

R1 = [3 4 0 3 3 0 0 0 3.5 3.5]';
C1 = [1 1.5 3.5 1.5 0 0 0 0 0 0]';
C2 = [1 1.5 3.5 1 0 0 0 0 0 0]';

gdm = [R1 C1 C2];
ns = char('R1','C1','C2');
[g,dl] = decsg(gdm,'(R1 + C1) - C2',ns');
h=csgdel(g,dl);

geometryFromEdges(model,h);

structuralProperties(model,'YoungsModulus',30E6,...
    'PoissonsRatio',0.3);

structuralBC(model,'Edge',1,'Constraint','fixed');
structuralBoundaryLoad(model,'Edge',[8,9],'SurfaceTraction',[0;100]);

for i=4:4:60
    mesh = generateMesh(model,'Hmax',1/i);
    R = solve(model);
    [Stress] = R.Stress.syy;
    [Sigma_max,i_max] = max(Stress);
    [Dy] = R.Displacement.y;
    [Uy_max,j_max] = max(Dy);
    sigma(i/4)=Sigma_max;
    uy(i/4)=Uy_max;
    nnodos(i/4)=size(mesh.Nodes,2);
    nelemen(i/4)=size(mesh.Elements,2);
    Size(i/4)=1/i;
    i
end

figure
plot(nelemen,sigma)
xlabel('Número de elementos')
ylabel('\sigma_{yy}')
figure
plot(nnodos,sigma)
```

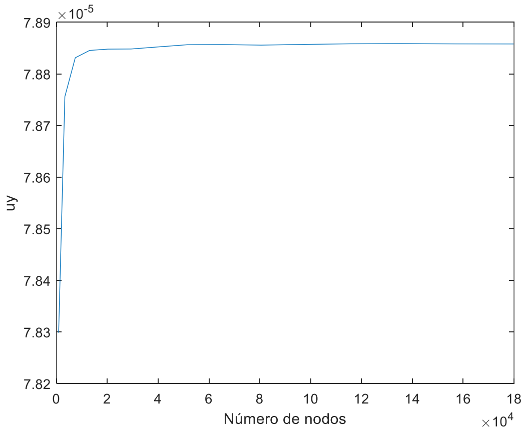
```
xlabel('Número de nodos')
ylabel('\sigma_{yy}')
figure
plot(Size,sigma)
xlabel('Tamaño de malla')
ylabel('\sigma_{yy}')

figure
plot(nelemen,uy)
xlabel('Número de elementos')
ylabel('uy')
figure
plot(nnodos,uy)
xlabel('Número de nodos')
ylabel('uy')
figure
plot(Size,uy)
xlabel('Tamaño de malla')
ylabel('uy')

table(Size',nnodos',nelemen',sigma',uy',...
    'VariableNames',{'Tamaño de malla',...
    'Número de nodos','Número de elementos','?yy','uy'})
```

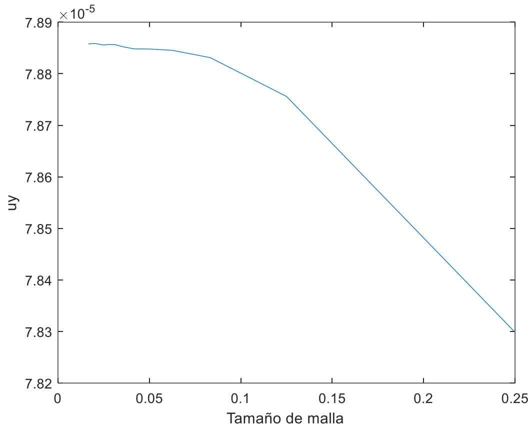












Tamaño de malla	Número de nodos	Número de elementos	σyy	uy
0.25	846	382	1045.9	7.8299e-05
0.125	3290	1560	1108.4	7.8756e-05
0.083333	7384	3566	1140.1	7.8831e-05
0.0625	13020	6342	1158.3	7.8845e-05
0.05	20190	9886	1173.1	7.8848e-05
0.041667	29346	14420	1186.9	7.8848e-05
0.035714	39824	19618	1196.7	7.8852e-05
0.03125	51414	25372	1205.4	7.8856e-05
0.027778	65206	32224	1213.7	7.8857e-05
0.025	80500	39830	1220.4	7.8856e-05
0.022727	97576	48326	1226.2	7.8857e-05
0.020833	1.1563e+05	57310	1231.8	7.8858e-05
0.019231	1.3571e+05	67308	1237.3	7.8859e-05
0.017857	1.5719e+05	78008	1241.9	7.8858e-05
0.016667	1.7986e+05	89300	1246.1	7.8858e-05

Para malla 0.01

Tamaño de malla	Número de nodos	Número de elementos	σyy	uy
0.01	4.9616e+05	2.4703e+05	1278.9	7.8859e-05

```
mesh = generateMesh(model,'Hmax',0.01);
    R = solve(model);
    [Stress] = R.Stress.syy;
    [Sigma_max,i_max] = max(Stress);
    [Dy] = R.Displacement.y;
    [Uy_max,j_max] = max(Dy);

table(0.01,size(mesh.Nodes,2),size(mesh.Elements,2),Sigma_max,...
    Uy_max,'VariableNames',{'Tamaño de malla',...
    'Número de nodos','Número de elementos','?yy','uy'})

figure
pdeplot(model,'XYData',R.Stress.syy,'ColorMap','jet')
axis equal

figure
pdeplot(model,'XYData',R.Displacement.y,'ColorMap','jet')
axis equal
```