

## %% Códigos

```
%% Nodos y elementos
cnod = [0 0 0
        0 0 1];
nnod = size(cnod,1);
elem = [1 2
        2 3];
nelem = size(elem,1);

%% Usar find
pos2=find(abs(sig)==max(abs(sig)));

%% Matrices de rigidez

% Vector director
v=cnod(elem(e,2),:)-cnod(elem(e,1),:);
long=norm(v);
vd=v/long;

memlong(e) = long;
memvecdirec(e,:) = vecdirec;

% Barra 1D
Kbarra=E*A/long*[1 -1; -1 1];

% Resorte
Kresorte=k*[1 -1; -1 1];

% Viga 2D
X=A*E/long;
Y4=2*E*I/long;
Y3=Y4*2;
Y2=Y4*3/long;
Y1=Y2*2/long;
Kbarra=X*[1 -1; -1 1];
Kviga=[Y1 Y2 -Y1 Y2
        Y2 Y3 -Y2 Y4
        -Y1 -Y2 Y1 -Y2
        Y2 Y4 -Y2 Y3];
```

```

% Viga 3D
X = A(e)*E/long;
Y4 = 2*E*Iz(e)/long;
Y3 = Y4*2;
Y2 = 3*Y4/long;
Y1 = 2*Y2/long;
Z4 = 2*E*Iy(e)/long;
Z3 = Z4*2;
Z2 = 3*Z4/long;
Z1 = 2*Z2/long;
S = G*J(e)/long;
Klocal = [X 0 0 0 0 0 -X 0 0 0 0 0
           0 Y1 0 0 0 Y2 0 -Y1 0 0 0 Y2
           0 0 Z1 0 -Z2 0 0 0 -Z1 0 -Z2 0
           0 0 0 S 0 0 0 0 0 -S 0 0
           0 0 -Z2 0 Z3 0 0 0 Z2 0 Z4 0
           0 Y2 0 0 0 Y3 0 -Y2 0 0 0 Y4
           -X 0 0 0 0 0 X 0 0 0 0 0
           0 -Y1 0 0 0 -Y2 0 Y1 0 0 0 -Y2
           0 0 -Z1 0 Z2 0 0 0 Z1 0 Z2 0
           0 0 0 -S 0 0 0 0 0 S 0 0
           0 0 -Z2 0 Z4 0 0 0 Z2 0 Z3 0
           0 Y2 0 0 0 Y4 0 -Y2 0 0 0 Y3];

%% Rotar

%Barras 2D
T=[vd 0 0 ; 0 0 vd];

% Barras 3D
T=[vd 0 0 0; 0 0 0 vd];

% Vigas 2D
dofr=[dof(elem(e,1),:) dof(elem(e,2),:)]';
lambda=[vd(1) vd(2) 0; -vd(2) vd(1) 0; 0 0 1];
T=blkdiag(lambda,lambda);
Kelem=T'*Klocal*T;
Kglobal(dofr,dofr)=Kglobal(dofr,dofr)+Kelem;

% Vigas 3D
%function Ke=rotar(Kl,p1,p2,p3)
v1 = p2-p1;
vd1 = v1/norm(v1);
vp = p3-p1;
vd3 = cross(vd1,vp)/norm(cross(vd1,vp));
vd2 = cross(vd3,vd1);
lambda = [ vd1 ; vd2 ; vd3 ];
T=blkdiag(lambda,lambda,lambda,lambda);
Ke = T.'*Kl*T;
end

```

```

%% Dof
dofpornodo=3;
dof=reshape([1:1:dofpornodo*nnod]',dofpornodo,nnod)';

%% BC
fijo=[0 0 0; 1 1 0; 1 1 1];
libre=~fijo;
libre=reshape(libre.',[],1);

%% Funciones de forma
% Barras
DPorNodo = reshape(D,DofPorNodo,NumeroNodos)';
Sigma = zeros(NumeroElementos,1);
for e = 1:NumeroElementos
    Delemento = [DPorNodo(Elementos(e,1),:) DPorNodo(Elementos(e,2),:)]';
    Dlocal = MemT(:, :, e)*Delemento;
    B = [-1 1]/Memlong(e);
    Sigma(e) = E*B*Dlocal;
end
SigmaMaxAbs = max(abs(Sigma));
Elemento = find(Sigma==SigmaMaxAbs);
SigmaMax = Sigma(Elemento);

% Vigas
DPorNodo = reshape(D,DofPorNodo,NumeroNodos)';
SigmaAxial = zeros(NumeroElementos,1);
SigmaFlexSup = zeros(NumeroElementos,30);
SigmaTotalSup = SigmaFlexSup;
SigmaTotalInf = SigmaFlexSup;
Memsub = SigmaFlexSup;
for e = 1:NumeroElementos
    Delemento = [DPorNodo(Elementos(e,1),:) DPorNodo(Elementos(e,2),:)]';
    Dlocal = MemT(:, :, e)*Delemento;
    Ba = [-1 1]/Memlong(e);
    SigmaAxial(e) = E*Ba*Dlocal([1 4]);
    sub = 0:Memlong(e)/29:Memlong(e);
    Memsub(e,:) = sub;
    N1=@(x) -6/Memlong(e)^2+12*x/Memlong(e)^3;
    N2=@(x) -4/Memlong(e)+6*x/Memlong(e)^2;
    N3=@(x) 6/Memlong(e)^2-12*x/Memlong(e)^3;
    N4=@(x) -2/Memlong(e)+6*x/Memlong(e)^2;
    Bf = [N1(sub); N2(sub); N3(sub); N4(sub)]';
    SigmaFlexSup(e,:) = (b/2*E*Bf*Dlocal([2 3 5 6]))';
    SigmaFlexInf = -SigmaFlexSup;
    SigmaTotalSup(e,:) = SigmaFlexSup(e, :)+SigmaAxial(e);
    SigmaTotalInf(e,:) = SigmaFlexInf(e, :)+SigmaAxial(e);
end
SigmaCompuesto = [(SigmaTotalSup); (SigmaTotalInf)];
SigmaMax = max(max(abs(SigmaCompuesto)));
[i,j] = find(abs(SigmaCompuesto)==SigmaMax);

%Deformación en vigas
N1=@(x) 1-3*(x/long).^2+2*(x/long).^3;
N2=@(x) x-2*x.^2/long+x.^3/long^2;
N3=@(x) 3*(x/long).^2-2*(x/long).^3;
N4=@(x) -x.^2/long +x.^3/long^2;

```

```

%% BC
% El desplazamiento conocido es uan condicion de borde
fijo = 1&[1 1 1 0 1 0];
libre = ~fijo;
%% Cargas conocidas
P = [0 0 0 M2 0 0]';
%% Desplazamientos conocidos
D = [0 0 0 0 U3 0]';
%% Reduccion
Kglobalx = Kglobal(libre,libre);
Kglobalcx = Kglobal(libre,fijo);
Kglobalc = Kglobal(fijo,fijo);
Pc = P(libre);
Dc = D(fijo);
%% Solver
Dx = Kglobalx\(Pc-Kglobalcx*Dc);
D(libre) = Dx;
P = Kglobal*D;

```