

ELEMENTOS FINITOS I

PARCIAL I

GUÍA DE EXÁMEN

FEDERICO A. HOJMAN
55143

Índice

1. Inicio	1
2. Información teórica	1
2.1. Generalidades	1
2.2. Cargas equivalentes	2
2.3. Rotación de elementos	2
2.3.1. Barras	2
2.3.2. Vigas planas	2
2.3.3. Viga 3D	3
2.4. Simetría y antisimetría	3
3. Código	5
3.1. Matriz DOF	5
3.2. Matriz de barras	5
3.3. Matriz de vigas	5
3.4. Viga + barra	5
3.5. Ciclo de resolución; vigas axiales	5
3.6. Tensiones	6
4. Revisión del examen	7

1. Inicio

- Leer atentamente qué piden. No hacer código innecesario
- Verificar simetrías, siempre ponen alguna y simplifica mucho los ejercicios
- Definir los tipos de elemento apropiados (barras, vigas planas, vigas planas con desplazamientos axiales o vigas 3D)
- Si no corre el programa no es bueno pero hay que seguir programando. No perder el tiempo

1. Datos
2. Coordenas
3. Matriz de elementos, nNodes, nElements
4. Condiciones de borde
5. Cargas (si es posible)
6. Matriz DOF
7. Matriz global de zeros
8. Ensamble
9. Resolución de desplazamientos
10. Tensiones

2. Información teórica

2.1. Generalidades

- Linealidad: todo el método es lineal. Si multiplico cualquiera de las matrices por un coeficiente, el sistema responde haciendo lo mismo con las demás. Ej: Si duplico las cargas, se duplican los desplazamientos. Si duplico las rigideces, divido en dos los desplazamientos.
- Las tensiones se calculan siempre en el sistema local de los elementos
- Toda carga va sobre los nodos

2.2. Cargas equivalentes

	f_{1y}	m_1	Loading case	f_{2y}	m_2
1.	$-\frac{P}{2}$	$-\frac{PL}{8}$		$-\frac{P}{2}$	$\frac{PL}{8}$
2.	$-\frac{Pb^2(L+2a)}{L^3}$	$-\frac{Pab^2}{L^2}$		$-\frac{Pa^2(L+2b)}{L^3}$	$\frac{Pa^2b}{L^2}$
3.	$-P$	$-\alpha(1-\alpha)PL$		$-P$	$\alpha(1-\alpha)PL$
4.	$-\frac{wL}{2}$	$-\frac{wL^2}{12}$		$-\frac{wL}{2}$	$\frac{wL^2}{12}$
5.	$-\frac{7wL}{20}$	$-\frac{wL^2}{20}$		$-\frac{3wL}{20}$	$\frac{wL^2}{30}$
6.	$-\frac{wL}{4}$	$-\frac{5wL^2}{96}$		$-\frac{wL}{4}$	$\frac{5wL^2}{96}$
7.	$-\frac{13wL}{32}$	$-\frac{11wL^2}{192}$		$-\frac{3wL}{32}$	$\frac{5wL^2}{192}$
8.	$-\frac{wL}{3}$	$-\frac{wL^2}{15}$		$-\frac{wL}{3}$	$\frac{wL^2}{15}$
9.	$-\frac{M(a^2+b^2-4ab-L^2)}{L^3}$	$\frac{Mb(2a-b)}{L^2}$		$\frac{M(a^2+b^2-4ab-L^2)}{L^3}$	$\frac{Ma(2b-a)}{L^2}$

Figura 2.1: Cargas nodales equivalentes

2.3. Rotación de elementos

$$[K_{global}] = [T]^T [K_{local}] [T] \quad (2.1)$$

2.3.1. Barras

$$\frac{V_{12}}{||V_{12}||} = \{l_1 m_1\}; T = \begin{bmatrix} l_1 & m_1 & 0 & 0 \\ 0 & 0 & l_1 & m_1 \end{bmatrix} \quad (2.2)$$

2.3.2. Vigas planas

Si la tengo que rotar, tengo que ponerle la parte de barras también

$$\frac{V_{12}}{||V_{12}||} = \{l_1 m_1\}; \Lambda = \begin{bmatrix} l_1 & m_1 & 0 \\ -m_1 & l_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$T_{6 \times 6} = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda \end{bmatrix} \quad (2.4)$$

2.3.3. Viga 3D

Lleva 3 versores directores. El primero es el vector director de la viga normalizado V_1 . Luego se sigue de la siguiente manera:

Definiendo un punto auxiliar, se fija el plano $x-y$ de la viga y se propone un vector V_{Aux} que una al punto de origen de la viga y al auxiliar. Se tiene entonces que

$$v_1 = \frac{V_1}{||V_1||}; v_3 = \frac{V_1 \times V_{Aux}}{||V_1 \times V_{Aux}||}; v_2 = v_3 \times v_1 \quad (2.5)$$

Se acomodan estos vectores de forma tal que se obtenga una matriz de rotación T :

$$\Lambda = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}; T_{12 \times 12} = \begin{bmatrix} \Lambda & 0 & 0 & 0 \\ 0 & \Lambda & 0 & 0 \\ 0 & 0 & \Lambda & 0 \\ 0 & 0 & 0 & \Lambda \end{bmatrix} \quad (2.6)$$

2.4. Simetría y antisimetría

La simetría restringe, en los nodos contenidos sobre el plano de simetría, los desplazamientos a través del plano y ambos giros perpendiculares. La antisimetría tiene el efecto inverso. Si hay simetría estructural sin simetría de

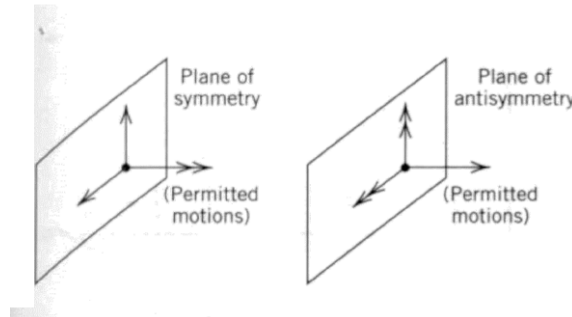


Figura 2.2: Movimientos permitidos

cargas, se puede resolver el caso simétrico y el antisimétrico y superponerlos, dado que el método es absolutamente lineal.

3. Código

3.1. Matriz DOF

```
DOF = reshape(1:1:(nNodesBeam)*dofxNode,dofxNode,[])';
```

3.2. Matriz de barras

```
barK = (E*A/L)*[1 -1;-1 1];
```

3.3. Matriz de vigas

```
beamK = (E*I/(L^3))*[ 12    6*L   -12   6*L
                      6*L    4*L^2 -6*L  2*L^2
                      -12   -6*L   12   -6*L
                      6*L    2*L^2 -6*L  4*L^2];
```

3.4. Viga + barra

```
localK = zeros(6);
```

```
beamK = (E*I/(L^3))*[ 12    6*L   -12   6*L
                      6*L    4*L^2 -6*L  2*L^2
                      -12   -6*L   12   -6*L
                      6*L    2*L^2 -6*L  4*L^2];
```

```
barK = (E*A/L)*[1 -1;-1 1];
```

```
localK([2 3 5 6],[2 3 5 6]) = beamK;
localK([1 4],[1 4]) = barK;
```

3.5. Ciclo de resolución; vigas axiales

```
for(e = 1:nElements)
    v = nodeCoordinates(elements(e,2),:)-nodeCoordinates(elements(e,1),:);
    L = norm(v);
    vectLong(e) = L;
    cosDir = v/L;
    lambda = [cosDir(1) cosDir(2) 0;-cosDir(2) cosDir(1) 0; 0 0 1];
    T = blkdiag(lambda,lambda);
```

```

vecT(:, :, e) = T;
localK = zeros(6);

beamK = (E*I/(L^3))*[ 12    6*L   -12   6*L
                     6*L    4*L^2 -6*L  2*L^2
                    -12   -6*L   12   -6*L
                     6*L    2*L^2 -6*L  4*L^2];
barK = (E*A/L)*[1 -1; -1 1];

localK([2 3 5 6], [2 3 5 6]) = beamK;
localK([1 4], [1 4]) = barK;

dofs = [beamDOF(elements(e,1), :) beamDOF(elements(e,2), :)];

globalK(dofs, dofs) = globalK(dofs, dofs) + T'*localK*T;

end

redD = globalK(free, free)\load(free);
D = zeros(3*nNodesBeam, 1);
D(free) = redD

```

3.6. Tensiones

```

%% Barras

sigma = zeros(nElements, 1);

for( e = 1:nElements)

    localD = D([DOF(elements(e,1), :) DOF(elements(e,2), :)]);
    localD = rotationVector(:, :, e)*localD;
    sigma(e) = ( E/vectorLargos(e))*[-1 1]*localD;

end

%% Vigas

for(e = 1:nElements)

```



```

L = vectorLongitudes(e);
B = [-(1/L) 1/L];

% Barras

barDisplacements = [barD(elements(e,1),:) barD(elements(e,2),:)]';
localBarDisplacements = barTvector(:, :, e)*barDisplacements;
sigmaBar(e) = E*B*localBarDisplacements;

%Vigas

beamDisp = [beamD(elements(e,1),:) beamD(elements(e,2),:)]';

localD = beamTvector(:, :, e)*beamDisp;
localBeamAx = localD([1 4]);

sigmaN = E*(1/L)*[-1 1]*localD([1 4]);

sigmaM = @(x) (E*h/2)*([(-6/L^2)+(12*x/L^3)
(-4/L)+(6*x/L^2)
((6/L^2)-(12*x)/L^3)
((-2/L)+(6*x/L^2))])'*localD([2 3 5 6]);

sigmaMax = sigmaN + sign(sigmaN)*max(max((abs([sigmaM(0) sigmaM(L)]))))

end

```

4. Revisión del examen

- Reshapes
- Reducción de la matriz global: $K(\text{free}, \text{free})$, no $K(\text{free})$
- Matrices de rotación
- Verificar que K global no sea inversible
- Verificar sistema de unidades
- Verificar que haya suma en el ensamble de la matriz global ($K(x, x) = K(x, x) + \text{local}$)