Assignment Part 2

Computational Engineering 1 MT2526

Group 6

Akhil Mora – 9507286954

Raghavendra Machipeddi – 9506066894

Anton Karlsson – 9303051230

Erik Ising – 9306020018

Abstract

This assignment is the second one of two parts in the course *Computational Engineering 1* and this is a group assignment, where the tasks are individually graded by how much effort each member put into the work. The tasks are solved by appointed groups where the grades also depends on the group performance. One part of the assignment are therefore to include a self-evaluation of all group members using rates to get the individually grades. The main purpose of the assignments in this course is to facilitate learning of the theory covered for the whole course, where most of the solutions are solved by Matlab.

The assignment was to find out the maximum deflection of the given roller supported cantilever beam, part of which is subjected to a varying load using finite difference method and finite element method. Euler-Bernoulli beam theory is used to analyze the deflection of the beam using Matlab. The deflection is also found out in COMSOL Multiphysics using two beam theories, Euler-Bernoulli and Timoshenko theory. The appropriate beam theory is suggested for the given cantilever beam. We also find an optimized cross section for the beam such that less deflection is obtained when compared to the actual design of the beam.

The maximum deflection of the beam with 72 elements was 1.637mm using finite difference method and 1.668mm using finite element method. The computational time was much longer for the FEM compared with FDM. Using COMSOL, the maximum deflection using Euler-Bernoulli beam theory is 1.653mm and 1.990mm when using Timoshenko Beam theory. Optimizing the design by using uniform cross-section, the maximum deflection was found to be less than that of the original design for a constant height of 0.2139m for the cross-section.

Table of contents

1 Notations

$q = Distributed\ load$

$h = Cross\ sectional\ height$

$h_0 = initial\ cross\ section\ of\ the\ beam$

$w = width\ of\ the\ beam$

$L = length\ of\ the\ beam$

$a = distance\ of\ roller\ from\ the\ fixed\ end$

$E = Young's modulus$

$\upsilon = Poisson's\ ratio$

2 Introduction

This assignment (*Part 2)* consists of a cantilever beam which is subjected to a varying load. The given beam is as shown in *figure 1*.

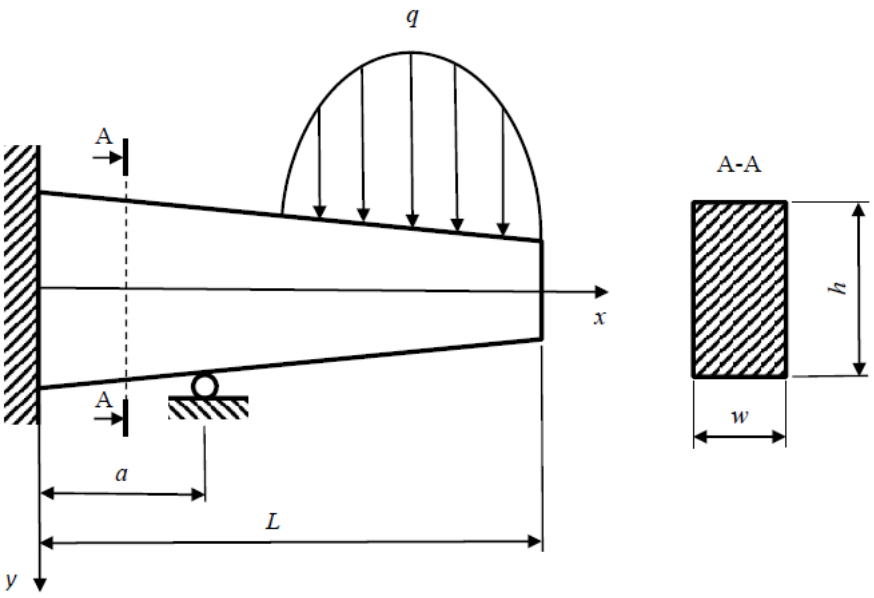


Figure 1: The Cantilever beam

From the above figure we can see the cross section of the given cantilever beam and the load acting on it. The cross sectional height and the load vary according to the length of the beam which are:

$$h = h_0 - h_e\left(\frac{x}{L}\right)$$

$$q = \begin{cases} 0, & 0 \le x < L/2 \\ -q_m * \sin(\frac{2\pi x}{L}), & L/2 \le x \le L \end{cases}$$

The parameter values are as follows:

Parameter	Value
q_m	200 kN/m
L	1,20 m
a	0,40 m
w	0,10 m
h_0	0,30 m
h_e	0,20 m
E	69 GPa
ν	0,33

3 Task 1

3.1 Solution

The first task of this assignment was to apply the Euler-Bernoulli beam theory into MATLAB and then determine the maximum deflection with less than 2% error using the finite difference method. The calculated deflection containing the right shape should also be plotted in a diagram.

3.2 Method

This task was solved using the FDM methodology in MATLAB. Where the differential equation for Euler-Bernoulli Beam theory is used, looking like this:

$$\frac{d^2}{dx^2}\left(EI\frac{d^2w}{dx^2}\right) = q$$

The x is the coordinate along the beam, E is Young's modulus, I is the area moment of inertia (in our case a parameter that is dependent on x), w is the searched deflection and q is the load per unit length.

3.3 Solution

The fourth-order differential equation described above gives a Finite difference approximation in this solution. To solve this FDM-task the *Figure 2* is created to get some extra fictive nodes that is necessary for solving this kind of problem. Those extra nodes is required to use the second-order central difference for higher-order equations.

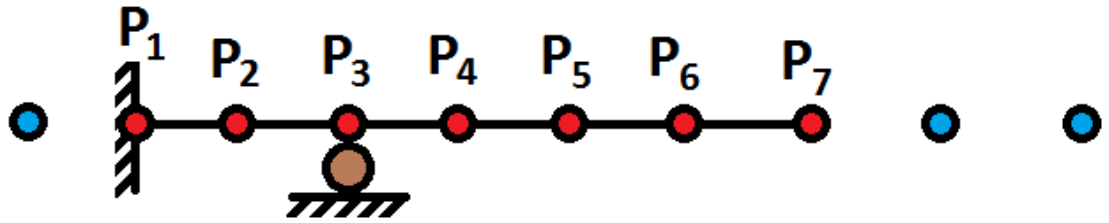


Figure 2: Nodes setup for the beam.

Those new fictive nodes also helps dealing with the given boundary conditions, for example the impact from the support. For the first few points and last points expressions of the central difference has to be derived or solved analytical before solving the problem with help of MATLAB. These relations are derived from the finite difference expressions for the first, second and third derivative. Our expressions are shown below:

$$\frac{dw}{dx} = \frac{w_2 - w_0}{2h} = 0 \quad \rightarrow \quad w_2 = w_0 \qquad eq(1)$$

$$\frac{dM}{dx} \quad \rightarrow \quad w_8 = w_4 \qquad eq(2)$$

For the internal nodes the second-order central difference looks like this:

$$\frac{(EI)_{i-1}}{\Delta x^4} w_{i-2} - 2\frac{(EI)_{i-1}+(EI)_i}{\Delta x^4} w_{i-1} + \frac{(EI)_{i-1+}+4(EI)_i+(EI)_{i+1}}{\Delta x^4} w_i - 2 \cdot \frac{(EI)_i+(EI)_{i+1}}{\Delta x^4} w_{i+1} + \frac{(EI)_{i+1}}{\Delta x^4} w_{i+2} = q_i \qquad eq(3)$$

This is an equation that follow the whole beam and to work with it more easily variable substitution can be made. After this the expression can be written as:

$$Aw_{i-2} - Bw_{i-1} + Cw_i - Dw_{i+1} + Ew_{i+2} = q_i$$

For node 2 we have that i=2, giving:

$$Aw_0 + Bw_1 + Cw_2 + Dw_3 + Ew_4 = \{Using\ eq(1)\} = Bw_1 + (A + C)w_2 - Dw_3 + Ew_4$$

For the second last point, i=5 in accordance to figure 2 giving:

$$Aw_3 + Bw_4 + Cw_5 + Dw_6 + E(2w_6 - w_5) = Aw_3 + Bw_4 + (C - E)w_5 + (D + 2E)w_6$$

And for the last point, i=6 in accordance to figure 2. Using similar calculations and with help from eq.(2) the following can be obtained:

$$(A + E)w_4 + (B + 4E - D)w_5 + (C + 2D + 4E)w_6$$

These equations are then introduced in the MATLAB code to receive a "correct" matrix, see ***appendix*** for the code.

3.4 Results

Our result for this task using the Finite Difference method can be seen in the *table 1* below, for different number of elements, that is also used for comparison between task 1 and task 2.

Elements	**Maximum Deflection [mm]**	**Time to execute code [s]**
18	1.571	0.23
36	1.619	0.25
72	1.637	0.29
144	1.646	0.32
288	1.650	0.34
576	1.652	0.36

Table 1: Result using FDM.

From this table and with different number of elements, we can see that the deflection converges to a steady value. See *figure 3* below for the plotted displacement accroding to the load and bounderys working on the beam.

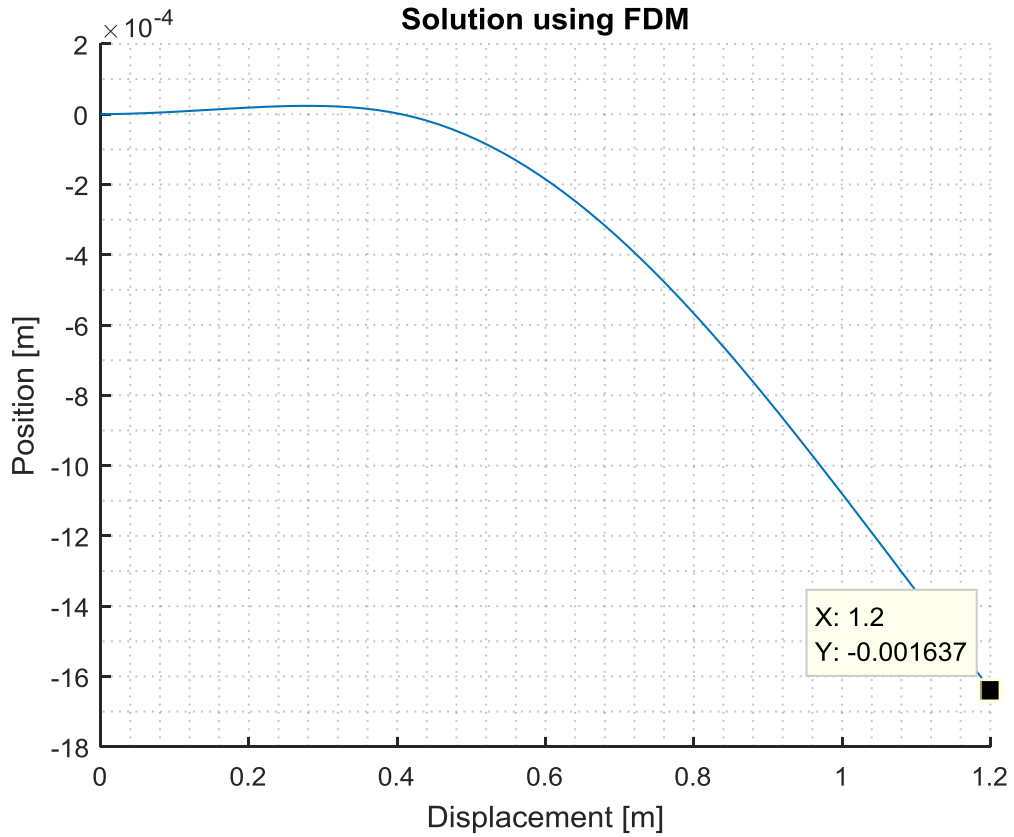


Figure 3: Displacement using FDM with 72 elements.

To guarantee an error of less than 2%, we use 72 elements. More details about this can be found in the discussion.

3.5 Discussion and Conclusions

To see the "correct" displacement value we made a graph showing how the deflection changes when we use different number of elements. We created a graph that shows how the displacement converges, see figure 4. Studying this figure, one can realize already after about 100 elements that the correct value for the deflection at the end point must be somwhere between 1.65-1.66mm.

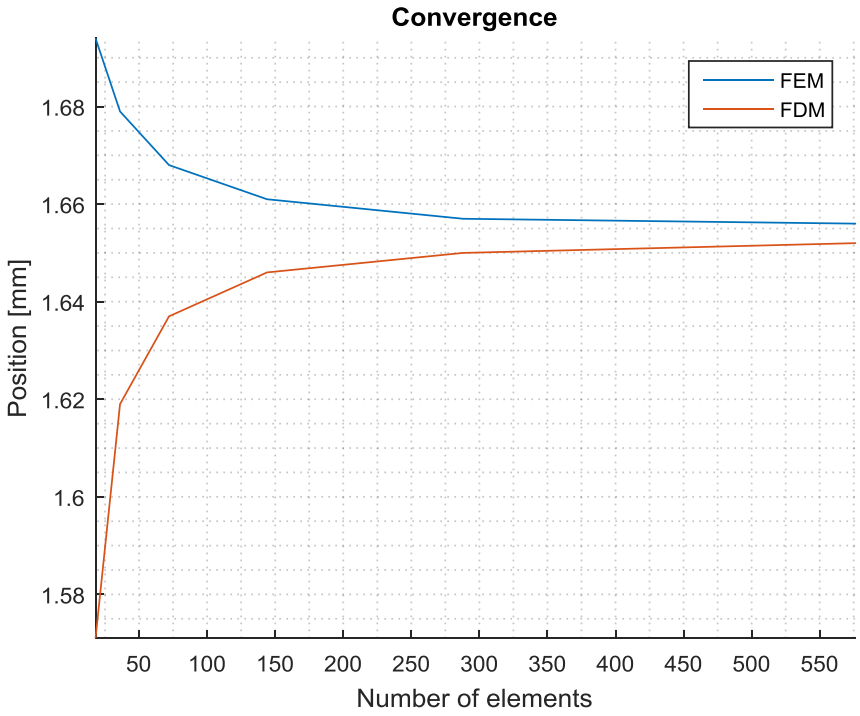


Figure 4: Showing the convergence between the two methods.

To be on the safe, we used the value 1.66 giving that a deflection larger than 1.6268mm is certainly within 2% error. In this simple plot the first value high enough is the value obtained while using 72 elements. Note that we used a really high value of the estimated "true" deflection and that the elements used vary a lot in this plot, so the elements required to obtain an error within 2% is in reallity likely less than 40 elements if we would know the true solution.

4 Task 2

4.1 Problem statement

This task was to use the Finite element method to solve the same problem as in task 1.

4.2 Method

To solve the problem with the given method, chapter 17 in the FEM-book (Ottosen & Petersson, p. 311-334) was used.

4.3 Solution

Following the method in the book, one can create elements where each element can be represented as a 4x4 stiffness matrix. The values in the matrices can be calculated using $B^e = \frac{d^2N^e}{dx^2}$, where N^e is the shape functions. Knowing this, the stiffness matrix K can be calculated as $K_{rc} = EI \cdot \int_0^L B_r^e \cdot B_c^e \ dx$. The load vector can be calculated using $K_u = \int_0^L N_r^e \cdot q \ dx$. After that the boundaries needs to be added. They can be added quite simple by multiplying the first and second diagonal values in the stiffness matrix by a large number, since the values corresponds to the elements displacement and angle respectively. The values can then be set to zero, or very close to zero, by setting the first and second values in the load vector to zero. The support was added in a similar way and then the result could be produced.

4.4 Results

Elements [psc]	Maximum deflection [mm]	Time to execute code [s]
18	1.694	19.32
36	1.679	32.55
72	1.668	71.84
144	1.661	125.32
288	1.657	299.74
576	1.656	602.90

Table 2: Result using FEM.

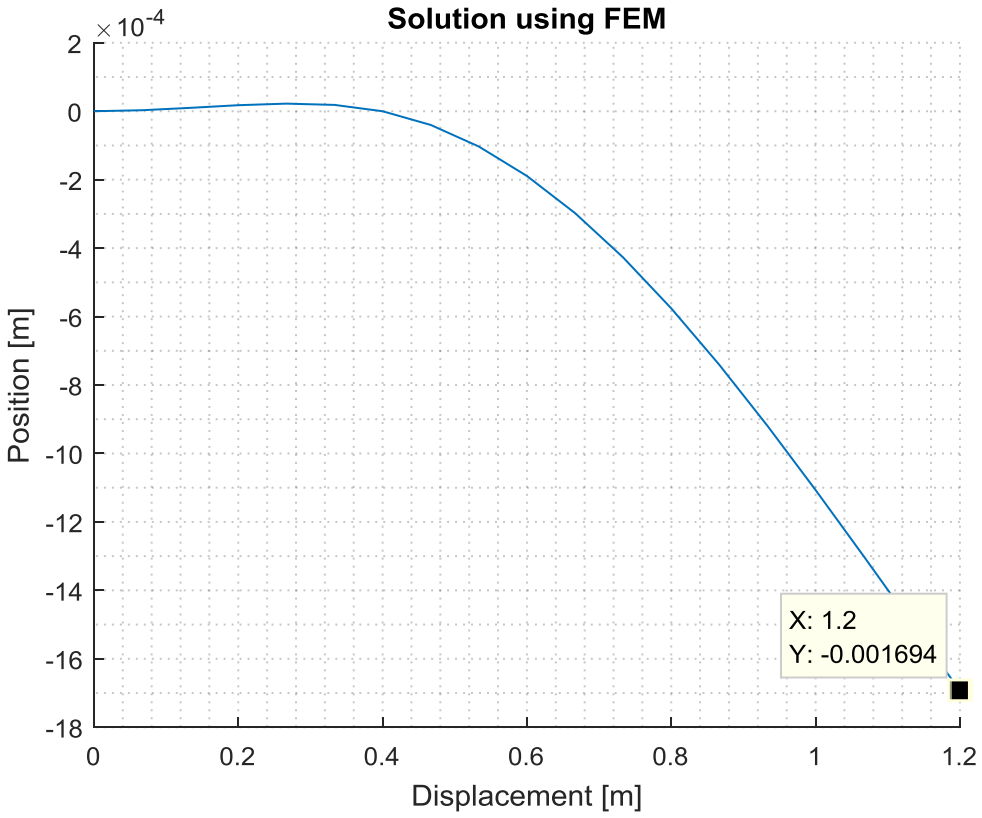


Figure 5: Solution with FEM using 18 elements.

4.5 Discussion and Conclusions

Looking at *figure 4* as, we can see that also the Finite element method convergences to a value. The difference is that with FEM a larger value is obtained and the maximum deflection decreases when using more elements while FDM works the other way around. Looking at the shape of the beam we can draw the conclusion that the method seems to work since the solution looks similar to what we expected before solving the tasks. With a fixed beam using a support somewhere in the middle and a force at the end the shape should be expected to have this kind of shape seen from both *figure 3 & 5*.

5 Task 3

5.1 Problem statement

This task is to compare the computational cost from the two earlier task.

5.2 Method

To solve this task MATLAB's own function "tic/toc" was used. This function returns the computational time between the commands tic and toc.

5.3 Solution

To see the time difference of running the code, between the two different methods, you can see the result from the two tasks before in *table 1 & 2.*

5.4 Result

FDM got a much lower computational cost than FEM. The computational time required for the different methods varies a lot. To calculate 576 elements using FEM took more than 1000 times longer than it did while using FDM.

5.5 Discussion and Conclusions

The reason for the enormous time differences is that FEM uses 4x4 = 16 values for the stiffness for each element while FDM only uses 5 values. The FEM method also need to integrate and derivate several times for each and every element as described in task 2, in the FDM the elements can be obtained straight away which probably is the largest reason for the computational time required.

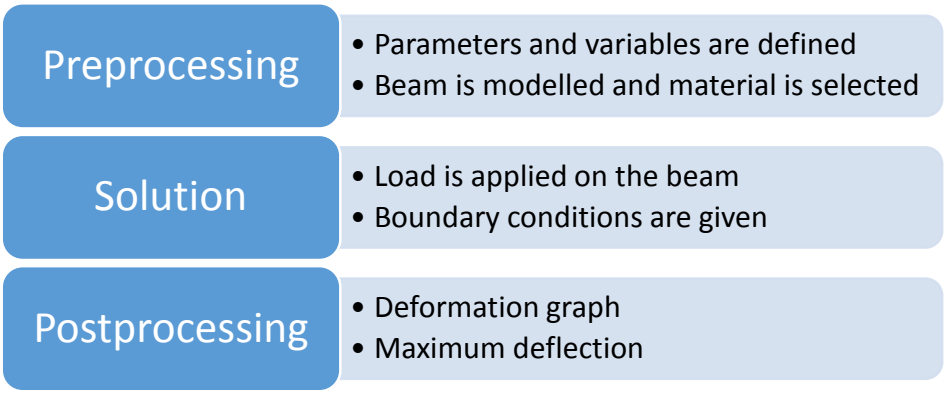
6 Task 4

6.1 Problem statement

The task is to find the maximum deflection of the beam in COMSOL using Euler-Bernoulli and Timoshenko beam theory.

6.2 Method

The beam is considered as a 2D object and solved in COMSOL. The steps involved in methodology are shown below.



6.3 Solution

In the model wizard, a 2D space dimension is selected. In physics, study of structural mechanics is selected where beam interface is used. As the problem is time independent, stationary study is selected. Input parameters and variables are defined.

Name	Expression	Value
qm	200[kN/m]	2E5 N/m
L	1.2[m]	1.2 m
wi	0.1[m]	0.1 m
h0	0.3[m]	0.3 m
he	0.2[m]	0.2 m

Table 3: Parameters.

Name	Expression	Unit
Q	-qm*(sin(2*pi*x/L))	N/m
H	h0-(he*(x/L))	m

Table 4: Variables.

The beam is modelled as a line, cross-sectional data and the material were added to the beam. The beam was then divided into a number of nodes using mesh. The load and boundary

conditions were applied to the beam and solved for results. The method is solved using Euler-Bernoulli and Timoshenko beam theories for deflection of the beam.

6.4 Results

The deflections obtained were 1.65383mm and 1.99045mm while using Euler-Bernoulli and Timoshenko beam theories respectively. The following figures shows the deflection of the beam for the different beam theories.

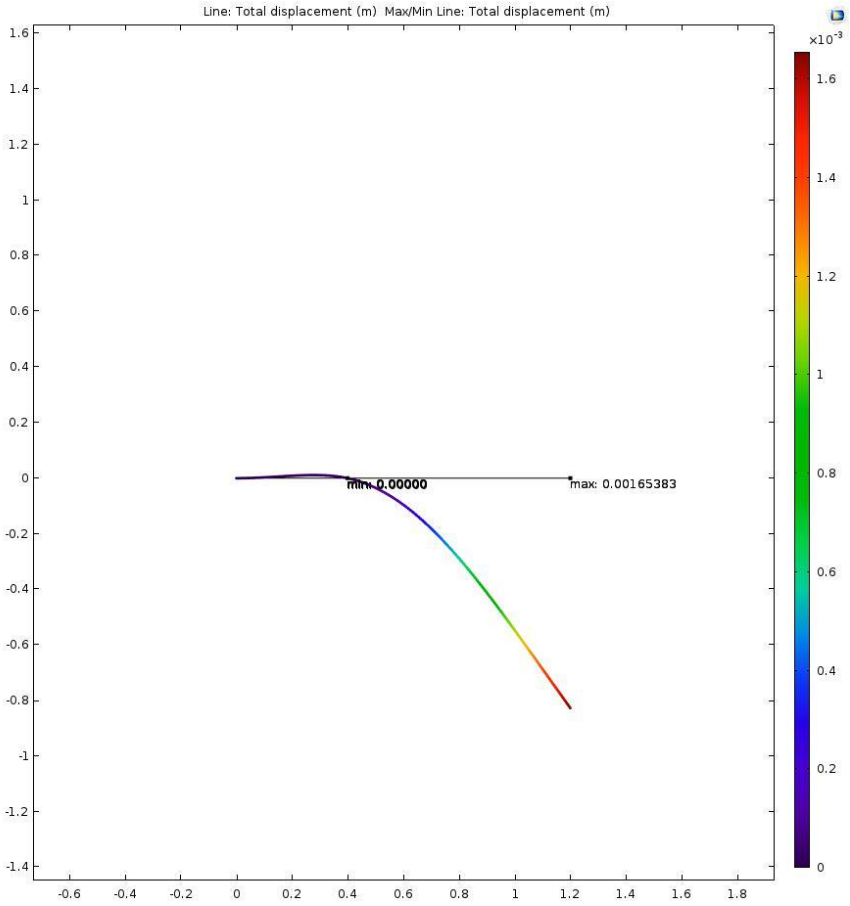


Figure 6: Displacement of the beam using Euler-Bernoulli beam theory

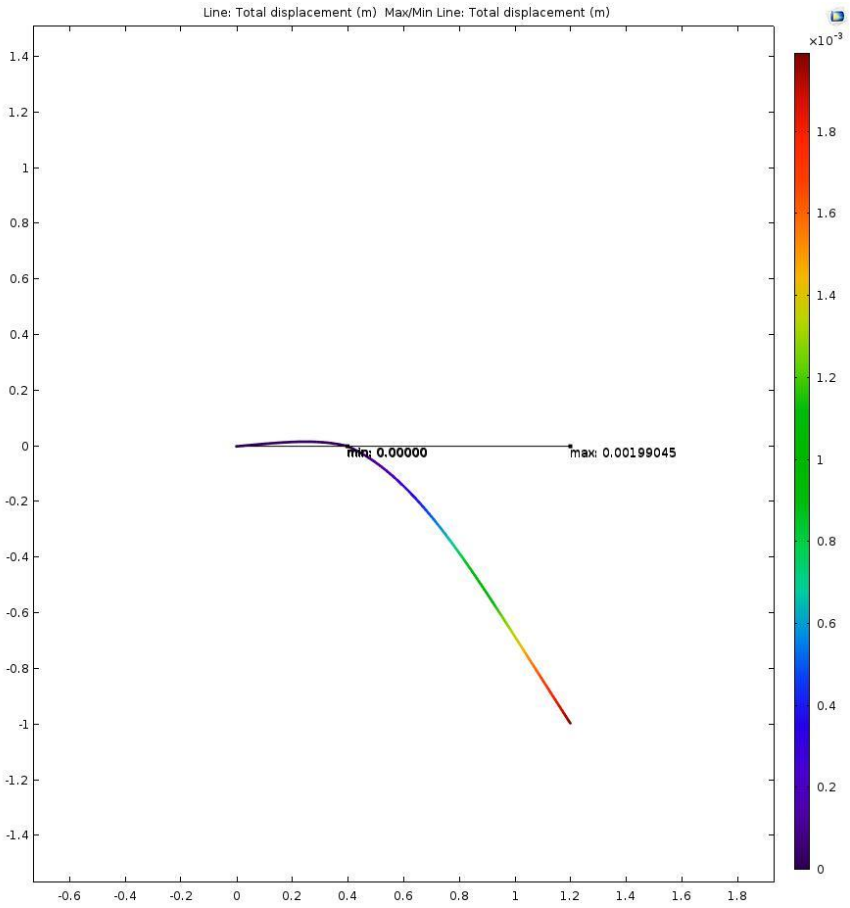


Figure 7: Displacement of the beam using Timoshenko beam theory

6.5 Discussion and Conclusions

Since the influence of shear stresses are negligible in Euler-Bernoulli beam theory, the deflection is comparatively less. We can see that the deflection varies for about 0.3mm for Euler-Bernoulli theory and Timoshenko theory. So for this case we find the Euler-Bernoulli beam theory appropriate as it also saves computational time.

7 Task 5

7.1 Problem statement

The task is to alter the design of the beam to lower the cost of manufacture where it should satisfy the following criteria.

(i) The maximum deflection must be less than or equal to that of the current design, while keeping the weight of the beam as low as possible.

(ii) The width should not be altered.

7.2 Method

As mentioned in the problem, a uniform rectangular cross-section is used instead of a tapered rectangular cross-section. The same method as the previous task was used to solve this problem. We have just replaced the height function with an approximated value such that the deflection of the new beam is less than or equal to that of the original beam. Parametric sweep is used to find out the value of the height such that the deflection is less than or equal to that of the original beam. The range is selected and a step size of 0.0001m is chosen. The problem is solved and the appropriate value of height is selected. This was done for both the theories.

7.3 Solution

The graphs below (figure 8 and 9) were obtained after parametric sweep for the two theories.

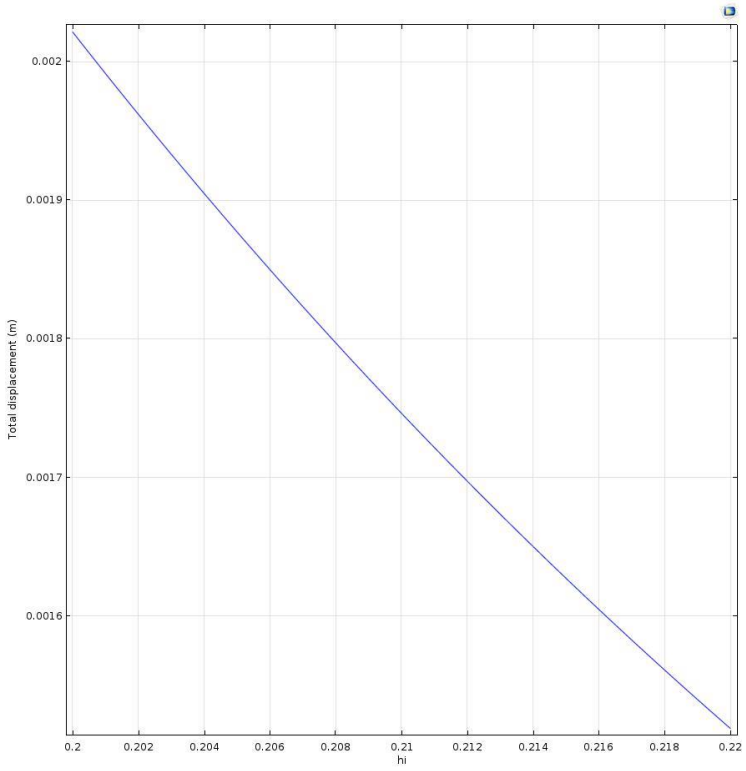


Figure 8: Parametric sweep: Height vs Maximum deflection (Euler-Bernoulli theory)

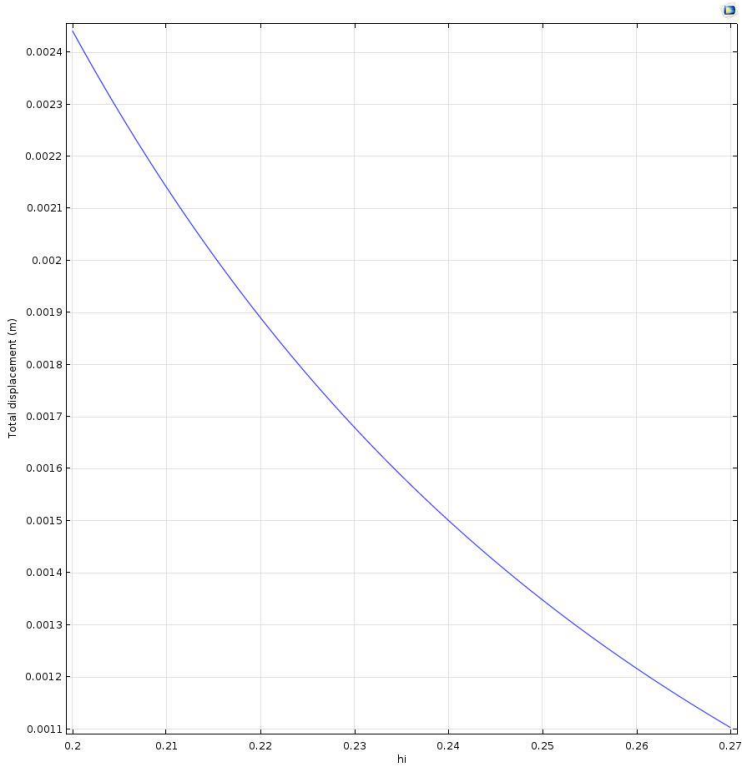


Figure 9: Parametric sweep: Height vs Maximum deflection (Timoshenko theory)

From the above figures we can observe that the required height is in a range of 0.21m to 0.22m. Now, the range of the parametric sweep is changed to the above values using the same step size. We can then identify the height for which the deflection is less compared to the original design.

7.4 Results

The results obtained from parametric sweep for Euler-Bernoulli theory are shown in the below table:

Height(m)	Total displacement(mm)
0.2	2.0215
0.21	1.7463
0.2137	1.6571
0.2139	**1.6525**
0.214	1.6502
0.22	1.5188

Table 5: Total displacement for different heights (Euler- Bernoulli theory)

For Euler-Bernoulli theory, we can from the table above that at a height of 0.2139m, the deflection is observed to be less than the one obtained for the original beam.

The deflection of the beam with a height 0.2139m is shown in the below figure:

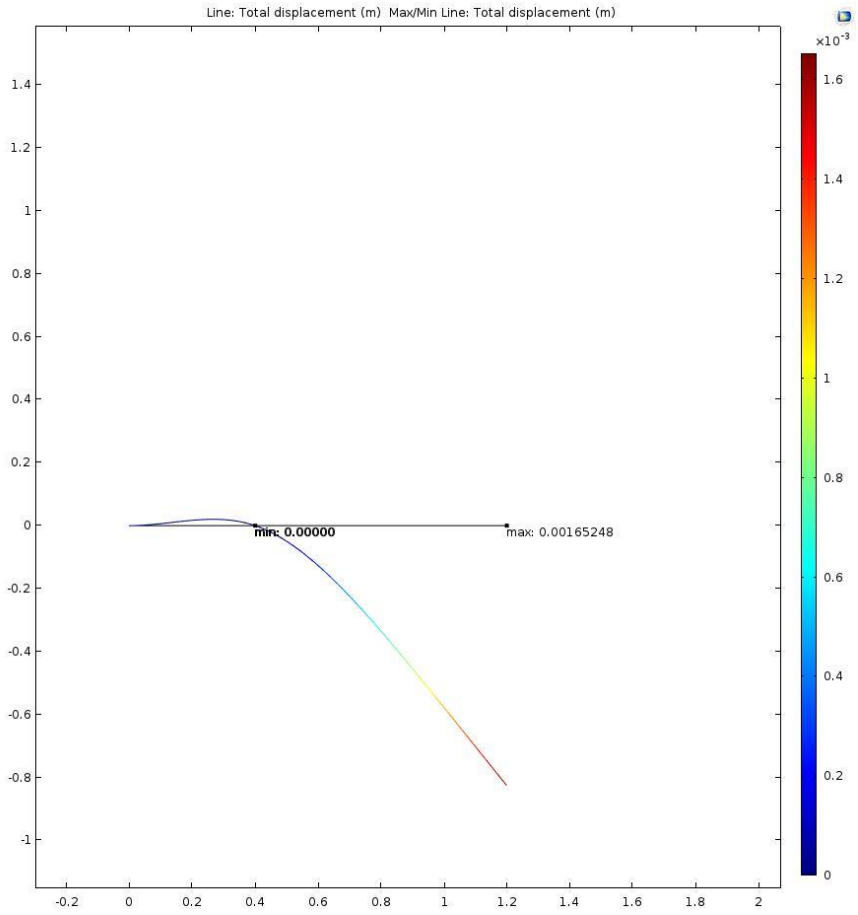


Figure 10: Total displacement of the new beam(Euler-Bernoulli theory)

The results obtained from parametric sweep for Timoshenko beam theory are shown in table 6

Height(m)	**Total displacement(mm)**
0.2	2.4413
0.21	2.1408
0.2157	1.9926
0.2158	**1.9901**
0.22	1.8903
0.23	1.6796

Table 6: Total displacement for different heights (Timoshenko theory)

For the Timoshenko beam theory, we can from the table above that at a height of 0.2158m, the deflection is observed to be less than the one obtained for the original beam.

The deflection of the beam with a height 0.2158m is shown in the below figure:

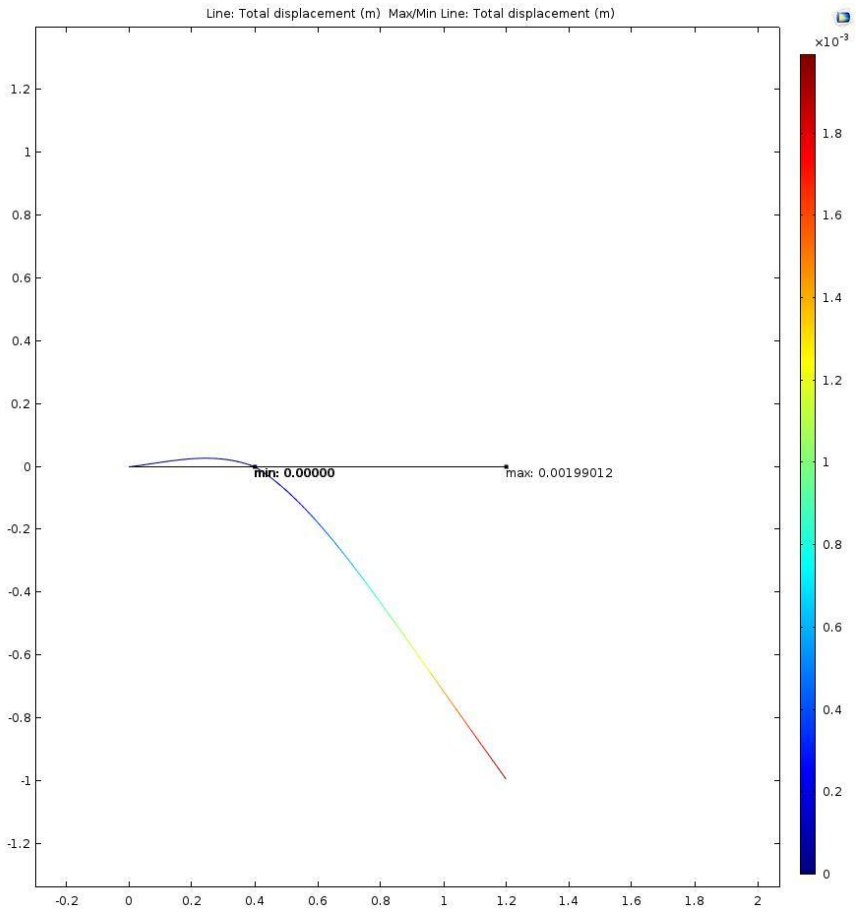


Figure 11: Total displacement of the new beam(Euler-Bernoulli theory)

7.5 Discussions and Conclusions

The required height of the beam is obtained. As the problem is to lower the manufacturing costs, we have to keep in mind the weight of the beam. After choosing a uniform rectangular cross-section we know the volume of the new design of the beam and the original design of the beam. To keep the weight of the beam lower than that of the original design, the height must be less than 0.2m. For the new design with a height of 0.2139m, the deflection is less than that of the original beam (when Euler-Bernoulli theory is considered). When we consider Timoshenko beam theory the deflection is less than the original beam for a height of 0.2158m.

For both this cases, the volume of the beam is more than the original one resulting in increase of weight of the beam. So, our suggestion is to use the tapered rectangular beam instead of the uniform rectangular beam because it has the same deflection and it weighs less than the new beam, unless the cost is more important than the weight of the beam.

8 Discussion and Conclusions

Using finite difference method to solve the problem, the number of elements of the beam has significance on the deflection of the beam. We can observe that the increase in number of elements results in more accurate value of the deflection. The percentage of error is definitely less than two percent for 72 elements or more, but is likely less even earlier. The maximum deflection for FDM with 72 elements was 1.637mm. Using finite element method, the maximum deflection was observed to be 1.668mm for the same number of elements. A convergence graph is plotted for FEM and FDM where one can observe converges for FDM and FEM. The maximum deflection lies at the point of convergence of both the methods between 1.65mm and 1.66m. The computational cost was much larger for the finite element method, because it takes 16 elements whereas finite difference method takes only 5 elements. Using COMSOL to solve the problem, the maximum deflection using Euler-Bernoulli beam theory was observed to be 1.653mm as expected from the convergence graph whereas it was 1.990mm using Timoshenko beam theory. By considering the uniform cross-section, with a height of 0.2139m, the maximum deflection was 1.6525mm. In this case, the weight of the beam is more than that of the original one whereas the deflection is almost the same. Hence the given design is suggested instead of uniform cross-sectional beam.

Names of group members	MM	M	E	L	ML	A
Erik Ising	X					
Anton Karlsson		X				
Raghavendra Machipeddi			X			
Akhil Mora			X			

9 References

1. Broman, G (2003). *Computational Engineering.* Karlskrona: Blekinge Institute of Technology.
2. Ottosen, N. & Petersson, H. (1992). *Introduction to the finite element method.* Hall International Ldt.

Appendix

FDM.m

```
tic
clc;
clear;
close all;

h0=0.3;
he=0.2;
L=1.2;
w=0.1;
qm=200e3;
step=576;
h=L/(step-1);    %stepsize
x=0;             %position
E=69*10^9;
RowSupport=round(0.4/h)+1;

A = zeros(step);
C = zeros(step,1);

for row=2:step
    x=x+h;

    differenteq = [(E*((w*(h0-he*((x-h)/L))^3)/12))/h^4,(-2*((E*((w*(h0-
he*((x-h)/L))^3)/12))+(E*((w*(h0-he*((x)/L))^3)/12))))/h^4,(((E*((w*(h0-
he*((x-h)/L))^3)/12))+(4*(E*((w*(h0-he*((x)/L))^3)/12)))+(E*((w*(h0-
he*((x+h)/L))^3)/12)))/h^4),((-2*((E*((w*(h0-
he*((x)/L))^3)/12))+(E*((w*(h0-he*((x+h)/L))^3)/12))))/h^4),(E*((w*(h0-
he*((x+h)/L))^3)/12))/h^4];

    for column=2:step
        %Force in each position
        if x<=(L/2)
            C(row)=0;
        else
            C(row)=qm*sin(2*pi*(x/L));
        end

        %Support
        if row==RowSupport
            A(row,row)=0;
        end

        %Not for last 2 points.
        if row<=step-2
            for colposadd = -2:2
                if row+colposadd >= column
                A(row,row+colposadd)=differenteq(3+colposadd);
                end
            end
        end
```

```
        %Second last point
        if row==step-1
            A(row,row-2)=differenteq(1);
            A(row,row-1)=differenteq(2);
            A(row,row)=differenteq(3)-differenteq(5);
            A(row,row+1)=differenteq(4)+(2*differenteq(5));
        end

        %Last point
        if row==step

A(row,row)=differenteq(3)+(2*differenteq(4))+(4*differenteq(5));
            A(row,row-1)=differenteq(2)-differenteq(4)-(4*differenteq(5));
            A(row,row-2)=differenteq(1)+differenteq(5);
        end
    end
end
A(1,1)=1; %Boundary condition in first point
Disp=A\C; %Displacement

hold on
grid minor
x=0:h:L;
plot(x,Disp)
title('Solution using FDM')
ylabel('Position [m]')
xlabel('Displacement [m]')
toc
```

FEM.m

```
tic
clc
clear
close all

L = 1.2;
E = 69e9;
h0 = 0.3;
he = 0.2;
w=0.1;
qm = 200e3;

%works for 3*n elements
elements = 3*4 ;

syms x
Lele = L/elements;

%Shape functions
N = [1-((3*x^2)/(Lele^2))+((2*x^3)/(Lele^3)),x*(1-
((2*x)/(Lele))+((x^2)/(Lele^2))),((x^2)/(Lele^2))*(3-
((2*x)/(Lele))),((x^2)/(Lele))*((x/Lele)-1)];
B = diff(N,x,2);
```

```
Kmatrix = zeros(2*elements+2);
Kumatrix = zeros(2*elements+2,1);

for element=1:elements
    position = (2*element-2)*L/(2*elements);
    I = (w*(h0-he*(position/L))^3)/12;

    %Force
    if position >= L/2
        q = -qm*sin(2*pi*position/L);
    else
        q=0;
    end

   for row=1:4
       for column=1:4
           Krc = E*I*int(B(row)*B(column), x, 0, Lele);
           Kmatrix((2*element-2)+row,(2*element-2)+column) =
Kmatrix((2*element-2)+row,(2*element-2)+column) + Krc;
       end
       Ku = int(N(row)*q, x, 0, Lele);
       Kumatrix((2*element-2)+row,1) = Kumatrix((2*element-2)+row,1) + Ku;
   end
       element
end

%Boundarys first point
Kmatrix(1,1) = 10^15*Kmatrix(1,1);
Kumatrix(1) = 0;
Kmatrix(2,2) = 10^15*Kmatrix(2,2);
Kumatrix(2) = 0;

%Boundaries for support
Kmatrix(round(1+(elements/3)*2),round(1+(elements/3)*2)) =
10^15*Kmatrix(round(1+(elements/3)*2),round(1+(elements/3)*2));
Kumatrix(round(1+(elements/3)*2)) = 0;

Disp = Kmatrix\Kumatrix;
hold on
grid minor
x=0:L/(elements):L;
plot(x,-Disp(1:2:end))
title('Solution using FEM')
ylabel('Position [m]')
xlabel('Displacement [m]')
% figure()
% plot(x,Kumatrix(1:2:end))
toc
```

Convergence.m

```
clc
clear
close all

elements = [18,36,72,144,288,576];
dispFEM = [1.694,1.679,1.668,1.661,1.657,1.656];
dispFDM = [1.571,1.619,1.637,1.646,1.650,1.652];

hold on
grid minor
plot(elements,dispFEM)
plot(elements,dispFDM)
axis tight
legend('FEM','FDM')
title('Convergence')
ylabel('Position [mm]')
xlabel('Number of elements')
```