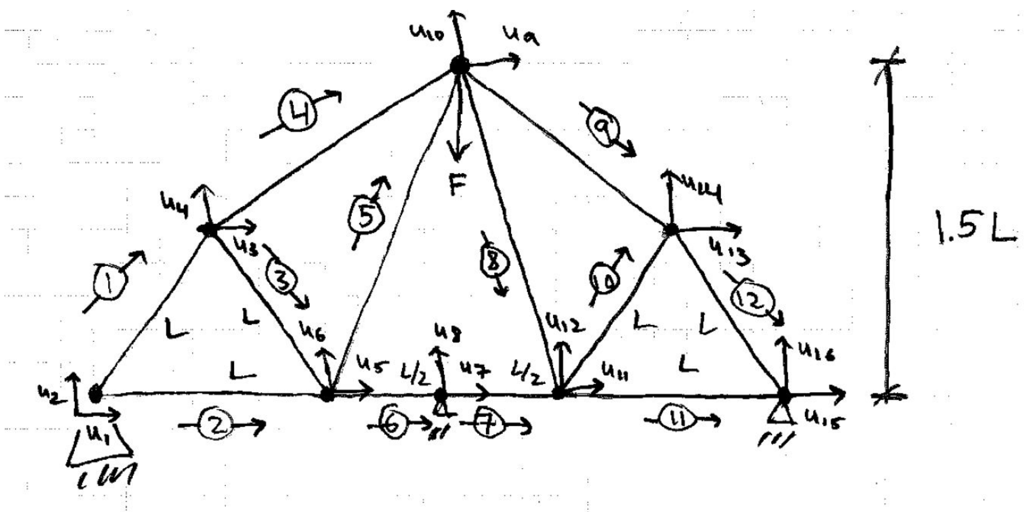
# PROJECT 1

GROUP 2 - JIMMIE ANDERSSON & GUSTAV GOOD

## Task 1

### a)




```
%Topology matrix
Edof = [1  1  2  3  4
        2  1  2  5  6
        3  3  4  5  6
        4  3  4  9  10
        5  5  6  9  10
        6  5  6  7  8
        7  7  8  11  12
        8  9  10  11  12
        9  9  10  13  14
        10  11  12  13  14
        11  11  12  15  16
        12  13  14  15  16];
```

```
%Boundary conditions
bc = [ 1 0
       2 0
       8 0
       15 0
       16 0
       10 -a0];
```

```
External loading f(10) = -F
```

## b)

After stating the edof, bc:s and input data in task a), we create a dof and coord matrix(each row of the dof correspond to the coordinates at the same row in coord matrix) Origo is placed in the bottom left corner. This is done to get the start and end coord (x&y value) for each element(Ex & Ey). We get the coords from the calfem function **`coordxtr`** where we input the edof,coord and dof matrices. Output from **`coordxtr`** are Ex & Ey. Then we state the material properties (Ep). These are same for all element so a 1x2 matrix is just needed. We then do a for loop over all the elements, where a local stiffness matrix (Ke) is created for each element (done with **`bar2e`** with the inputs: Ex,Ey and Ep). Each Ke is added in the right place of the global stiffness matrix (K) with help of the edof. By using **`solveq`** (with inputs: stiffness matrix, forces and boundary conditions) we get the reaction forces and deformations. By using **`extract`** with inputs: Edof and deformations we can get the element deformations in x&y at start & end. It is used for plotting the deformations with **`eldisp2`**. With **`eldraw2`** we plot the original state and we can compare the figures.

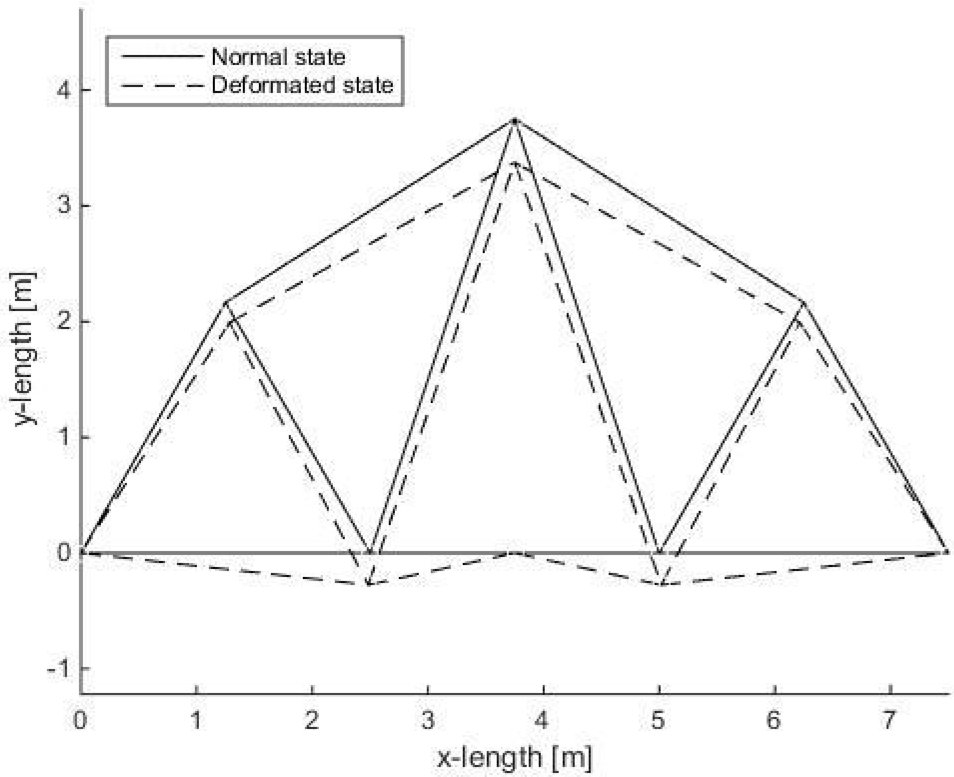
## c)

```
L =2.5; %Length[m]
f= zeros(16,1); %Force matrix
K = zeros(16); % Stiffness matrix
d= 0.1; %Diameter[m]
E = 210*10^9; %Youngs modulus[N/m^2]
A = pi*(d/2)^2; %Area of bar [m]
a0 = 0.006; %Displacement of node (for task c)
```

```
Dof = [1 2
        3 4
        5 6
        7 8
        9 10
        11 12
        13 14
        15 16];

%Origo at node1, node2
Coord =       [0 0
               L/2 sqrt(3)*L/2
               L 0
               3*L/2 0
               3*L/2 1.5*L
               2*L 0
               2.5*L sqrt(3)*L/2
               3*L 0];

%Ex &Ey is the coords of all the elements
% 2 is number of element dof's

[Ex,Ey] = coordxtr(Edof,Coord,Dof,2);


%Material properties for each element
Ep = [E A];

  for i=1:12
       Ke = bar2e(Ex(i,:),Ey(i,:),Ep(1,:));
       K(Edof(i,2:5),Edof(i,2:5)) = K(Edof(i,2:5),Edof(i,2:5))+ Ke;
  end

%Getting the reaction forces and deformations
[a,r] = solveq(K,f,bc);
Ed=extract(Edof,a);


%The wanted force
F = r(10);
```

**F = -2.4400e+06 N**

```
%Drawing the original state and the deformation
eldisp2(Ex,Ey,Ed,[2 6 1]);
eldraw2(Ex,Ey,[1 6 1]);
```




The required force is **2.44 MN**.

The non-zero deformation is accounted for by adding it in the **a**-matrix in the **Ka**=**f** solution, along with the locked dof's which will be set to zero. The rest are unknowns.

## d)

```
%The rest of the forces are in the r matrix (output from solveq). Each row represents the force at the degree of freedom with that row-index.
876 kN
1220 kN
0
0
0
0
0
```

```
0
0
-2440 kN
0
0
0
0
-876 kN
1220 kN

%If the sum of all the forces in x direction and y %direction is zero and the total
%moment is zero it is equilibrium.

  rx =0;
  ry=0;
  Mtot=0

  for i=1:length(r)/2
       r1 = r(i*2-1);
       r2 = r(i*2);
       rx = rx + r1;
       ry = ry + r2;
       Mtot= Mtot+ r1*Coord(i,2); %Rx * length in y-dir
       Mtot= Mtot+ r2*Coord(i,1);
 end

  %The sum is very close to zero (not zero of because of rounding
  %errors) so we have equilibrium
  Mtot = 1,86264514923096e-09     [N*m]
  rx =0                           [N]
  ry = -4.6566e-10                [N]
```

Task 2 (1)

a) $$\frac{d}{dx}\left(k(x)A(x)\frac{dT(x)}{dx}\right)+Q(x)=0 \quad (1)$$

$$q_n(0)=\alpha_{in}\left(T(0)-T_{in}\right)$$

$$q_n(500)=\alpha_{out}\left(T(500)-T_{out}\right)$$

b) We multiply eq 1 with the weight function $v(x)$ to ~~obtain~~ obtain

$$v\left[\frac{d}{dx}\left(Ak\frac{dT}{dx}\right)+Q\right]=0 \quad (2)$$

We may integrate this over the related region $0\leq x\leq L$

$$\int_0^L v\left[\frac{d}{dx}\left(Ak\frac{dT}{dx}\right)+Q\right]dx=0 \quad (3)$$

We may use the chain rule and integration by parts to obtain:

$$\int_0^L v\frac{d}{dx}\left(Ak\frac{dT}{dx}\right)dx=\left[vAk\frac{dT}{dx}\right]_0^L-\int_0^L\frac{dv}{dx}Ak\frac{dT}{dx}dx \quad (4)$$

By combining (3) and (4) we get:

$$\int_0^L\left(\frac{dv}{dx}Ak\frac{dT}{dx}\right)dx=\left[vAk\frac{dT}{dx}\right]_0^L+\int_0^L vQ\,dx \quad (5)$$

② 

$$\left[ vAk\frac{dT}{dx} \right]_0^L = \left( vAk\frac{dT}{dx} \right)_{x=L} - \left( vAk\frac{dT}{dx} \right)_{x=0} \quad (6)$$

Using $q = -k\frac{dT}{dx}$ and (6) in (5):

$$\int_0^L \left( \frac{dv}{dx} Ak \frac{dT}{dx} \right) dx = -vAq(L) + vAq(0) + \int_0^L vQ\,dx$$

Using $q_n(0) = -q(0)$ and the boundary conditions we can conclude that:

$$\boxed{\int_0^L \left( \frac{dv}{dx} Ak \frac{dT}{dx} \right) dx = -vA\alpha_{out} (T(0.5) - T_{out}) - vA\alpha_{in} ( T(0) - T_{in} ) + \underbrace{\int_0^L vQ\,dx}_{=0 \text{ in our case}}}$$

③

c) $\mathbb{K}\mathbf{a} = \mathbf{f} = \mathbf{f}_b + \mathbf{f}_\ell \qquad T = \bar{\mathbb{N}}\alpha$ where $\bar{\mathbb{N}} = [1 \quad x]$

we use linear approximation:

$$T = \alpha_1 + \alpha_2 x$$

$$T_i = \alpha_1 + \alpha_2 x_i \qquad \Rightarrow \qquad \begin{bmatrix} T_i \\ T_j \end{bmatrix} = \begin{bmatrix} 1 & x_i \\ 1 & x_j \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

$$T_j = \alpha_1 + \alpha_2 x_j$$

This can be written as

$$\mathbf{a}^e = \begin{bmatrix} T_i \\ T_j \end{bmatrix} \qquad \mathbb{C} = \begin{bmatrix} 1 & x_i \\ 1 & x_j \end{bmatrix} \qquad \mathbf{a}^e = \mathbb{C}\alpha$$

$$\Rightarrow \quad \alpha = \mathbb{C}^{-1}\mathbf{a}^e \quad \Rightarrow \quad T = \bar{\mathbb{N}}\mathbb{C}^{-1}\mathbf{a}^e$$

This can be written as $T = \mathbb{N}^e \mathbf{a}^e$ where

$$\mathbb{N}^e = \bar{\mathbb{N}}\mathbb{C}^{-1} = \begin{bmatrix} N_i^e & N_j^e \end{bmatrix}$$

$$\mathbb{C}^{-1} = \frac{1}{L}\begin{bmatrix} x_j & -x_i \\ -1 & 1 \end{bmatrix} \qquad \text{where} \quad L = x_j - x_i$$

$$\Rightarrow \quad N_i^e = \frac{-1}{L}(x - x_j)$$

$$N_j^e = \frac{1}{L}(x - x_i).$$

We define $\mathbb{B}^e = \dfrac{d\mathbb{N}^e}{dx} \Rightarrow \dfrac{dT}{dx} = \mathbb{B}^e \mathbf{a}^e$

(4)

$$\mathbb{B}^e = \left[ \frac{dN_i^e}{dx} \quad \frac{dN_j^e}{dx} \right] = \left[ -\frac{1}{L} \quad \frac{1}{L} \right]$$

$T(x) = \mathbb{N} a$ and $\mathbb{B}^e = \frac{d\mathbb{N}^e}{dx}$ gives us

$$\frac{dT(x)}{dx} = \mathbb{B}\, a \quad (7)$$

insert (7) in weak form:

$$(**) \int_0^L \frac{dv}{dx} Ak\, \mathbb{B} a\, dx = -vA\alpha_{in}(T(0) - T_{in}) - vA\alpha_{out}(T(0.5) - T_{out})$$

we let $v = \mathbb{N}\mathbb{C} \Rightarrow \frac{dv}{dx} = \mathbb{B}\mathbb{C}$

insert into $(**)$:

$$\int_0^L \mathbb{B}\mathbb{C} Ak\, \mathbb{B}\, dx\, a = -\mathbb{N}\mathbb{C}\alpha_{in}(T(0) - T_{in}) - \mathbb{N}\mathbb{C}\alpha_{out}(T(0.5) - T_{out})$$

$\mathbb{C}$ is constant, thus:

$$\mathbb{C}^T \int_0^L \mathbb{B}^T A\, k\, \mathbb{B} dx\, a = \mathbb{C}^T \Big( -\mathbb{N}(0)^T A(0)\alpha_{in}(T(0) - T_{in}) - \mathbb{N}(0.5)^T A(0)\alpha_{out}(T(0.5) - T_{in}) \Big)$$

$$\Rightarrow \underbrace{\int_0^L \mathbb{B}^T Ak\, \mathbb{B}\, dx}_{\mathbb{K}} a = \underbrace{-\mathbb{N}(0)^T A(0)\alpha_{in}(T(0) - T_{in}) - \mathbb{N}(0.5)^T A(0)\alpha_{out}(T(0.5) - T_{out})}_{\mathbb{f}_b}$$

(5)

using ~~[illegible]~~ $\mathbb{N}a = T(x)$:

$$\int_0^L \mathbb{B}^T A k \mathbb{B} dx\, a = -\alpha_{in} N^T(0) A(0) N(0) a - \alpha_{out} N^T(0.5) A(0.5) N(0.5) a$$

$$+ \alpha_{in} N^T(0) A(0) \cdot T_{in} + \alpha_{out} N^T(0.5) A(0.5) T_{out}$$

$$= -\mathbb{K}_{c1} a - \mathbb{K}_{c2} a + f_b^c = \mathbb{K} a$$

$$\Rightarrow \left(\mathbb{K} + \mathbb{K}_{c1} + \mathbb{K}_{c2}\right) a = f_b^c$$

$$\mathbb{K}_{c1} = \alpha_{in} A(0) \mathbb{N}^T(0) \mathbb{N}(0) = \alpha_{in} A(0) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_{in} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbb{K}_{c2} = \alpha_{out} A(0.5) \mathbb{N}^T(0.5) \mathbb{N}(0.5) = \alpha_{out} A(0.5) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{out} \end{bmatrix}$$

$$f_b^c = \alpha_{in} N^T(0) A(0) T_{in} + \alpha_{out} N^T(0.5) A(0.5) T_{out} = \begin{bmatrix} \alpha_{in} T_{in} \\ 0 \\ 0 \\ \alpha_{out} T_{out} \end{bmatrix}$$

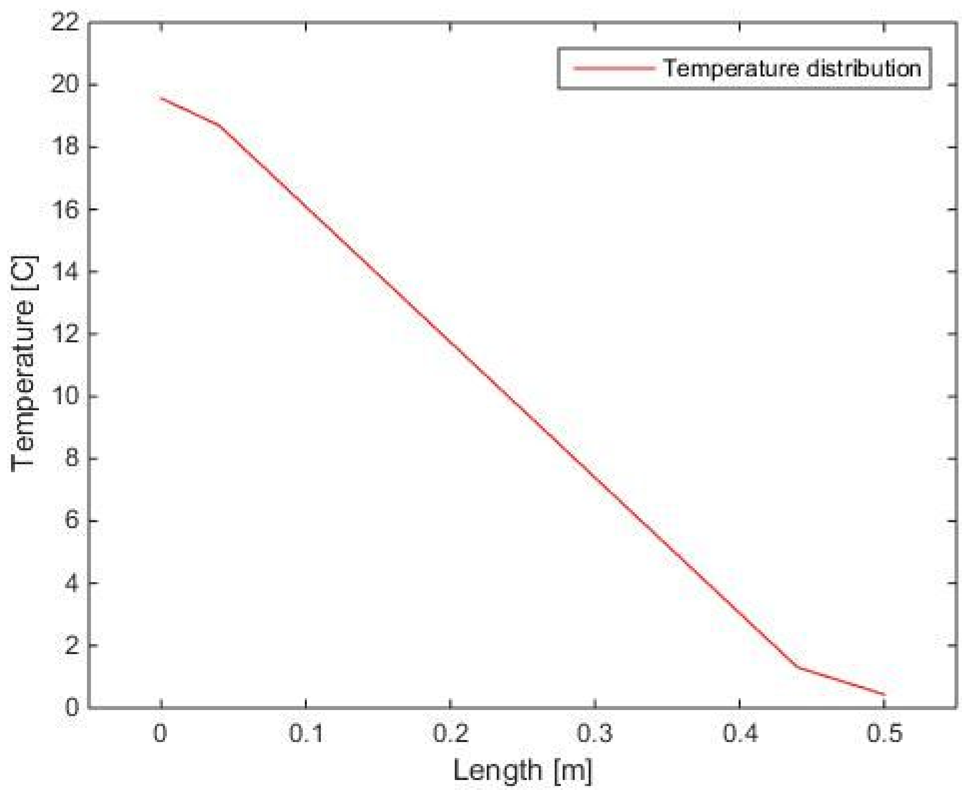
### d)

```
L=[0; 0.04; 0.44; 0.5]; %Length vector[m]
f= zeros(4,1);
K = zeros(4);
k= [0.2 ; 0.1 ; 0.3]; %heat conductivity coeffs. [W/(m*C)]
aIn = 10; %Convection coeff.[W/(m^2*C)]
aOut = 10; %Convection coeff.[W/(m^2*C)]
tIn = 20; %Indoor temp [C]
tOut = 0; %Outdoor temp[C]
A = 1; %Areas A1=A2=A3=A [m2]

%Defining the Kc matrices
Kc1 = zeros(4);
Kc2 = zeros(4);
Kc1(1,1) = aIn*A;
Kc2(4,4) = aOut*A;

fb=zeros(4,1);
fb(1,1)= tIn*aIn*A;
fb(4,1)= tOut*aOut*A;

%Topology matrix
Edof = [1 1 2
       2 2 3
       3 3 4];

%Coord and dof is not needed since it's such a simple structure

%Material properties for each element

Ep = [A*k(1)/(L(2)-L(1))
       A*k(2)/(L(3)-L(2))
       A*k(3)/(L(4)-L(3))];

  for i=1:3
       Ke = bar1e(Ep(i));
       K(Edof(i,2:3),Edof(i,2:3)) = K(Edof(i,2:3),Edof(i,2:3))+ Ke;
  end

  K =K+Kc1+Kc2;
```

```
%The temperatures at the nodes [C]
a=K\fb;
```

```
a = 19.5652
    18.6957
    1.3043
    0.4348
```

```
%Plot the temperatures
plot(L,a,'red');
axis([-0.05 0.55 0 22])
xlabel('Length [m]')
ylabel('Temperature [C]')
legend('Temperature distribution','Location','northeast')
```




The results are reasonable, the temperature does not exceed 20 degrees or being less than 0 degrees in any wall segments. The temperature constantly decreases from left to right which is expected.

```
%Heat loss
%Heat loss is same through all wall elements q=-dT/dL
%(dt temp diff of element dL thickness of element)

q= zeros(3,1);
for i =1:3
     q(i) = -k(i)*(a(i)-a(i+1))/(L(i+1)-L(i));
end
```

**q = -4.3478 W/m^2**

## e)

Adding more nodes and elements in the system would not affect our solution. Since we are using a linear C-matrix the temperature difference between the nodes are linear. If we would have used a different C-matrix that was not linear the results would be different with more nodes, but the linear C-matrix seems most appropriate for this task since temperature gradients in homogenous materials are considered to be linear.

# Task 3

a) $\frac{d}{dx}\left(A(x)E\cdot\frac{du}{dx}\right)+b=0 \qquad (1)$ ①

$u(0)=0$

$\varepsilon(L)=0$

b) Multiply eq. 1 with a weight function $v(x)$ to obtain:

$$v\left[\frac{d}{dx}\left(A(x)E\frac{du}{dx}\right)+b\right]=0 \qquad (2)$$

Integrate over the domain $0\le x\le L$

$$\int_0^L v\left[\frac{d}{dx}\left(A(x)E\frac{du}{dx}\right)+b\right]dx=0 \qquad (3)$$

We use the chain rule and integration by parts to obtain:

$$\int_0^L v\frac{d}{dx}\left(A(x)E\frac{du}{dx}\right)dx=\left[vA(x)E\frac{du}{dx}\right]_0^L-\int_0^L\frac{dv}{dx}A(x)E\frac{du}{dx}dx \quad (4)$$

By combining (3) & (4) we obtain:

$$-\int_0^L\left(\frac{dv}{dx}A(x)E\frac{du}{dx}\right)dx+\left[v(x)A(x)E\frac{du}{dx}\right]_0^L+\int_0^L v(x)b\,dx=0 \qquad (5)$$

$$\left[v(x)A(x)E\frac{du}{dx}\right]_0^L=\left(v(x)A(x)E\frac{du}{dx}\right)_{x=L}-\left(v(x)A(x)E\frac{du}{dx}\right)_{x=0} \quad (6)$$

②

using $\frac{du}{dx} = \mathcal{E}$ and (6) in (5):

$$\int_0^L \left( \frac{dv}{dx} A(x) E \frac{du}{dx} \right) dx = \left( v(x) A(x) E \, \mathcal{E}(x) \right)_{x=L} - \left( v(x) A(x) E \, \mathcal{E}(x) \right)_{x=0} + \int_0^L v(x) b \, dx \qquad (7)$$

we define $\quad V(x) = \mathbb{N} \mathbb{C}^{\text{- constant}} \Rightarrow \frac{dv}{dx} = \mathbb{B} \mathbb{C}, \quad \mathbb{B} = \left[ \frac{dN_1}{dx} \ \frac{dN_2}{dx} \ \frac{dN_n}{dx_n} \right]$

$$u = \mathbb{N} a_1 \Rightarrow \frac{du}{dx} = \mathbb{B} a_1$$

$$N^e = \left[ N_I^e \ N_J^e \right] \qquad N_I^e = -\frac{1}{L}(x - x_J)$$

$$N_J^e = \frac{1}{L}(x - x_i)$$

Insert in (7):

$$\int_0^L \mathbb{B} \mathbb{C} A(x) E \mathbb{B} \, dx \, a_1 = \left( \mathbb{N} \mathbb{C} A(x) E \, \mathcal{E}(x) \right)_{x=L} - \left( \mathbb{N} \mathbb{C} A(x) E \, \mathcal{E}(x) \right)_{x=0} + \int_0^L \mathbb{N} \mathbb{C} \, b \, dx$$

should hold for any $\mathbb{C}$

we take the transpose and can remove $c^T$ from all terms

since $\mathcal{E}(L) = 0$ we remove that term

$$\underbrace{\int_0^L \mathbb{B}^T A(x) E \mathbb{B} \, dx}_{\mathbb{K}} \, a_1 = -\left( \mathbb{N}^T A(x) E \, \mathcal{E}(x) \right)_{x=0} + \underbrace{\int \mathbb{N}^T b \, dx}_{f_L}$$

$$\mathbb{K} = E \int_0^L \begin{bmatrix} -\frac{1}{L} \\ -\frac{1}{L} \end{bmatrix} A(x) \begin{bmatrix} -\frac{1}{L} & -\frac{1}{L} \end{bmatrix} dx = \frac{E A_0}{L^{e2}} \int_0^L \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \left( 1 - \frac{3x}{4L} \right) dx =$$

$$= \frac{E A_0}{L^{e2}} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \int_0^L \left( 1 - \frac{3x}{4L} \right) dx$$

(5)

$$\mathbb{K}^e = \frac{EA_0}{L_e^2}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\int_{x_i}^{x_j}\left(1-\frac{3x}{4L}\right)dx \qquad (8)$$

$$\int_{x_i}^{x_j}\left(1-\frac{3x}{4L}\right)dx = \left[x-\frac{3x^2}{8L}\right]_{x_i}^{x_j} = x_j - \frac{3x_j^2}{8L} - x_i + \frac{3x_i^2}{8L} \underset{x_j-x_i=L^e}{=} L_e - \frac{3}{8L}\left(x_j^2-x_i^2\right) = L_e - \frac{3}{8L}\underbrace{(x_j-x_i)}_{L_e}(x_j+x_i) =$$

$$= L_e\left(1-\frac{3}{8L}(x_j+x_i)\right) \qquad (a)$$

Insert (a) in (8):

$$\underline{K^e} = \frac{A_0E}{L_e^{\cancel{2}}}\cancel{L_e}\left(1-\frac{3}{8L}(x_j+x_i)\right)\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} = \underline{\frac{A_0E}{L_e}\left(1-\frac{3}{8L}(x_j+x_i)\right)\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}}$$

$$f_L^e = \int_{x_i}^{x_j} N^T b\,dx = \begin{bmatrix}\frac{1}{L_e}(x-x_j)\\ \frac{1}{L_e}(x-x_i)\end{bmatrix}\rho g A(x)\,dx = \frac{\rho g A_0}{L^e}\int_{x_i}^{x_j}\begin{bmatrix}x_j-x\\ x-x_i\end{bmatrix}\left(1-\frac{3x}{4L}\right)dx \qquad (10)$$

$$\int_{x_i}^{x_j}\begin{bmatrix}x_j-x\\ x-x_i\end{bmatrix}\left(1-\frac{3x}{4L}\right)dx = \begin{bmatrix}-\frac{x^2}{2}+\frac{x^3}{4L}+x_jx-\frac{3x^2x_j}{8L}\\ \frac{x^2}{2}-\frac{x^3}{4L}-x_ix+\frac{3x^2x_i}{8L}\end{bmatrix}_{x_i}^{x_j} =$$

$$= \begin{bmatrix}-\frac{x_j^2}{2}+\frac{x_j^3}{4L}+x_j^2-\frac{3x_j^3}{8L}+\frac{x_i^2}{2}-\frac{x_i^3}{4L}-x_jx_i+\frac{3x_i^2x_j}{8L}\\ \frac{x_j^2}{2}-\frac{x_j^3}{4L}-x_ix_j+\frac{3x_j^2x_i}{8L}-\frac{x_i^2}{2}+\frac{x_i^3}{4L}+x_i^2-\frac{3x_i^3}{8L}\end{bmatrix} =$$

$$= \begin{bmatrix}\underline{\frac{x_j^2}{2}}-\frac{x_j^3}{8L}+\underline{\frac{x_i^2}{2}}-\frac{x_i^3}{4L}-\underline{x_jx_i}+\frac{3x_i^2x_j}{8L}\\ \underline{\frac{x_i^2}{2}}-\frac{x_i^3}{8L}+\underline{\frac{x_j^2}{2}}-\frac{x_j^3}{4L}-\underline{x_jx_i}+\frac{3x_j^2x_i}{8L}\end{bmatrix} = \begin{bmatrix}\frac{(x_j-x_i)^2}{2}-\left(\frac{x_j^3}{8L}+\frac{x_i^3}{4L}-\frac{3x_i^2x_j}{8L}\right)\\ \frac{(x_j-x_i)^2}{2}-\left(\frac{x_i^3}{8L}+\frac{x_j^3}{4L}-\frac{3x_j^2x_i}{8L}\right)\end{bmatrix}$$

The expression above is inserted into (4)
(10) and tested numerically and gives the
same results as $f_l^e = \frac{\rho g A_0 L_e}{2}\begin{bmatrix} 1 - \frac{3(2x_i + x_j)}{12L} \\ 1 - \frac{3(x_i + 2x_j)}{12L} \end{bmatrix}$

## c)

```
%Chosen values
p =700; %density [kg/m^3]
g = 9.82; %gravitational constant [m/s^2]
L = 10; % Total length of bar [m]
A0 =0.05; % Initial area [m^2]
E= 200*10^9; %Youngs modulus [N/m^2]
nodes = 10; %Number of nodes
elements = nodes-1; %Number of elements

K=zeros(nodes);
fl=zeros(nodes,1);

%boundary conditions
bc=[1 0];

for i = 1:elements
    xi= (i-1)*L/elements; %Start node of element
    xj= i*L/elements; %End node of element
    Le = xj-xi; %Element length
    Ke =(A0*E/Le)*(1-3*(xi+xj)/(8*L))*[1 -1;-1 1];
    K(i:i+1,i:i+1) = K(i:i+1,i:i+1) + Ke;
    fe = p*g*A0*Le*0.5*[1-3*(2*xi+xj)/(12*L); 1-3*(xi+2*xj)/(12*L)];
    fl(i:i+1) = fl(i:i+1) + fe;
end
```

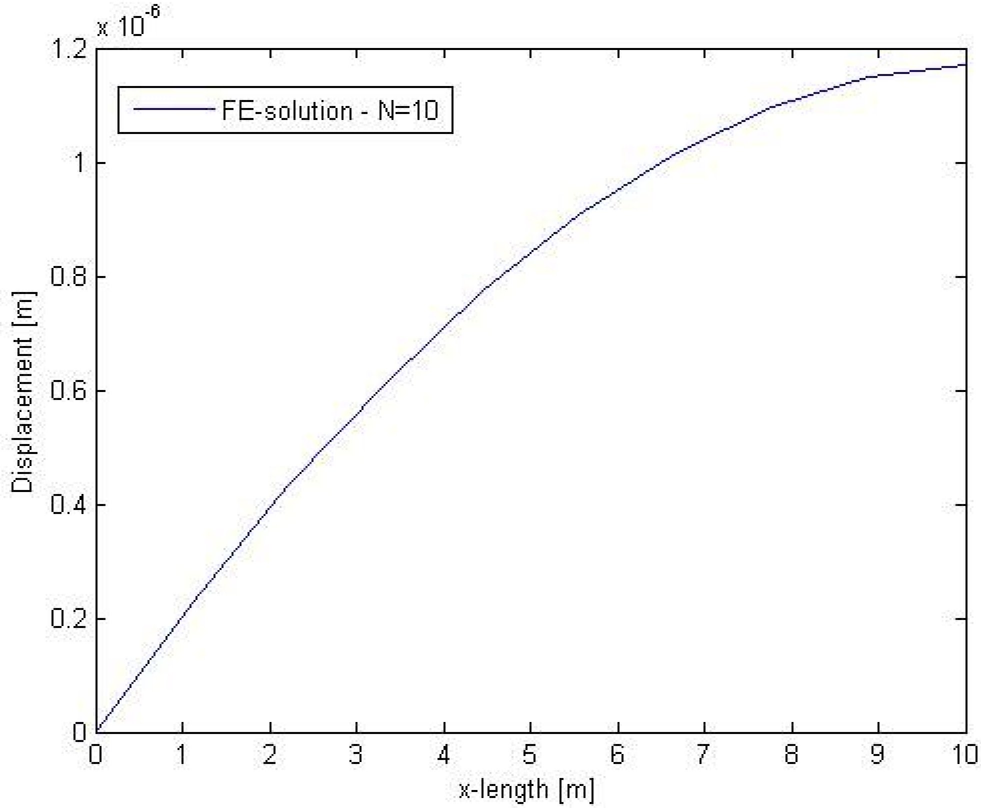
```
%Solving the displacement
[a r] = solveq(K,fl,bc);
```

```
a =  1.0e-05 *

         0
    0.0228
    0.0432
    0.0614
    0.0772
    0.0906
    0.1015
    0.1097
    0.1150
    0.1171
```

Displacement at x=L is 0.1171*10^-5 meter

```
%Plot the displacement
x= linspace(0,L,nodes);
plot(x,a);
xlabel('x-length [m]')
ylabel('Displacement [m]')
```

```
legend('FE-solution - N=10','Location','northwest')
```




### d)

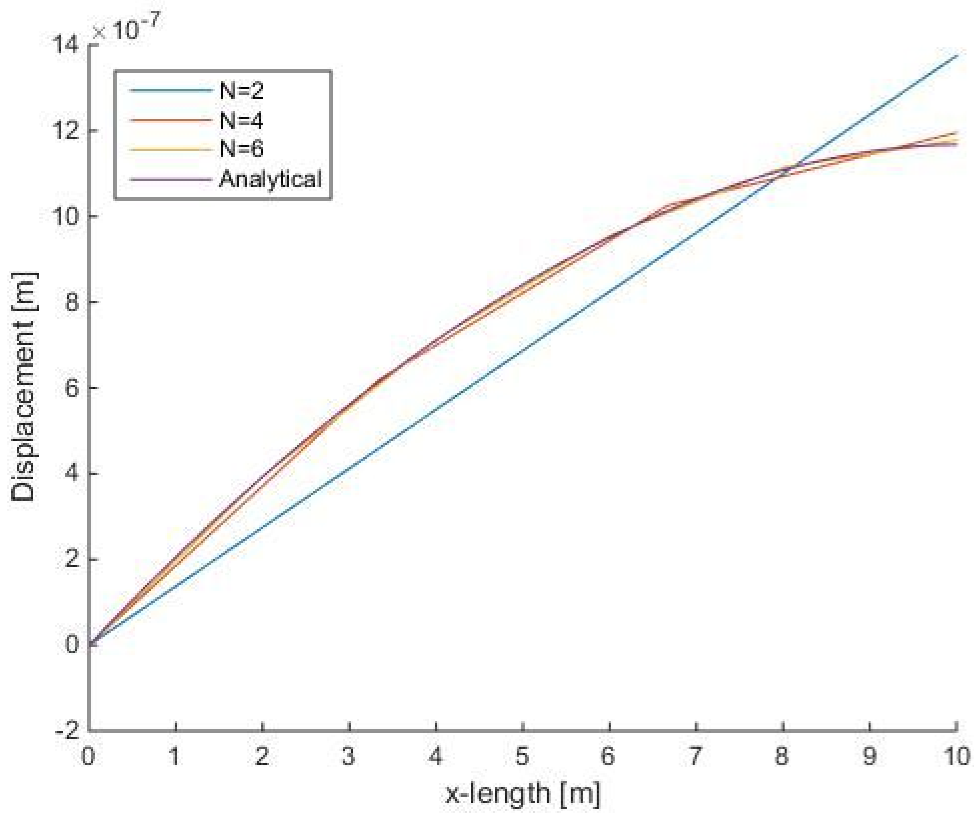
```
%Few values for number of nodes
nodes = [2 4 6];
diff = zeros(length(nodes),1);
for i = 1:length(nodes)

       %function for calculating the displacements
       a = BarFunc(nodes(1,i));
       x= linspace(0,L,nodes(i));

       %Analytical method
       u = (g*p*((2*L*x)/3 - x.^2/4 + (8*L^2*log(x - (4*L)/3))/9))/E -
(8*L^2*g*p*log(-(4*L)/3))/(9*E) - (5*L^2*g*p*log(1 - (3*x)/(4*L)))/(6*E);

       %Difference for last node of both the numerical and analytical method
       diff(i) = (a(end)-u(end))/u(end);
       plot(x,a);
end
```

```
%Plot the analytic function with 100 nodes
x= linspace(0,L);
u = (g*p*((2*L*x)/3 - x.^2/4 + (8*L^2*log(x - (4*L)/3))/9))/E -
(8*L^2*g*p*log(-(4*L)/3))/(9*E) - (5*L^2*g*p*log(1 - (3*x)/(4*L)))/(6*E);
plot(x,u);
xlabel('x-length [m]')
ylabel('Displacement [m]')
legend('N=2','N=4','N=6','Analytical','Location','northwest')
```

BarFunc which is called in every loop  is an own made fucntion and looks like following:

```
function [a] = BarFunc(nodes)

% d)
%Chosen values
p =700; %density
g = 9.82; %gravitational constant
L = 10;
A0 =0.05; %m2
E= 200*10^9; %Pa
elements = nodes-1; %Number of elements

K=zeros(nodes);
fl=zeros(nodes,1);

%boundary conditions
bc=[1 0];

for i = 1:elements
       xi= (i-1)*L/elements;
       xj= i*L/elements;
       Le = xj-xi;
       Ke =(A0*E/Le)*(1-3*(xi+xj)/(8*L))*[1 -1;-1 1];
       K(i:i+1,i:i+1) = K(i:i+1,i:i+1) + Ke;


       fe = p*g*A0*Le*0.5*[1-3*(2*xi+xj)/(12*L); 1-3*(xi+2*xj)/(12*L)];
       fl(i:i+1) = fl(i:i+1) + fe;
end
```

```
[a r] = solveq(K,fl,bc);
```




The differences of the displacement in the last node between FE- and analytical solution with an increasing number of nodes(N=2, N=4, N=6)

diff =

0.1777

0.0240

0.0089

From the FE-solutions with different N values we can see that the FE solution converges with an increasing number of nodes. A model with 6 nodes gives a error margin of less than 1% which we think is enough. See appendix for calculations

# APPENDIX

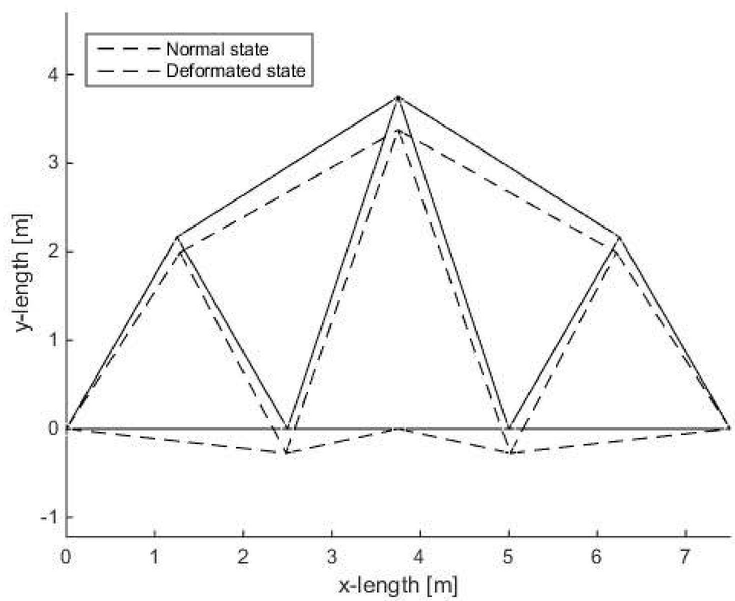
```
%=========================================================================
% Project 1
% Gustav Good och Jimmie Andersson
% 2015-11-06
%VSM-167
%=========================================================================
%Task 1

clc
clf

L =2.5; %Length [m]
f= zeros(16,1);
K = zeros(16);
d= 0.1; %Diameter [m]
E = 210*10^9; % Youngs modulus[N/m^2]
A = pi*(d/2)^2; %Area of bar [m^2]
a0 = 0.006; %Displacement of node (for task c)

%a)

%Topology matrix
Edof = [1 1 2 3 4
        2 1 2 5 6
        3 3 4 5 6
        4 3 4 9 10
        5 5 6 9 10
        6 5 6 7 8
        7 7 8 11 12
        8 9 10 11 12
        9 9 10 13 14
        10 11 12 13 14
        11 11 12 15 16
        12 13 14 15 16];

%Boundary conditions
bc = [1 0
      2 0
      8 0
      15 0
      16 0
      10 -a0];

%Rest of task a) can be found at the attached hand written paper

%c)

Dof = [1 2
        3 4
        5 6
        7 8
        9 10
```

1

```
        11 12
        13 14
        15 16];

%Origo at node1, node2
Coord = [0 0
        L/2 sqrt(3)*L/2
        L 0
        3*L/2 0
        3*L/2 1.5*L
        2*L 0
        2.5*L sqrt(3)*L/2
        3*L 0];

  %Ex &Ey is the coords of all the elements
  %2 is number of element dof's
  [Ex,Ey] = coordxtr(Edof,Coord,Dof,2);

  %Material properties for each element
  Ep = [E A];

  for i=1:12
      Ke = bar2e(Ex(i,:),Ey(i,:),Ep(1,:));
      K(Edof(i,2:5),Edof(i,2:5)) = K(Edof(i,2:5),Edof(i,2:5))+ Ke;
  end

  %Getting the reaction forces and deformations
  [a,r] = solveq(K,f,bc);
  Ed=extract(Edof,a);

  %The wanted force
  F = r(10);

  %F = -2.4400e+06 N

  %Drawing the original state and the deformation
  eldisp2(Ex,Ey,Ed,[2 6 1]);
  eldraw2(Ex,Ey,[1 6 1]);
  xlabel('x-length [m]')
  ylabel('y-length [m]')
  legend('Normal state','Deformated state','Location','northwest')

  %d)
  %The rest of the forces are in the r matrix. Each row represents the
  %force at the degree of freedom with that row-index.
  %If the sum of all the forces in x direction and y direction is zero
  % and the total moment is zero it is equilibrium.
  rx =0;
  ry=0;
  Mtot=0

  for i=1:length(r)/2
        r1 = r(i*2-1);
        r2 = r(i*2);
```

```
        rx = rx + r1;
        ry = ry + r2;
        Mtot= Mtot+ r1*Coord(i,2); %Rx * length in y-dir
        Mtot= Mtot+ r2*Coord(i,1);
    end

    %The sum is very close to zero (not zero beacuse of beacuse of rounding
    %errors) so we have equilibrium
```

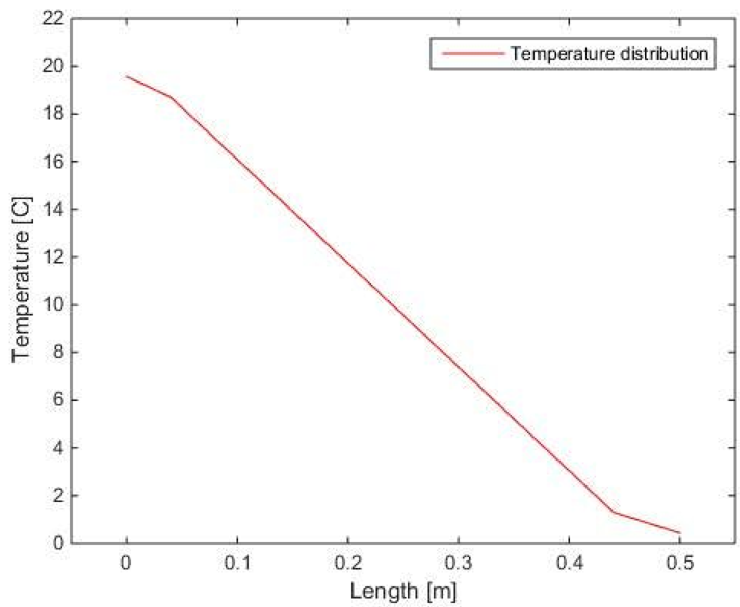
```
Mtot =

     0
```




*Published with MATLAB® R2014b*

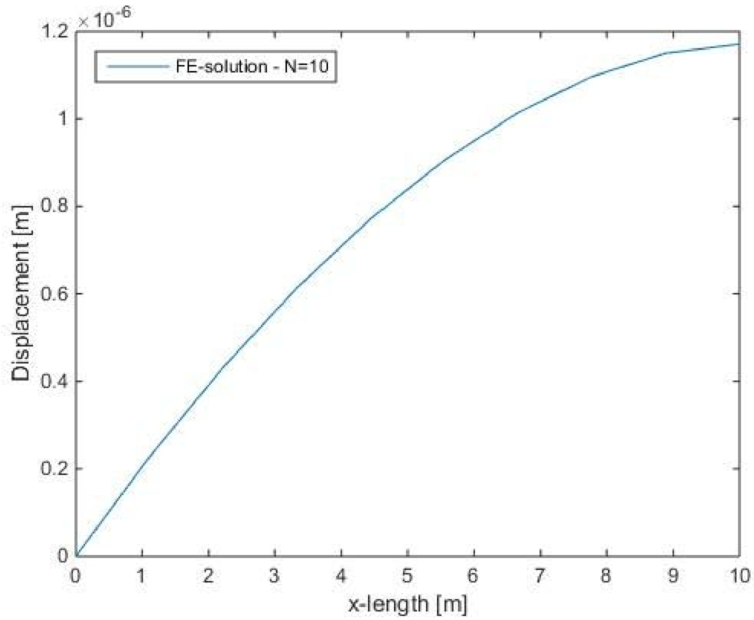
```
%=========================================================================
% Project 1
% Gustav Good och Jimmie Andersson
% 2015-11-13
%VSM-167
%=========================================================================
%Task 2
% d)
clc
clf

L=[0
   0.04
   0.44
   0.5]; %Length vector[m]
f= zeros(4,1);
K = zeros(4);
k= [0.2
   0.1
   0.3]; %heat conductivity coeffs. [W/(m*C)]
aIn = 10; %Convection coeff.[W/(m^2*C)]
aOut = 10; %Convection coeff.[W/(m^2*C)]
tIn = 20; %Indoor temp [C]
tOut = 0; %Outdoor temp[C]
A = 1; %Areas A1=A2=A3=A [m2]

%Defining the Kc matrices
Kc1 = zeros(4);
Kc2 = zeros(4);
Kc1(1,1) = aIn*A;
Kc2(4,4) = aOut*A;

fb=zeros(4,1);
fb(1,1)= tIn*aIn*A;
fb(4,1)= tOut*aOut*A;

%Topology matrix
Edof = [1 1 2
        2 2 3
        3 3 4];

%Coord and dof is not needed since it's such a simple structure

%Material properties for each element
Ep = [A*k(1)/(L(2)-L(1))
        A*k(2)/(L(3)-L(2))
        A*k(3)/(L(4)-L(3))];

  for i=1:3
      Ke = bar1e(Ep(i));
      K(Edof(i,2:3),Edof(i,2:3)) = K(Edof(i,2:3),Edof(i,2:3))+ Ke;
  end
```

```
K =K+Kc1+Kc2;

%The temperatures at the nodes
a=K\fb;

%Plot the temperatures
plot(L,a,'red');
axis([-0.05 0.55 0 22])
xlabel('Length [m]')
ylabel('Temperature [C]')
legend('Temperature distribution','Location','northeast')

%Heat loss

%Heat loss is same through all wall elements q=-dT/dL
%(dt temp diff of element dL thickness of element)
q= zeros(3,1);
for i =1:3
    q(i) = -k(i)*(a(i)-a(i+1))/(L(i+1)-L(i));
end
```



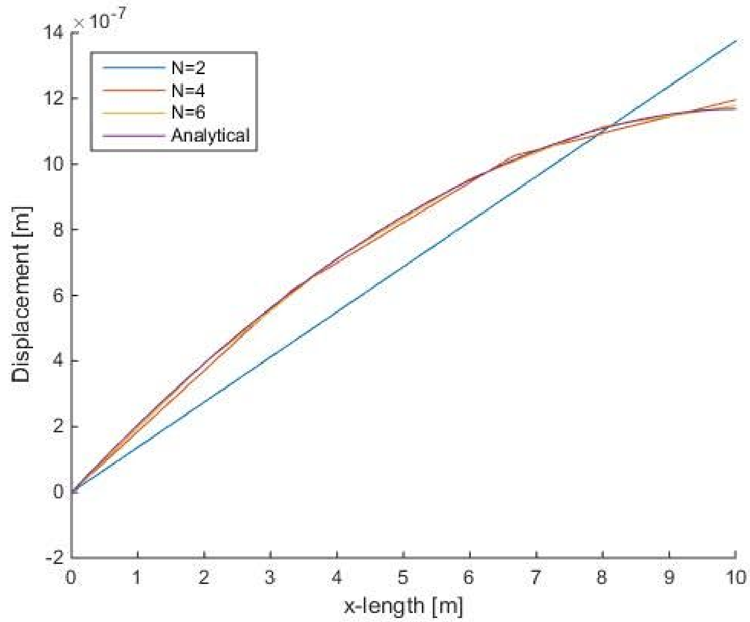

*Published with MATLAB® R2014b*

```matlab
%==========================================================================
% Project 1
% Gustav Good och Jimmie Andersson
% 2015-11-20
%VSM-167
%==========================================================================
%Task 3
% c)
clc
clear all
clf

%Chosen values
p =700; %density [kg/m^3]
g = 9.82; %gravitational constant [m/s^2]
L = 10; % Total length of bar [m]
A0 =0.05; % Initial area [m^2]
E= 200*10^9; %Youngs modulus [N/m^2]
nodes = 10; %Number of nodes
elements = nodes-1; %Number of elements

K=zeros(nodes);
fl=zeros(nodes,1);

%boundary conditions
bc=[1 0];

for i = 1:elements
    xi= (i-1)*L/elements; %Start node of element
    xj= i*L/elements; %End node of element
    Le = xj-xi; %Element length
    Ke =(A0*E/Le)*(1-3*(xi+xj)/(8*L))*[1 -1;-1 1];
    K(i:i+1,i:i+1) = K(i:i+1,i:i+1) + Ke;

    fe = p*g*A0*Le*0.5*[1-3*(2*xi+xj)/(12*L); 1-3*(xi+2*xj)/(12*L)];
    fl(i:i+1) = fl(i:i+1) + fe;

end

%Solving the displacement
[a r] = solveq(K,fl,bc);

%Plot the displacement
x= linspace(0,L,nodes);
plot(x,a);
xlabel('x-length [m]')
ylabel('Displacement [m]')
legend('FE-solution - N=10','Location','northwest')
```




```
%d)
clc
clf
hold on

%Few values for number of nodes
nodes = [2 4 6];
diff = zeros(length(nodes),1);
for i = 1:length(nodes)

    %function for calculating the displacements
    a = BarFunc(nodes(1,i));
    x= linspace(0,L,nodes(i));

    %Analytical method
    u = (g*p*((2*L*x)/3 - x.^2/4 + (8*L^2*log(x - (4*L)/3))/9))/E - (8*L^2*g*p*log

    %Difference for last node of both the numerical and analytical method
    diff(i) = (a(end)-u(end))/u(end);
    plot(x,a);
end

%Plot the analytic function with 100 nodes
x= linspace(0,L);
u = (g*p*((2*L*x)/3 - x.^2/4 + (8*L^2*log(x - (4*L)/3))/9))/E - (8*L^2*g*p*log(-(4
plot(x,u);
```

```
xlabel('x-length [m]')
ylabel('Displacement [m]')
legend('N=2','N=4','N=6','Analytical','Location','northwest')

hold off
```




*Published with MATLAB® R2014b*