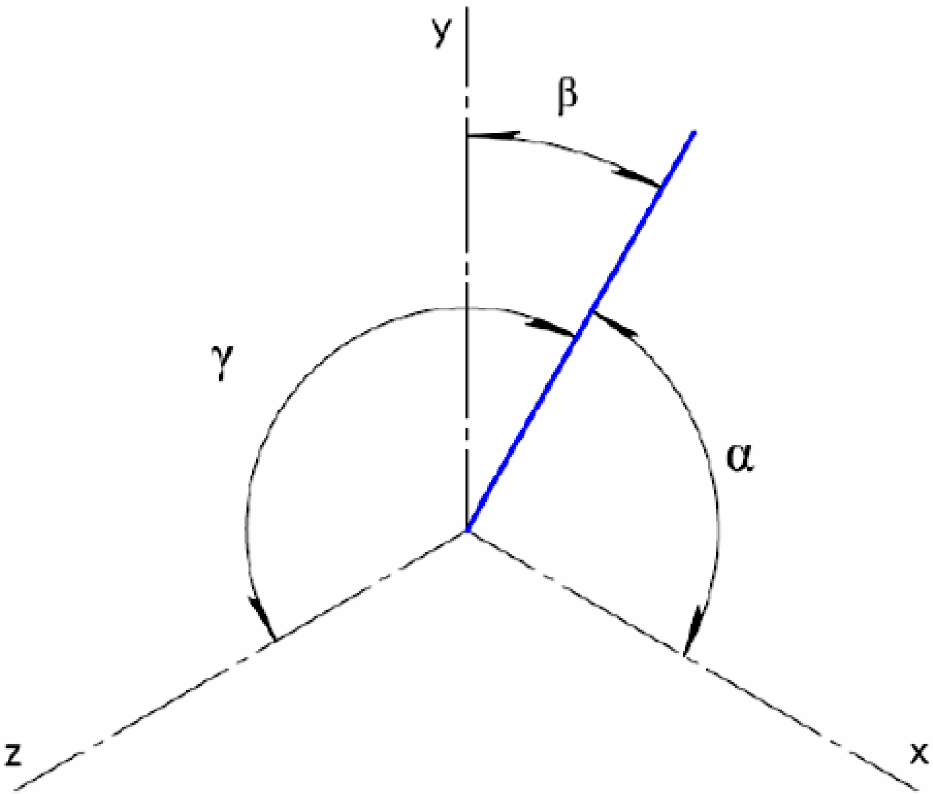
Bar element - Space truss

Formulation



$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$u_1 = d_1 \cos\alpha = d_1 l_z$$

$$v_1 = d_1 \cos\beta = d_1 m_z$$

$$q_1 = d_1 \cos\gamma = d_1 n_z$$

$$u_2 = d_2 \cos\alpha = d_2 l_z$$

$$v_2 = d_2 \cos\beta = d_2 m_z$$

$$q_2 = d_2 \cos\gamma = d_2 n_z$$

$$l_z = \cos\alpha = \frac{x_2 - x_1}{L}$$

$$m_z = \cos\beta = \frac{y_2 - y_1}{L}$$

$$n_z = \cos\gamma = \frac{z_2 - z_1}{L}$$

Transformation matrix

$$\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}' \cdot \begin{bmatrix} l_s & m_s & n_s & 0 & 0 & 0 \\ 0 & 0 & 0 & l_s & m_s & n_s \end{bmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ q_1 \\ u_2 \\ v_2 \\ q_2 \end{pmatrix}$$

$$\Downarrow$$

$$T = \begin{bmatrix} l_s & m_s & n_s & 0 & 0 & 0 \\ 0 & 0 & 0 & l_s & m_s & n_s \end{bmatrix}$$

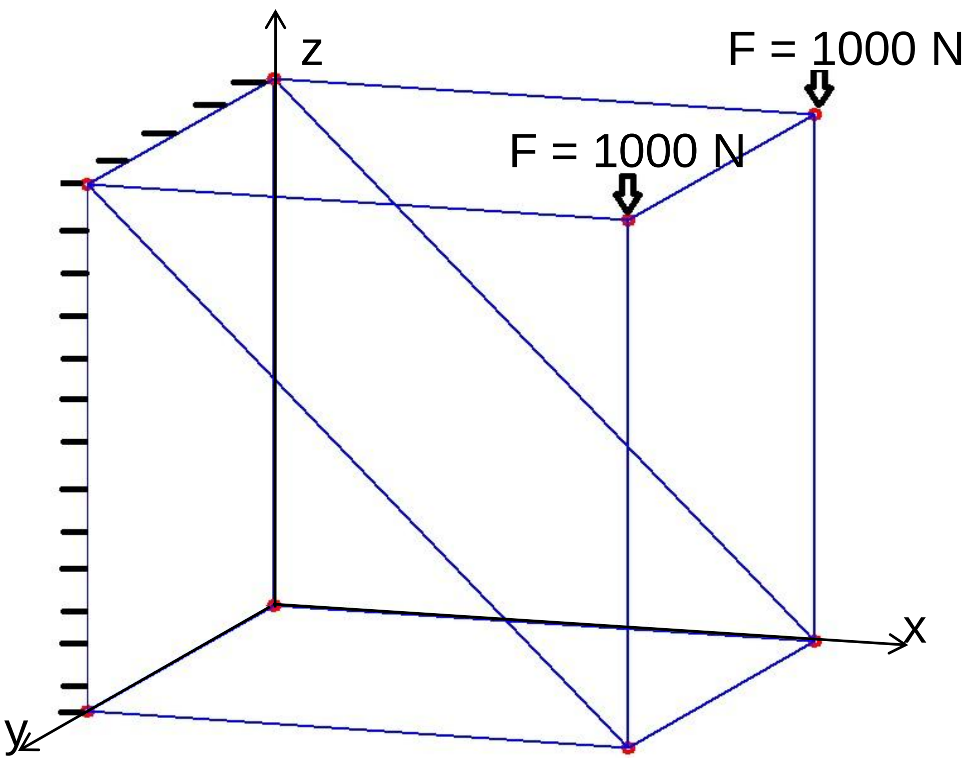
Stiffness matrix

$$[T]^T[K_e][T]\{u\} = [T]^T\{P_e\} \quad \rightarrow \quad [K_e]\{u\} = \{P_e\}$$

$$[K_e] = [T]^T[K_e][T] = \begin{bmatrix} \\ [T]^T \\ \\ \end{bmatrix} \frac{EA}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} l_s & m_s & n_s & 0 & 0 & 0 \\ 0 & 0 & 0 & l_s & m_s & n_s \end{bmatrix}$$

$$\frac{EA}{L}\begin{bmatrix} l_s^2 & l_s m_s & l_s n_s & -l_s^2 & -l_s m_s & -l_s n_s \\ l_s m_s & m_s^2 & m_s n_s & -l_s m_s & -m_s^2 & -m_s n_s \\ l_s n_s & m_s n_s & n_s^2 & -l_s n_s & -m_s n_s & -n_s^2 \\ -l_s^2 & -l_s m_s & -l_s n_s & l_s^2 & l_s m_s & l_s n_s \\ -l_s m_s & -m_s^2 & -m_s n_s & l_s m_s & m_s^2 & m_s n_s \\ -l_s n_s & -m_s n_s & -n_s^2 & l_s n_s & m_s n_s & n_s^2 \end{bmatrix}\begin{pmatrix} u_1 \\ v_1 \\ q_1 \\ u_2 \\ v_2 \\ q_2 \end{pmatrix} = \begin{pmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ F_{x2} \\ F_{y2} \\ F_{z2} \end{pmatrix}$$

Computacional implementation

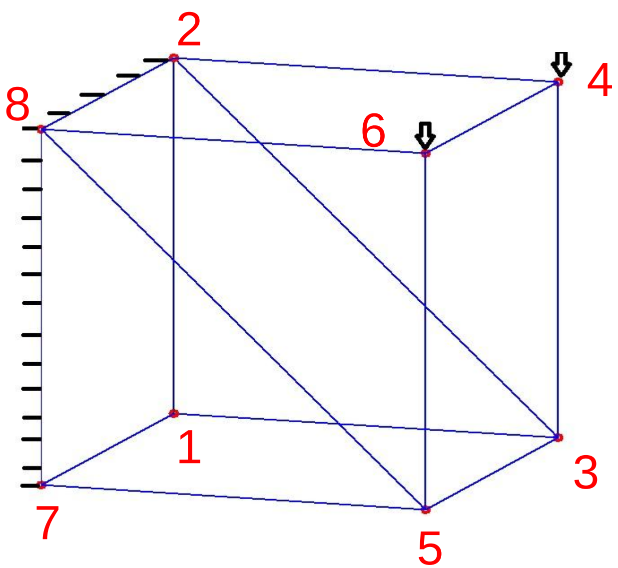


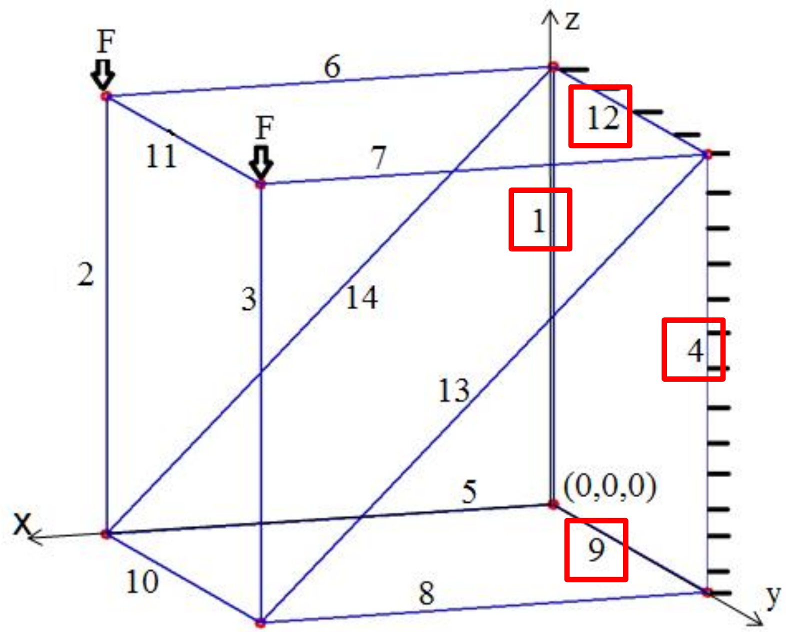
First of all you need to download and integrate to Scilab “FEMTruss. A Truss finite element code for scilab”: http://atoms.scilab.org/toolboxes/femtruss

or develop your own code to plot the mesh.

Elements and nodes connection

Restricted





DoFs:

1 –> 1,2,3
2 –> 4,5,6
3 –> 7,8,9
4 –> 10,11,12
5 –> 13,14,15
6 –> 16,17,18
7 –> 19,20,21
8 –> 22,23,24

Nodes connection

1 –> 1-2
2 –> 3-4
3 –> 5-6
4 –> 7-8
5 –> 1-3
6 –> 2-4
7 –> 8-6
8 –> 7-5
9 –> 1-7
10 –> 3-5
11 –> 4-6
12 –> 2-8
13 –> 8-5
14 –> 2-3

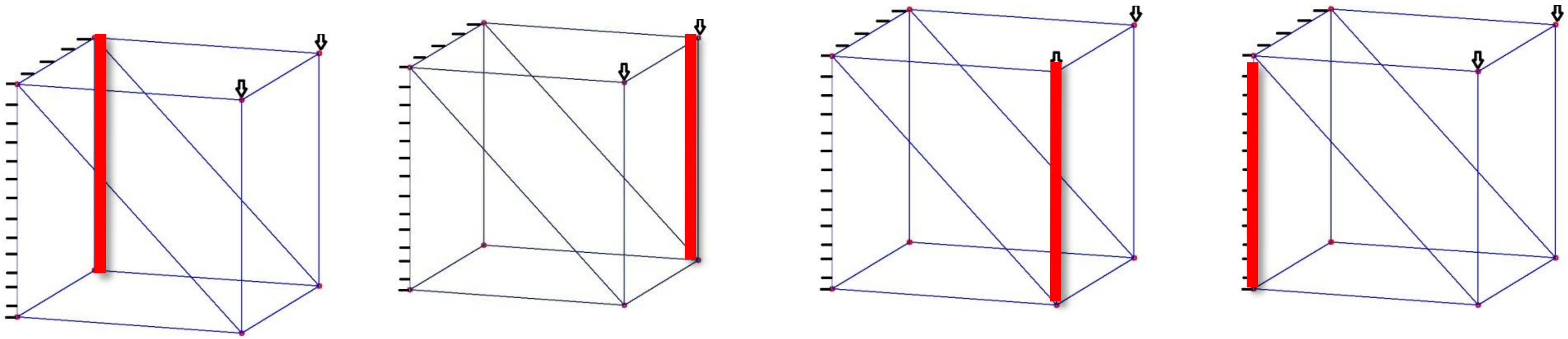
Scilab code

```
clear; clc;
E =200000;d = 25;A = ((d^2)*%pi)/4;EA = E*A; F = -1000; // Material: steel (MPA)
cd = 1000; // cube dimensions
node = [0,0,0; 0,0,cd; cd,0,0; cd,0,cd; cd,cd,0; cd,cd,cd; 0,cd,0; 0,cd,cd];//n
xx = node(:,1);yy = node(:,2);zz = node(:,3);// xx = 1st col;  yy = 2nd col; zz =
element = [1,2;3,4;5,6;7,8;1,3;2,4;8,6;7,5;1,7;3,5;4,6;2,8;8,5;2,3]//;1,4;7,6];/
numnode = size(node,1);//  total number of nodes
numelem = size(element,1);// total number of  elements
// Matrices initialization
U = zeros(3*numnode,1);// Displacement matrix with all elements equal to zero
K = zeros(3*numnode,3*numnode);// Stiffness matrix with all elements equal to ze
// Displacements
for e = 1:numelem
  indice = element(e,:);  // Index 1 represents the 1st element, index 2 represe
  indiceB = [3*indice(1)-2,3*indice(1)-1,3*indice(1),3*indice(2)-2,3*indice(2)-1,
  // lists the DoF of each element (3 DoF per node)
  xa = xx(indice(2))-xx(indice(1));  // X difference in the coordinate of each e
  ya = yy(indice(2))-yy(indice(1));  // Y difference in the coordinate of each e
  za = zz(indice(2))-zz(indice(1));  // Z difference in the coordinate of each e
  length_element = sqrt(xa*xa+ya*ya+za*za);  // element length
  l = xa/length_element;  // alpha slope
  m = ya/length_element;  // beta slope
  n = za/length_element;  // gamma slope
```

Stiffeness matrix

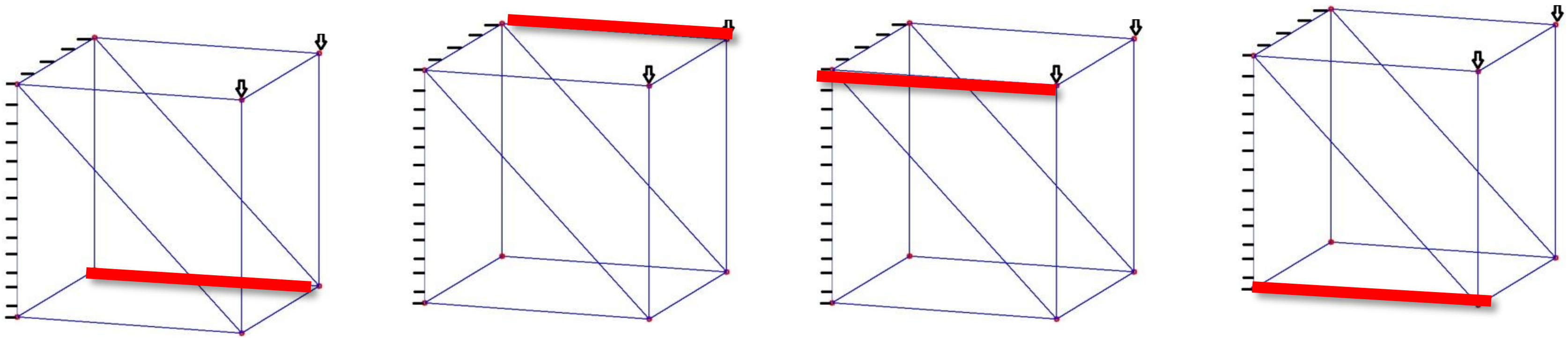
```
  k1 = (EA/length_element)*[l*l,l*m,l*n,-l*l,-l*m,-l*n;
                            l*m,m*m,m*n,-l*m,-m*m,-m*n;
                            l*n,m*n,n*n,-l*n,-m*n,-n*n;
                            -l*l,-l*m,-l*n,l*l,l*m,l*n;
                            -l*m,-m*m,-m*n,l*m,m*m,m*n;
                            -l*n,-m*n,-n*n,l*n,m*n,n*n];
  K(indiceB,indiceB) = K(indiceB,indiceB)+k1;  // Global stiffness matrix
end;
P = [0,0,0,0,0,0,0,0,0,0,0,F,0,0,0,0,0,F,0,0,0,0,0,0]';// F acts on the 3rd GL of nodes 4 and 6
f = P(7:18);
K2 = K(7:18,7:18);// Six first DoFs (nodes 1 and 2) and last six DoFs(nodes 7 and 8) clamped
x = K2\f;
xd = [0,0,0,0,0,0,(x(1:12))',0,0,0,0,0,0]'; //  Returns the restricted DoFs = 0.
// Stress (same looping as displacements, except last lines)
for e = 1:numelem
  indice = element(e,:);
  indiceB = [3*indice(1)-2,3*indice(1)-1,3*indice(1),3*indice(2)-2,3*indice(2)-1,3*indice(2)];
  xa = xx(indice(2))-xx(indice(1));
  ya = yy(indice(2))-yy(indice(1));
  za = zz(indice(2))-zz(indice(1));
  length_element = sqrt(xa*xa+ya*ya+za*za);
  c = xa/length_element;
  s = ya/length_element;
  r = za/length_element;
  T = [c,s,r,0,0,0;0,0,0,c,s,r];  // Transformation matrix
  d1 = T*xd(indiceB);  // displacement of each node
  deltaL = d1(2,:)-d1(1,:);
  Eps = deltaL*(1/length_element);  // Strain
  Sigma(1,e) = E*Eps;  // Stress
end;
```

Stiffness matrix of elements 1 through 4



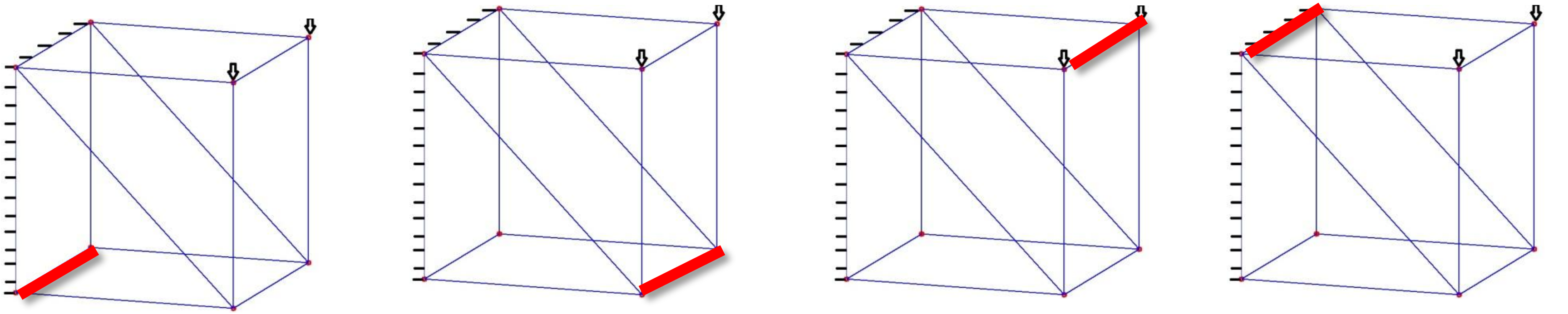
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	98174.77	0.	0.	- 98174.77
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	- 98174.77	0.	0.	98174.77

Stiffness matrix of elements 5 through 8



98174.77	0.	0.	- 98174.77	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
- 98174.77	0.	0.	98174.77	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.

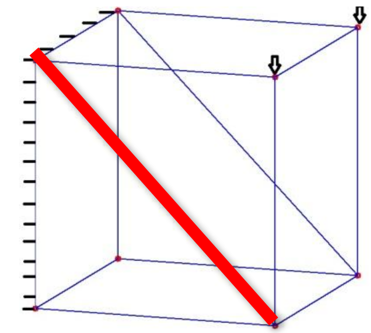
Stiffness matrix of elements 9 through 12

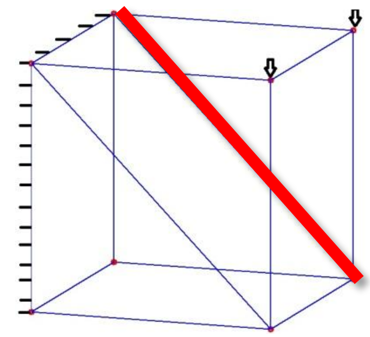


$$
\begin{matrix}
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 98174.77 & 0. & 0. & -98174.77 & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. \\
0. & -98174.77 & 0. & 0. & 98174.77 & 0. \\
0. & 0. & 0. & 0. & 0. & 0.
\end{matrix}
$$

Stiffness matrix of elements 13 through 14

34710.023	0.	- 34710.023	- 34710.023	0.	34710.023
0.	0.	0.	0.	0.	0.
- 34710.023	0.	34710.023	34710.023	0.	- 34710.023
- 34710.023	0.	34710.023	34710.023	0.	- 34710.023
0.	0.	0.	0.	0.	0.
34710.023	0.	- 34710.023	- 34710.023	0.	34710.023

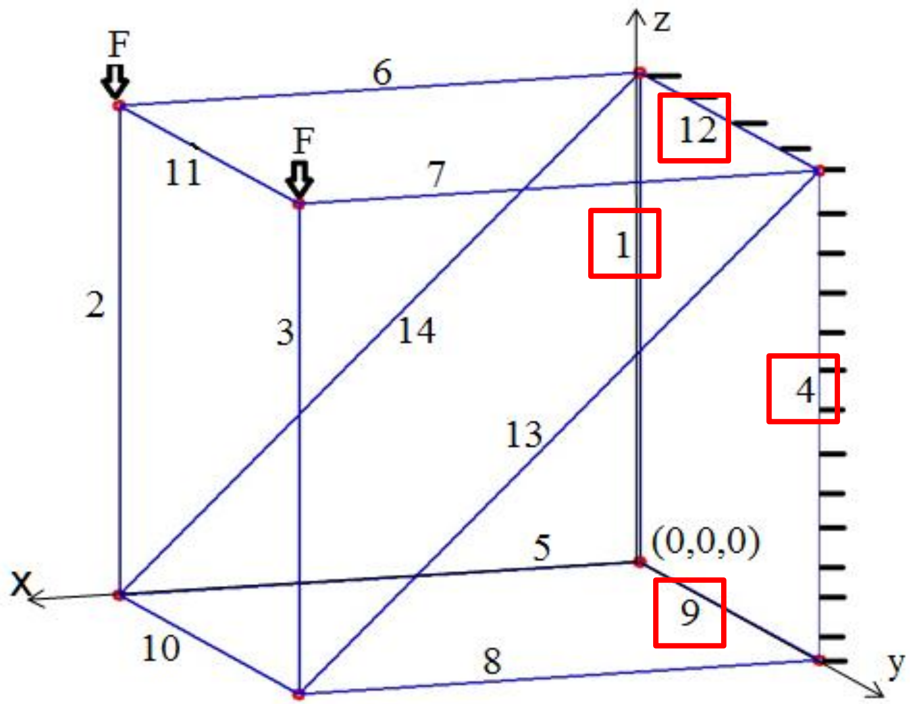




Commands to plot results

```
disp(Sigma)
dn = [(xd(1:3))';(xd(4:6))';(xd(7:9))';(xd(10:12))';(xd(13:15))';(xd(16:18))';(xd(19:21))';(xd(22:24))'];
drawlater;
t = element; p = node; u = dn;
plotmesh(t,p,0,0,'green');
s = 1000;
pd = p + s * dn;
plotmesh(t,pd,0,0,'red');
legends(['before loading','after loading'], [color('green') color('red')], 'ur');
drawnow;
```

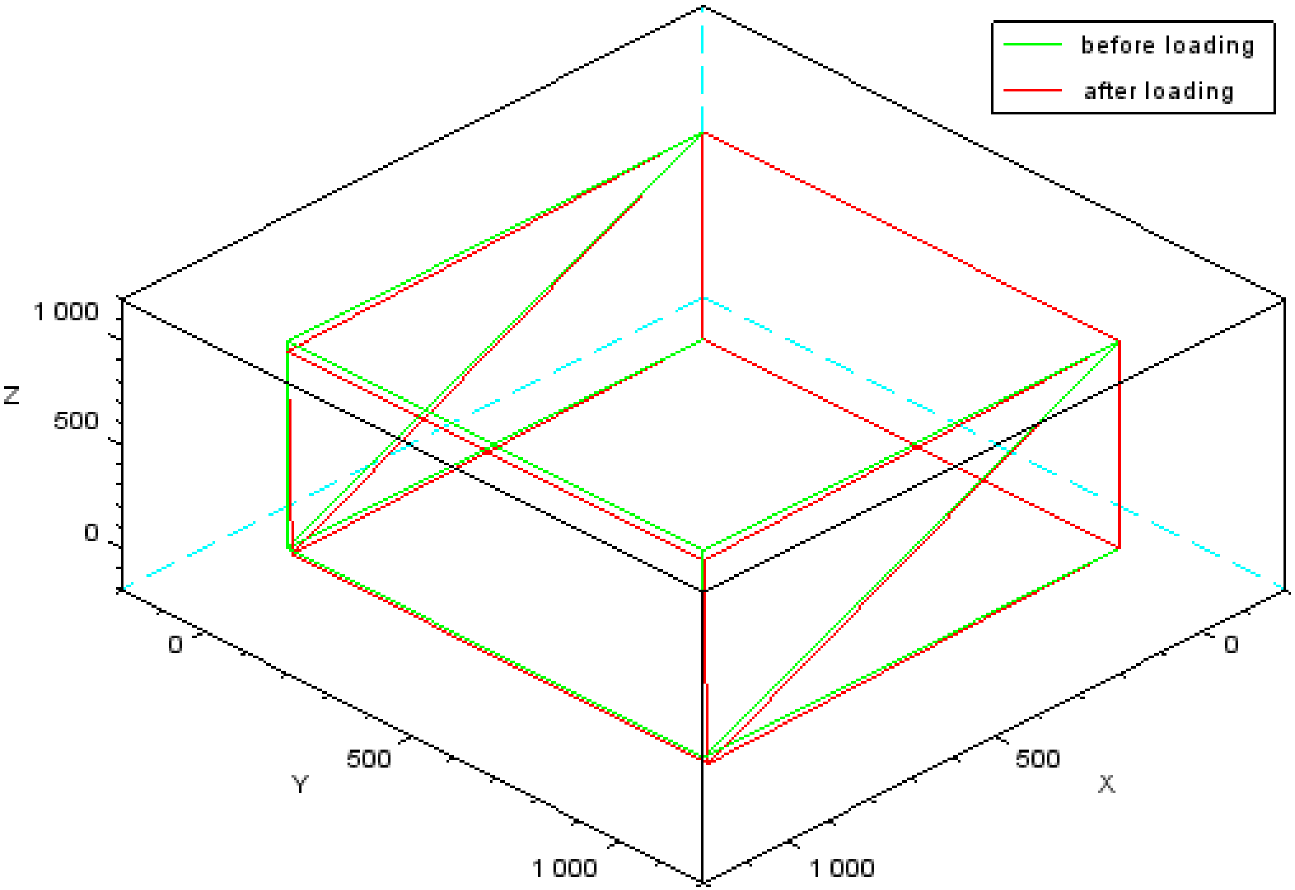
Stress results (MPa)



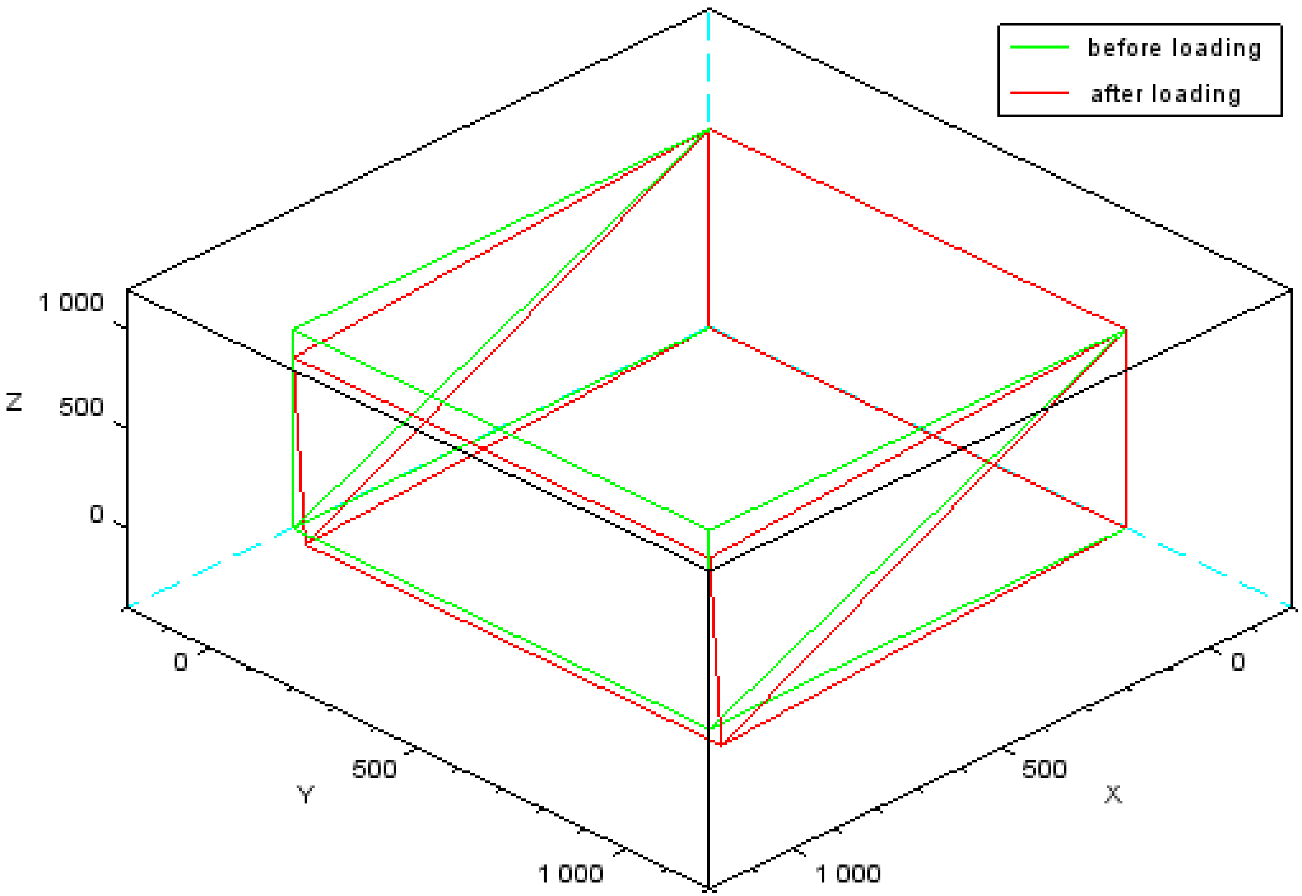
Element	1	2	3	4	5	6	7
stress	0.0	-2.037	-2.037	0.0	-2.037	- 2.24e-16	0.0
Element	**8**	**9**	**10**	**11**	**12**	**13**	**14**
stress	-2.037	0.0	3.930e-16	0.0	0.0	2.88	2.88

Displacements (x1000, mm): steel

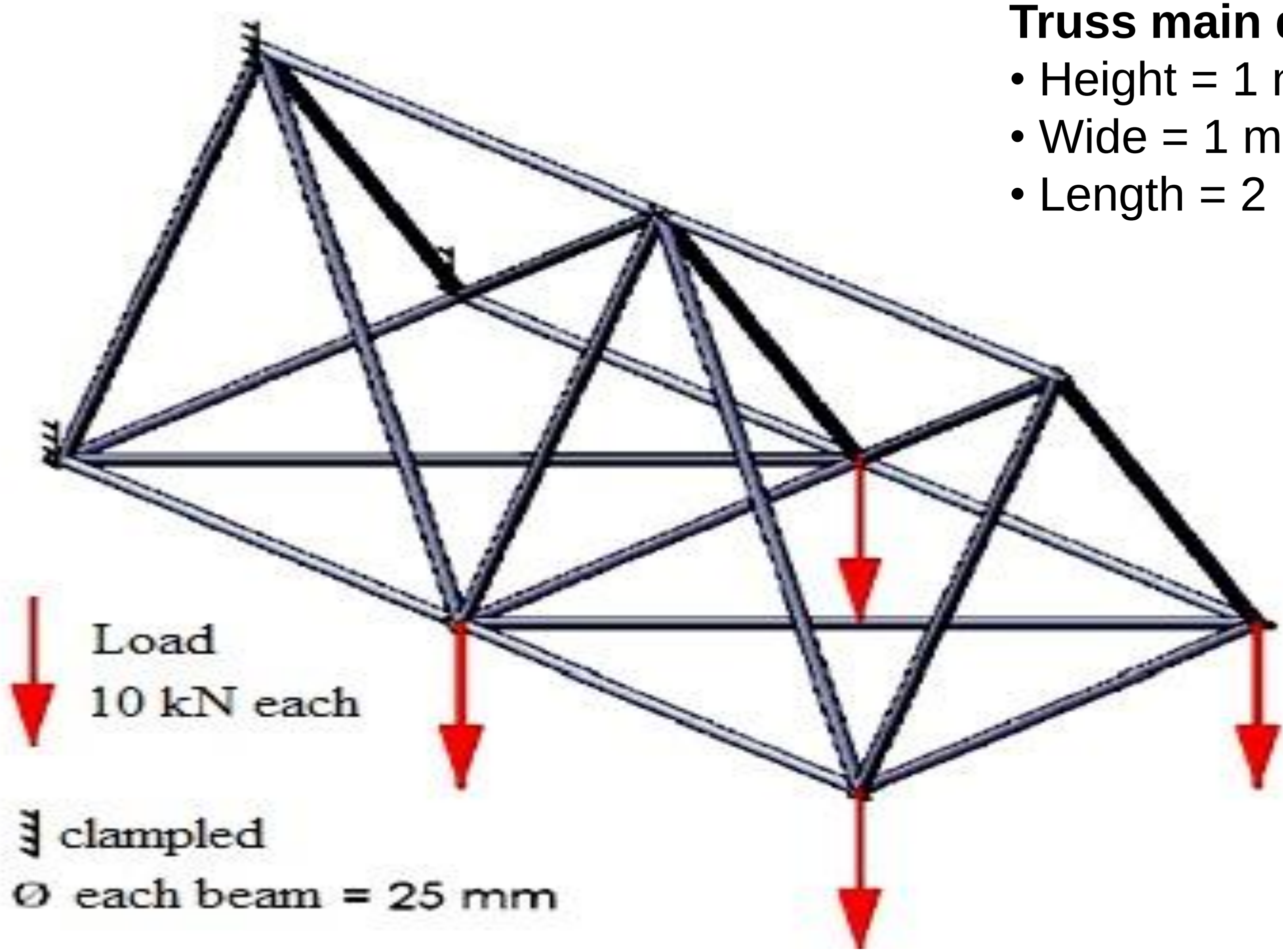
E = 210 GPa



Displacements (x1000, mm): aluminium
E = 70 GPa



Exercise: Write a Scilab code to calculate the stresses and plot displacements, considering E = 200 GPa.



Truss main dimensions:
- Height = 1 m
- Wide = 1 m
- Length = 2 m