Basic example of 1-dimension finite elements

Develop a code to calculate the displacement of the nodes and plot them in the original and displaced positions of a bar element with 4 nodes, the first and last nodes are restricted and a load of 3000 N is applied to the second node. Data:

$E = 30e6\ N/m^2$;

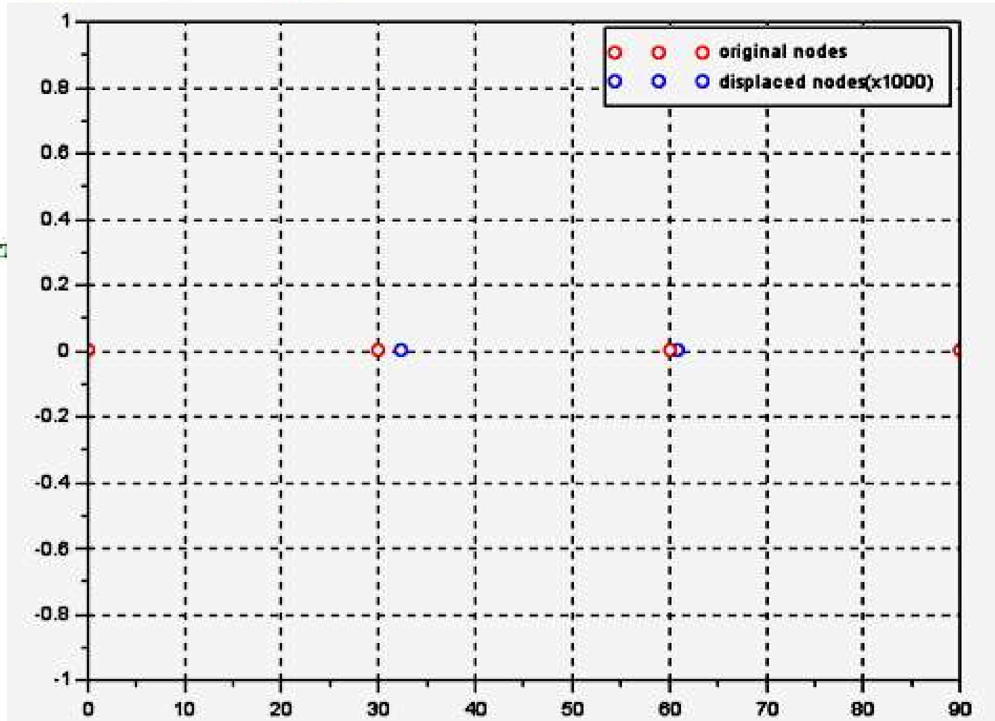
$A=1\ m^2$;

$L = 90\ m$;

$$[k]^e = \begin{bmatrix} \frac{AE}{L} & -\frac{AE}{L} \\ -\frac{AE}{L} & \frac{AE}{L} \end{bmatrix}$$

Computational implementation proposal

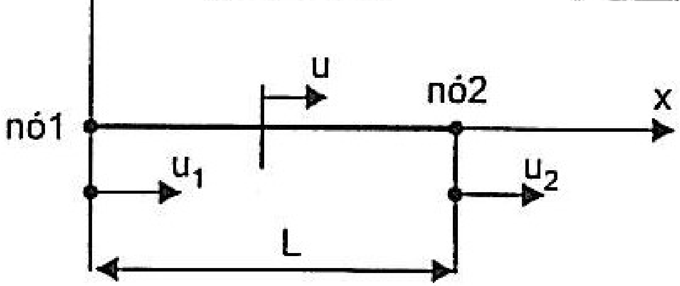
```
clear;clc;
// input data
E = 30000000;A = 1;EA = E*A;L = 90;
numx = 3;// number of elements
node = linspace(0,L,numx+1);
// Define nodes, elements and DoFs
    n=1:numx;
    element(:,1)=n;
    element(:,2)=element+1;  // define elements in sequential order
    numnode=size(node,2);
    numelem=size(element,1);
    ndof = numnode;
// Stiffness matrix
K=zeros(numnode,numnode);
for e=1:numelem;  // cover all elements
    j=element(e,:); // index nodes to elemen
    length_element=node(j(2))-node(j(1));
   k=(EA/length_element)*[1 -1;-1 1];
K(j,j)=K(j,j)+k;
end;
//loads vector
f = zeros(numnode,1);
f(2) = 3000;
U = K\f; // displacement
// to plot nodes
Y = node'+U*1000;
plot(node(1),0,"ro");
plot(Y(2),0,"bo");plot(Y(3),0,"bo");
plot(node(2),0,"ro");plot(node(3),0,"ro");plot(node(4),0,"ro");
hl=legend(['original nodes','displaced nodes(x1000)']);
xgrid
```



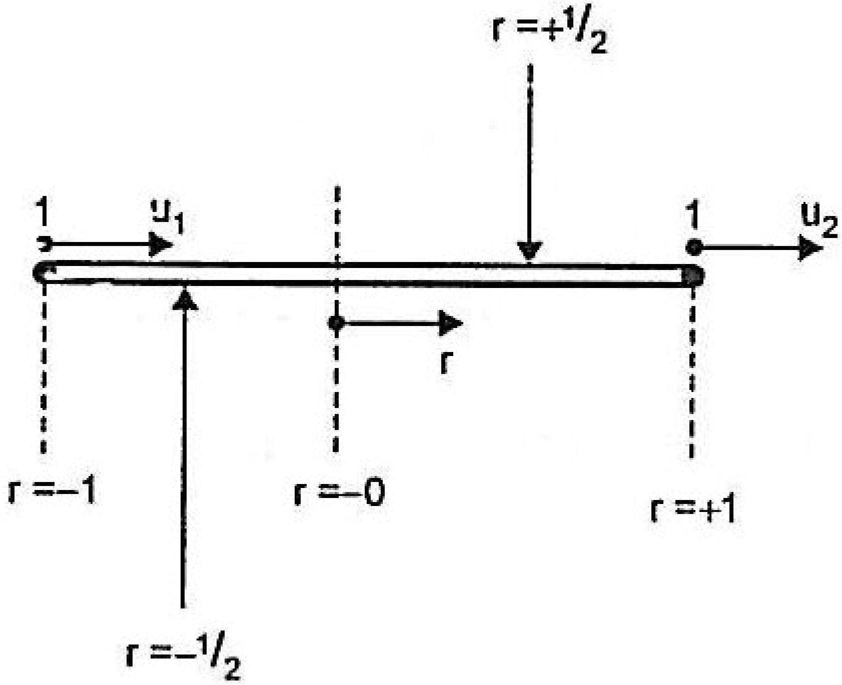
Isoparametric formulation for finite elements

Interpolation function

$$u(x) = u_1 + \frac{u_2 - u_1}{L} \cdot x$$



Isoparametric



Shape function (h_1 e h_2)

$$u(x) = h_1 \cdot u_1 + h_2 \cdot u_2 = \sum_{i=1}^{2} h_i \cdot u_i$$

$\Leftarrow$

$$u(x) = \left(\frac{1-r}{2}\right) \cdot u_1 + \left(\frac{1+r}{2}\right) \cdot u_2$$

Isoparametric formulation for finite elements

$$\boxed{\varepsilon = \frac{\partial u}{\partial x}}$$ strain

By the chain rule, we have:

$$\varepsilon = \frac{\partial u}{\partial x} = \frac{\partial u}{\partial r} \cdot \frac{\partial r}{\partial x}$$

being:

$$\begin{cases} u = \dfrac{1-r}{2} \cdot u_1 + \dfrac{1+r}{2} \cdot u_2 \\ x = \dfrac{1-r}{2} \cdot x_1 + \dfrac{1+r}{2} \cdot x_2 \end{cases}$$

$$\begin{cases} \dfrac{\partial u}{\partial r} = -\dfrac{u_1}{2} + \dfrac{u_2}{2} = \dfrac{u_2 - u_1}{2} \\ \dfrac{\partial r}{\partial x} = \dfrac{1}{\dfrac{\partial x}{\partial r}} = \dfrac{1}{-\dfrac{x_1}{2} + \dfrac{x_2}{2}} = \dfrac{1}{\dfrac{(x_2 - x_1)}{2}} = \dfrac{1}{L/2} = \dfrac{2}{L} \end{cases} \quad (1)$$

$$\varepsilon = \frac{\partial u}{\partial r} \cdot \frac{\partial r}{\partial x} = \frac{u_2 - u_1}{2} \cdot \frac{2}{L}$$

$$\boxed{\varepsilon = \begin{bmatrix} -\dfrac{1}{L} & \dfrac{1}{L} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}}$$

Isoparametric formulation for finite elements

$$[B]=\begin{bmatrix} \frac{-1}{L} & \frac{1}{L} \end{bmatrix} \quad e \quad [B]^T=\begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix}$$

$$[k]^e = \int_{vol} [B]^T \cdot [D] \cdot [B] \cdot dvol$$

$$[k]^e = \int_{x_1}^{x_2} [B]^T \cdot [D] \cdot [B] \cdot A \cdot dx \quad (2)$$

$$\begin{cases} dvol = A \cdot dx \\ [D] = E \quad \text{(uniaxial stress state)} \\ \\ \textbf{From Eq.1:} \frac{\partial r}{\partial x} = \frac{2}{L} \therefore \boxed{\partial x = \frac{L}{2} \cdot \partial r} \quad (3) \end{cases}$$

Equation 3 relates the length *dx* (cartesian system) and the length *dr* (natural coordinate system).

The domain of Integral (Equation 2) in natural coordinates is given by the range from -1 to +1, according to the equation below.

$$[k]^e = \int_{-1}^{+1} \begin{bmatrix} -1/L \\ 1/L \end{bmatrix} \cdot E \cdot \begin{bmatrix} \frac{-1}{L} & \frac{1}{L} \end{bmatrix} \cdot A \cdot \frac{L}{2} \cdot \partial r$$

where:

$$\partial x = J \cdot \partial r$$

$$\boxed{[k]^e = \begin{bmatrix} \frac{AE}{L} & -\frac{AE}{L} \\ -\frac{AE}{L} & \frac{AE}{L} \end{bmatrix}}$$

$$\boxed{J = \frac{L}{2}}$$

Example in Scilab

Develop a code to calculate stresses and plot nodes displacement in the original and displaced positions of an **isoparametric** bar element with 4 nodes. The first and last nodes are restricted and a load of 3000 N is applied to the second node.

Data:

$E = 30e6$ N/m^2;

$A = 1$ m^2;

$L = 90$ m.

Computational implementation proposal

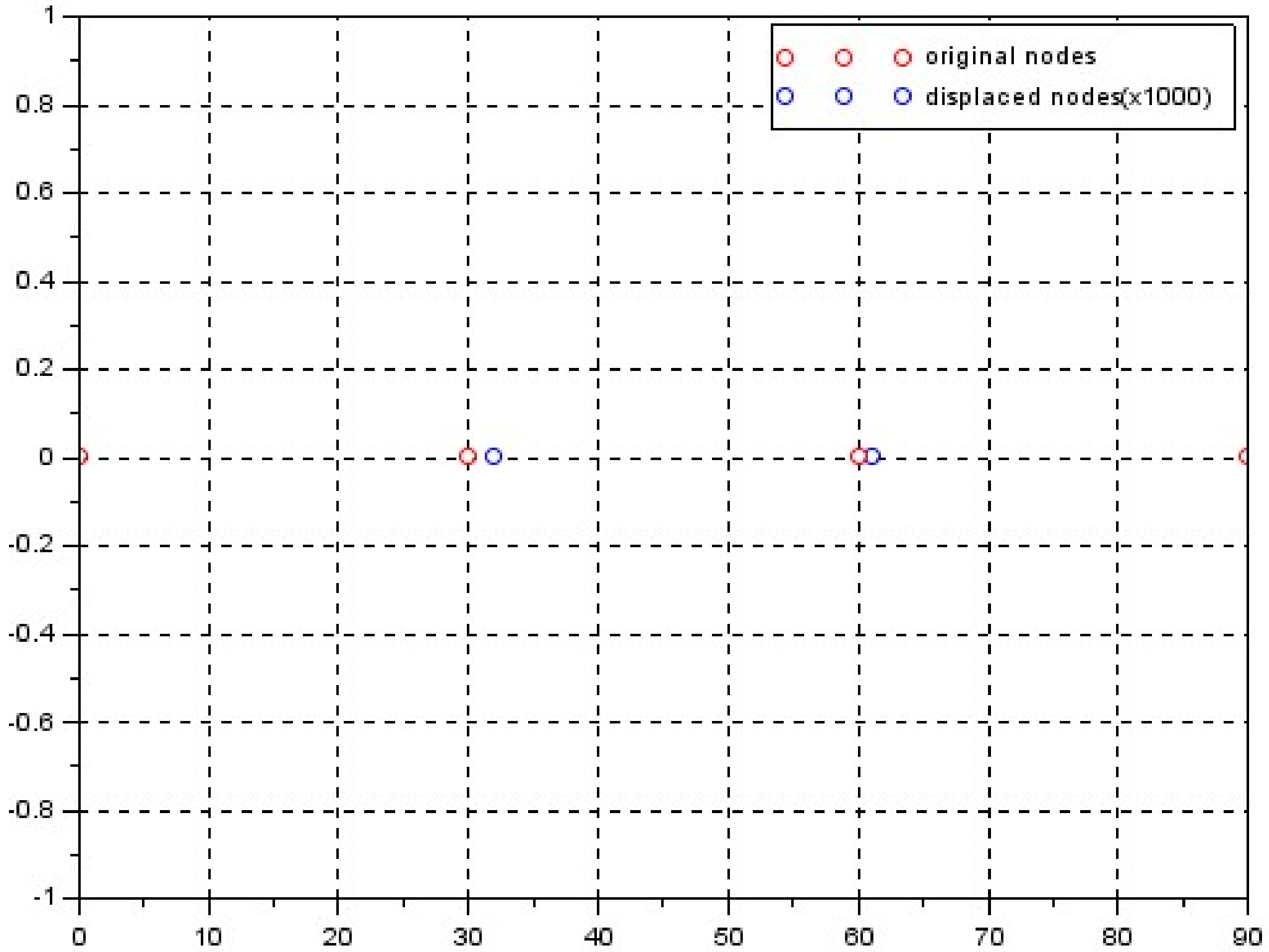
```
clear;clc;
// input data
E = 30000000;A = 1;EA = E*A;L = 90;
numx = 3;// number of elements
node = linspace(0,L,numx+1);
node1=[(node(1:4))', [0 0 0 0 ]'];
// Define nodes, elements and DoFs
    n=1:numx;
    element(:,1)=n;
    element(:,2)=element+1;  // define elements in sequential order
    numnode=size(node,2);
    numelem=size(element,1);
    ndof = numnode;
    xx=node;
// Stiffness matrix
K = sparse([],[],[numnode,numnode]);
for e = 1:numelem
  index  = element(e,:);  // vector  DoF for each element
  nn = size(index ,"*");  // vector index length
  length_element = node(index (2))-node(index (1));
  detJ0 = length_element/2;  // J=L/2
  invJ0 = 1/detJ0;  // J^-1
  N = ([1,1]/2)';  // shape function matrix
  dNdxi = [-1;1]/2;  // derived from the shape function matrix in relation
  // to the natural coordinates
  dNdx = dNdxi*invJ0;
// derived from the shape function matrix in relation to X (local coordinates)
  // matrix B *** [K] = sum of [B] ^ T. [D]. [B]. | J | , where [D] = E.A ***
  B = [-1/length_element  1/length_element]; // deformation vector
  K(index ,index) = K(index,index)+(B'*E*B*A*detJ0*2);  //B*2 = 2/L = dr/dx
end;
K=full(K)
```

```
//Load vector
f = zeros(numnode,1);
f(2)=3000;
// Boundary conditions
fixedNodeW =find(xx==min(node(:)) | xx==max(node(:)))'; dofs=[fixedNodeW];

//1st and last nodes restricted
U = zeros(numnode,1);
act = setdiff((1:ndof)',dofs);
U = K(act,act)\f(act);
U1 = zeros(ndof,1);
U1(act)=U;
U = U1
// stresses calculation
stress = zeros(numelem,size(element,2),1);
stressPoints(1,1) = -1;stressPoints(1,2) = 1;
for e = 1:numelem
  index  = element(e,:);  indexB = index ;  nn = size(index ,"*");
  for q = 1:nn
    pt = stressPoints(q);
    N = ([1-pt,1+pt]/2)';
    dNdxi = [-1;1]/2;
    dNdx = dNdxi*invJ0;  // same as lines of previous looping up to here
    // B matrix
     B = [-1/length_element  1/length_element]; // deformation vector
    strain = B*U(indexB);  stress(e,q) = E*strain;
  end;
end;
disp (stress(:,1))
Y = node'+U*1000;
plot(node(1),0,"ro");
plot(Y(2),0,"bo");plot(Y(3),0,"bo");
plot(node(2),0,"ro");plot(node(3),0,"ro");plot(node(4),0,"ro");
hl=legend(['original nodes','displaced nodes(x1000)']);
xgrid
```

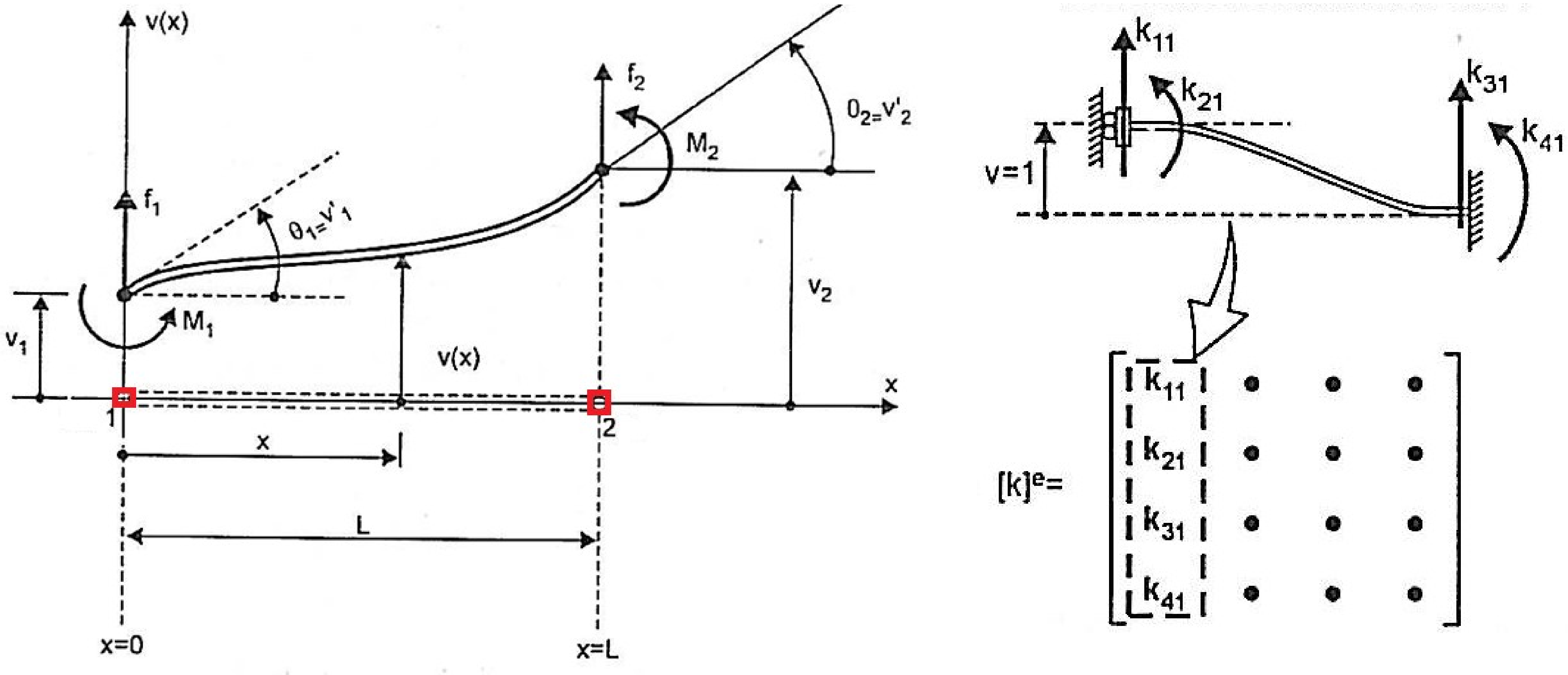
Stress results (N/m^2) and nodes displacements plot

2000. - 1000. - 1000.

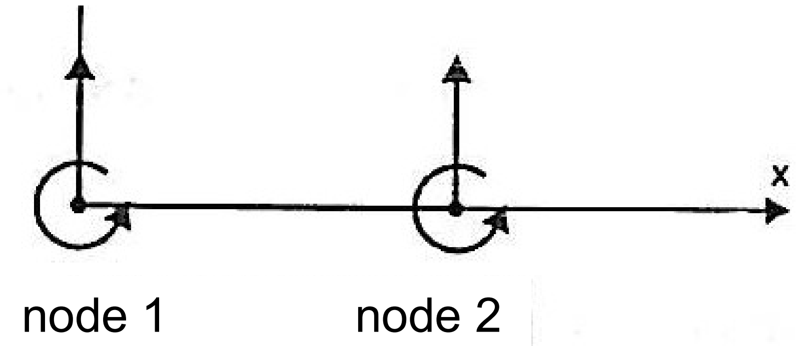


STRENGTH MATRIX OF A BAR ELEMENT CONSIDERING

BENDING AND ROTATION



INTERPOLATION FUNCTION



4 degrees of freedom

$$\boxed{v(x) = C_1 + C_2 . x + C_3 . x^2 + C_4 . x^3}$$

INTERNAL LOADS IN A BEAM

$$v(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \cdot \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix} \qquad v(x) = [H(x)] \cdot \{C\}$$

Matrix which represents displacement and rotation **(v')**:

$$\{\delta(x)\} = \begin{Bmatrix} v(x) \\ v'(x) \end{Bmatrix} = \begin{bmatrix} 1 & x & x^2 & x^3 \\ 0 & 1 & 2 \cdot x & 3 \cdot x^2 \end{bmatrix} \cdot \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

Substituting the value of *x* by 0 and L, we have:

$x = 0 \longrightarrow$

$$v(x) = v(0) = v_1 = C_1 + C_2 \cdot 0 + C_3 \cdot 0^2 + C_4 \cdot 0^2 = C_1$$

$$v'(x) = v'(0) = v'_1 = C_2 + 2 \cdot C_3 \cdot 0 + 3 \cdot C_4 \cdot 0^2 = C_2$$

$x = L \longrightarrow$

$$v(x) = v(L) = v_2 = C_1 + C_2 \cdot L + C_3 \cdot L^2 + C_4 \cdot L^3$$

$$v'(x) = v'(L) = v'_2 = C_2 + 2 \cdot C_3 \cdot L + 3 \cdot C_4 \cdot L^2$$

$$\{\delta\} = [A].\{C\}$$

$$\{\delta\} = \begin{Bmatrix} v_1 \\ v'_1 \\ v_2 \\ v'_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & L & L^2 & L^3 \\ 0 & 1 & 2 \cdot L & 3 \cdot L^2 \end{bmatrix} \cdot \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

Being:

$$v(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \cdot \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

and considering the inverse of A:

$$[A]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{L^2} & -\frac{2}{L} & \frac{3}{L^2} & -\frac{1}{L} \\ \frac{2}{L^3} & \frac{1}{L^2} & -\frac{2}{L^3} & \frac{1}{L^2} \end{bmatrix}$$

$$v(x) = [H(x)] \cdot \{C\}$$

$$\boxed{\{C\} = [A]^{-1} \cdot \{\delta\}} \longrightarrow \boxed{v(x) = [H(x)] \cdot [A]^{-1} \cdot \{\delta\}}$$

$$\boxed{[H(x)].[A]^{-1} = [N(x)]}$$

SHAPE FUNCTION

$$N(x)=\begin{bmatrix}1 & x & x^2 & x^3\end{bmatrix}_{1x4}\cdot\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ -\frac{3}{L^2} & -\frac{2}{L} & \frac{3}{L^2} & -\frac{1}{L}\\ \frac{2}{L^3} & \frac{1}{L^2} & -\frac{2}{L^3} & \frac{1}{L^2}\end{bmatrix}_{4x4}$$

$$N(x)=\left[\left(\underbrace{1+0\cdot x-3\cdot\frac{x^2}{L^2}\cdot x^2+2\cdot\frac{x^3}{L^3}}\right)\left(\underbrace{1\cdot 0+x\cdot 1-2\frac{x^2}{L}+\frac{x^3}{L^2}}\right)\left(\underbrace{1\cdot 0+x\cdot 0+3\frac{x^2}{L^2}-2\frac{x^3}{L^3}}\right)\left(\underbrace{1\cdot 0+x\cdot 0-\frac{x^2}{L}+\frac{x^3}{L^2}}\right)\right]$$

$$N(x)=\begin{bmatrix}\eta_{11} & \eta_{12} & \eta_{13} & \eta_{14}\end{bmatrix}_{1x4}$$

CURVATURE CALCULATION

$$\boxed{\frac{1}{\rho}=\frac{M}{E\cdot I}=\frac{d^2v}{dx^2}=v''(x)}$$

$$\boxed{v''(x)=2\cdot C_3+6\cdot C_4\cdot x}$$

$$v''(x)=\begin{bmatrix}0 & 0 & 2 & 6\cdot x\end{bmatrix}\begin{Bmatrix}C_1\\C_2\\C_3\\C_4\end{Bmatrix}$$

being: $\{C\}=[A]^{-1}\cdot\{\delta\}$, we have:

$$v''(x)=\begin{bmatrix}0 & 0 & 2 & 6\cdot x\end{bmatrix}\cdot[A]^{-1}\cdot\{\delta\}$$

being: $[B(x)]=\begin{bmatrix}0 & 0 & 2 & 6\cdot x\end{bmatrix}\cdot[A]^{-1}$

$$\boxed{v''(x)=[B(x)].\{\delta\}}$$

Performing the matrix multiplication, we have:

$$[B]=\left[\left(-\frac{6}{L^2}+12\cdot\frac{x}{L^3}\right)\ \left(-\frac{4}{L}+6\cdot\frac{x}{L^2}\right)\ \left(\frac{6}{L^2}-12\cdot\frac{x}{L^3}\right)\ \left(-\frac{2}{L}+6\cdot\frac{x}{L^2}\right)\right]$$

By means of $[B]^T.[B]$ and integrals, it is obtained:

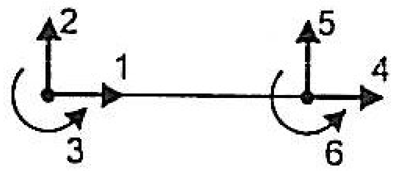
$$\{k\}^e=\begin{bmatrix}\frac{12.E.I}{L^3} & \frac{6.E.I}{L^2} & \frac{-12.E.I}{L^3} & \frac{6.E.I}{L^2}\\ \frac{6.E.I}{L^2} & \frac{4.E.I}{L} & \frac{-6.E.I}{L^2} & \frac{2.E.I}{L}\\ \frac{-12.E.I}{L^3} & \frac{-6.E.I}{L^2} & \frac{12.E.I}{L^3} & \frac{-6.E.I}{L^2}\\ \frac{6.E.I}{L^2} & \frac{2.E.I}{L} & \frac{-6.E.I}{L^2} & \frac{4.E.I}{L}\end{bmatrix}$$

The matrix [B] allows to change nodal displacements into deformations within the element.

Being $\{k\}^e$, the stiffness matrix, we have: $\{k\}^e = E.I.\int_0^L [B]^T.[B]dx$

Considering the stiffness matrix of the element for axial loads and bending moments simultaneously, we have:

$$\{k\}^e = \{k\}^e_{axial} + \{k\}^e_{bending}$$



$$a = \frac{E.A}{L} \quad e \quad b = \frac{EI}{L^3}$$

$$[k]^e = \begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & \\ \left[\begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\right. & a & 0 & 0 & -a & 0 & 0 & 1 \\ & 0 & 12b & 6bL & 0 & -12b & 6bL & 2 \\ & 0 & 6bL & 4bL^2 & 0 & -6bL & 2bL^2 & 3 \\ & -a & 0 & 0 & a & 0 & 0 & 4 \\ & 0 & -12b & -6bL & 0 & 12b & -6bL & 5 \\ & 0 & 6bL & 2bL^2 & 0 & -6bL & 4bL^2 & 6 \end{array}$$

$$f = \begin{Bmatrix} f_{x_1} \\ f_{y_1} \\ M_1 \\ f_{x_2} \\ f_{y_2} \\ M_2 \end{Bmatrix} \qquad \delta = \begin{Bmatrix} u_1 \\ v_1 \\ \theta_1 \\ u_2 \\ v_2 \\ \theta_2 \end{Bmatrix}$$

Example in Scilab

Develop a code to calculate displacements (u, v) and curvature of a 2-nodes single element. The first node is restricted and loads (fx = 3000 and fy = -500 N) and a moment of 50 Nm are applied to the second node.

Please consider:

$E = 30e6\ N/m^2$;

$A = 6.8\ m^2$;

$L = 90\ m$;

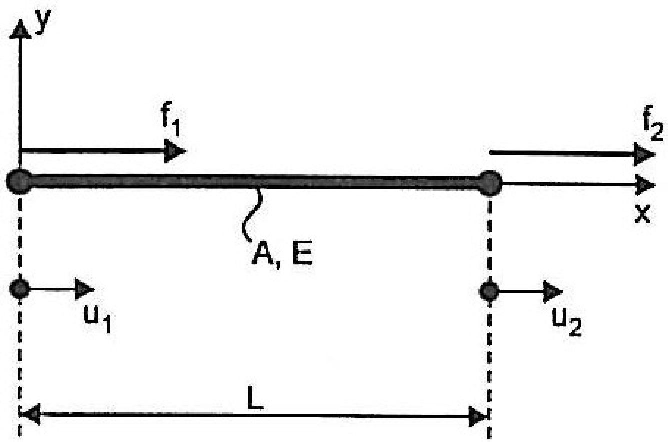
$I = 65\ m^4$.

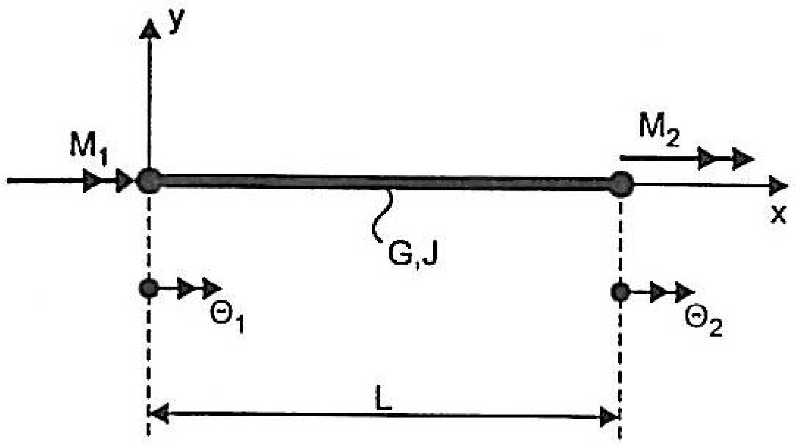
Computacional implementation proposal

```
clear;clc;
// Input data
E = 30000000;A = 6.8;I = 65;EA = E*A;EI = E*I;
node = [0,0;1,0];
xx = node(:,1);yy = node(:,2);
element = [1,2];
numnode = size(node,1);
numelem = size(element,1);
// Matrix initialization
U = zeros(3*numnode,1);
f = zeros(3*numnode,1);
K = sparse([],[],[3*numnode,3*numnode]);
// Load
f(4)=3000; f(5)=-500; f(6)=50;// fx; fy; Moment
for e = 1:numelem
  index = element(e,:);
  indexB = [index,index+numnode,index+2*numnode];
  xa = xx(index(2))-xx(index(1));
  ya = yy(index(2))-yy(index(1));
  L = sqrt(xa*xa+ya*ya);   //cosa=xa/L;    sina=ya/L;
  a = EA/L;  b = EI/(L^3);
  k1 = [a,0,0,-a,0,0;
        0,12*b,(6*b)*L,0,-12*b,(6*b)*L;
        0,(6*b)*L,(4*b)*(L^2),0,-(6*b)*L,(2*b)*(L^2);
        -a,0,0,a,0,0;
        0,-12*b,-(6*b)*L,0,12*b,-(6*b)*L;
        0,(6*b)*L,(2*b)*(L^2),0,-(6*b)*L,(4*b)*(L^2)];
  K(indexB,indexB) = K(indexB,indexB)+k1;
end;
// Contourn conditions
b = [1,1,1,0,0,0];
K([1,2,3],:) = [];// delete 1st, 2nd and 3rd lines of K which displacement = 0
K(:,[1,2,3]) = [];// delete 1st, 2nd and 3rd column of K which displacement = 0
f([1,2,3],:) = [];// delete 1st, 2nd and 3rd lines of f which displacement = 0
d1 = K\f;// // Displacements calculation
disp(d1)
```

```
 d1 =
   0.0000147
 - 7.265D-08
 - 0.0000001
```

STIFFNESS MATRIX CONSIDERING TORSION





$$\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \cdot \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

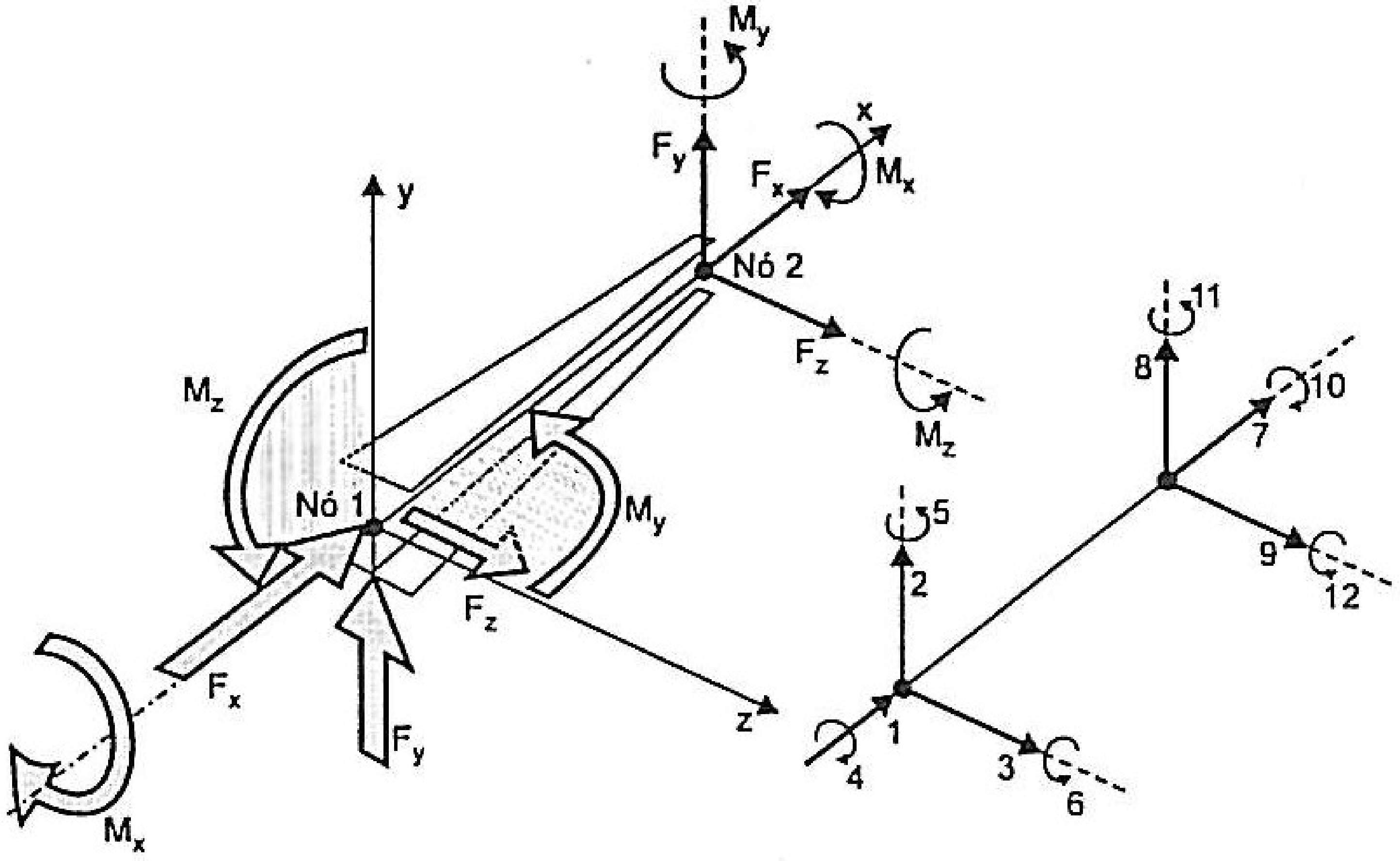
$$\begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix} = \begin{bmatrix} \frac{GJ}{L} & -\frac{GJ}{L} \\ -\frac{GJ}{L} & \frac{GJ}{L} \end{bmatrix} \cdot \begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix}$$

$$\frac{E}{2.(1+\nu)}$$

$$\longrightarrow \boxed{[k]^e = \begin{bmatrix} \frac{GJ}{L} & -\frac{GJ}{L} \\ -\frac{GJ}{L} & \frac{GJ}{L} \end{bmatrix}}$$

$$\boxed{M = \frac{G.J}{L}.\theta}$$

DoFs



GLOBAL STIFFNESS MATRIX

$$
[k]^e = \begin{bmatrix}
a \\
0 & 12b_z \\
0 & 0 & 12b_y \\
0 & 0 & 0 & t \\
0 & 0 & -6b_y \cdot L & 0 & 4b_y \cdot L^2 \\
0 & 6b_z \cdot L & 0 & 0 & 0 & 4b_z \cdot L^2 \\
-a & 0 & 0 & 0 & 0 & 0 & a \\
0 & -12b_z & 0 & 0 & 0 & -6b_z \cdot L & 0 & 12b_z \\
0 & 0 & -12b_y & 0 & -6b_y \cdot L & 0 & 0 & 0 & 12b_y \\
0 & 0 & 0 & -t & 0 & 0 & 0 & 0 & 0 & t \\
0 & 0 & -6b_y \cdot L & 0 & 2 \cdot b_y L^2 & 0 & 0 & 0 & 6b_y \cdot L & 0 & 4b_y \cdot L^2 \\
0 & 6b_z \cdot L & 0 & 0 & 0 & 2b_z \cdot L^2 & 0 & -6b_z \cdot L & 0 & 0 & 0 & 4b_z \cdot L^2
\end{bmatrix}
$$

$$a = \frac{EA}{L}$$

$$b_z = \frac{EI_z}{L^3}$$

$$b_y = \frac{E \cdot I_y}{L^3}$$

$$t = \frac{GJ}{L}$$

Exercise on Scilab

Based on the previous code data, calculate displacements and torsion being : E=30e6; A=6.8; Iz=65; Iy=45; G=80e6; J=50; f(7)=3000.0;f(8)=500;f(9)=300;f(10)=500;f(11)=300;f(12)=400.

```
k=[ a     0    0        0    0        0       -a     0     0     0    0       0 ;
    0  12*bz   0        0    0     6*bz*ll   0  -12*bz   0     0    0       6*bz*ll;
    0     0  12*by      0 -6*by*ll   0        0     0    -12*by 0  -6*by*ll  0 ;
    0     0    0        t    0        0        0     0     0    -t    0       0 ;
    0     0  -6*by*ll   0  4*by*ll^2 0        0     0 -6*by*ll  0  2*by*ll^2 0 ;
    0   6*by*ll    0        0    0   4*bz*ll^2    0 -6*bz*ll  0     0    0  2*bz*ll^2 ;
   -a     0    0        0    0        0        a     0     0     0    0       0 ;
    0  -12*bz  0        0    0     -6*bz*ll   0   12*bz   0     0    0      -6*bz*ll;
    0     0 -12*by      0 -6*by*ll   0        0     0    -12*by 0   6*by*ll  0 ;
    0     0    0       -t    0        0        0     0     0     t    0       0 ;
    0     0 -6*by*ll    0  2*by*ll^2 0        0     0  6*by*ll  0  4*by*ll^2 0 ;
    0 6*bz*ll  0        0    0   2*bz*ll^2    0 -6*bz*ll  0     0    0  4*bz*ll^2   ];
```