AE 458 Term project – FEA

Shane Chambry – Jimmie Rush III – Joshua Wheeler

Abstract:

Throughout the semester, structural problems have been solved in the perspective of finite element analysis. Each beam, axial member, torsional member, and shear panel is broken up into smaller pieces to be analyzed separately. This project comes in two parts. Firstly, the creation of original FEA software is to be accomplished. Using Matlab, a general code will be written to solve any manner of simple beam problems. Secondly, Comsol will be used to analyze a full wing under several loading conditions, some of which combine bending and torsion.

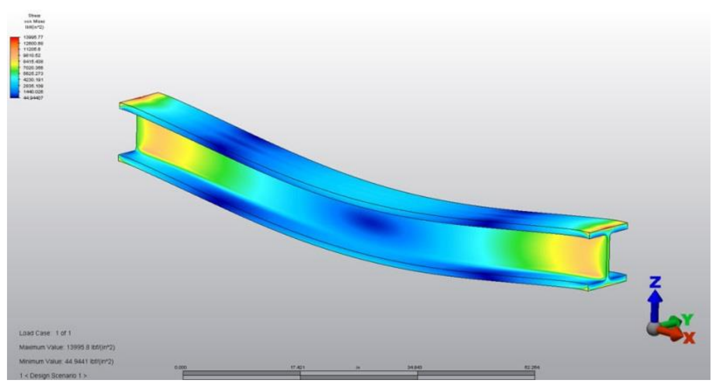
Intro:

To truly gain an understanding of finite element analysis, a small program will be written to solve beam problems. The code should be general enough to solve a wide array of loading cases and changes in geometry or material properties. To do this, each beam will be divided into sections by the user. There should be a section for each load and whenever there is a change in beam geometry or composition. In addition, boundary conditions will be input by the user.

With the input information, the program can solve for nodal displacements for the entire beam. In addition, it can produce a plot for displacement, bending moment, and shear force as a function of the axial position. Examples of the finished program at work are included.

After grasping the basics of FEA, a more complex problem will be approached - a wing, to be specific. Comsol, a fully developed finite element program, is used to solve this problem. Using the provided schematics for the wing, a finite element model of the wing is produced in Comsol. With the completed model, the response can be computed for any loading case. For this project, the model will be subjected vertical loads at different locations along the free end. With this data, the shear center can be located. Also, the wing will be tested under a distributed “drag,” load on the leading edge. This case will be tested with a variety of mesh densities to observe its effect on the results.

Theory:



Finite element analysis evolved from the need to solve complex structural analysis problems. It is a numerical method that involves breaking up a structure into elements, and computing its response to a load as if each element was connected to its neighbors with a spring of certain stiffness. Each element has nodes at its boundaries. FEA finds the displacement of each node, depending on the force applied to the structure.

Beam element stiffness

$$[k^{(e)}] = \left(\frac{EI}{L^3}\right)^{(e)} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}$$

The first step is to discretize the structure into elements. Generally, more elements correlate to a more accurate solution. The second step is to determine the stiffness of each element. This is characterized by the geometry of the element and the material it is composed of. These stiffness values are then assembled into a global stiffness matrix. Each location in the global stiffness matrix corresponds to a node that lies between two elements. In this manner, the displacements of each node can be solved for simultaneously. The response of the structure is proportional to the applied force by its stiffness. Therefore, after assembly is finished, the displacements of the nodes can be found by solving $[k]\{q\} = \{f\}$, where [k] is the global stiffness matrix, {q} is a vector of nodal displacements, and {f} is a vector of nodal forces (or equivalent distributed loads). The {q} vector may also include the slope at a node if a beam is being analyzed.

The last step is referred to as post-processing. This step includes calculating the stresses and strains from the displacement information and failure analysis. Upon reaching this step, the goal of the preceding calculations has been achieved. Several questions about the structure can be answered at this point.

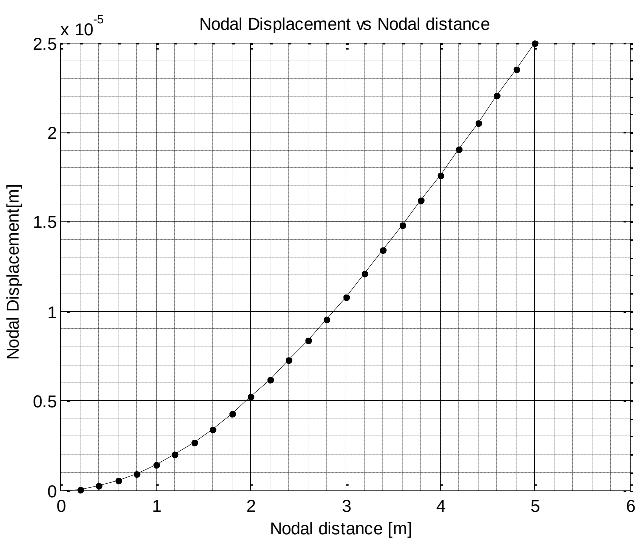
"How much does the structure deform?"
"How much stress is it under? "
"Will the structure fail under this loading?"

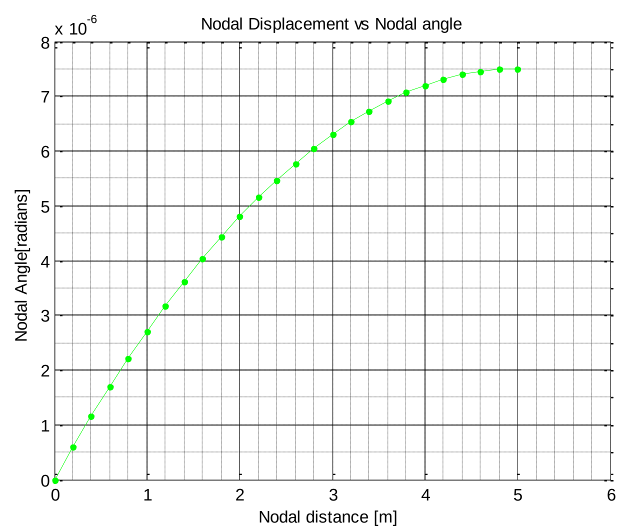
The last question listed is inarguably the most important question, even more so when the structure is a building, a car, or an airplane. The lives of passengers and inhabitants depend on a structure's ability to withstand a load without failure.

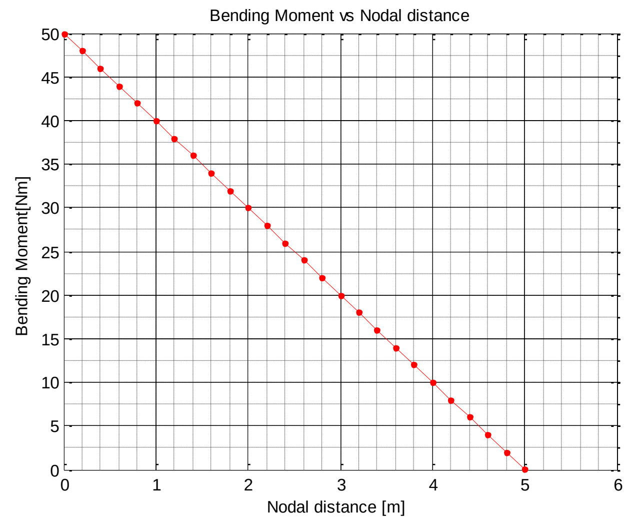
Results:

As stated above, the first part of this project involved writing some Matlab code that can be used to solve beam problems with a finite element analysis approach. The code itself is included in Appendix I. The code takes the information about the beam, including how it is to be discretized, the applied forces, and the boundary conditions, and solves for the shear force, bending moment, and displacement. The following plots show the results from two different beam problems.

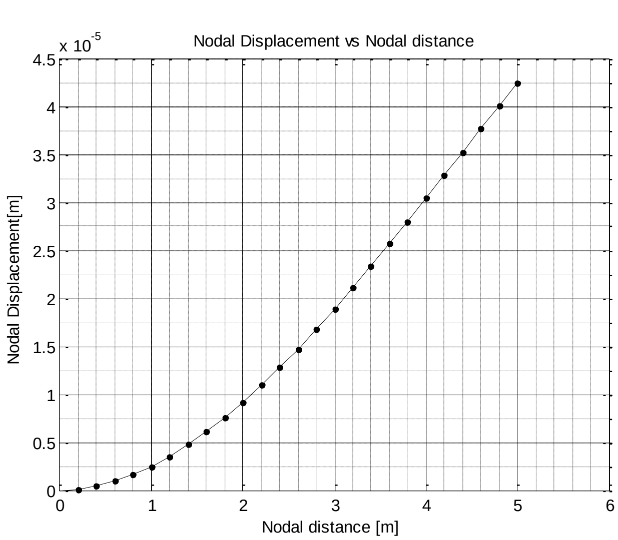
For the first test of the code, a simple cantilevered beam is used. It is fixed at one end, made of steel, has a square cross section of .1m x .1m, and is subjected to a transverse loading of 10 Newtons at the free end.

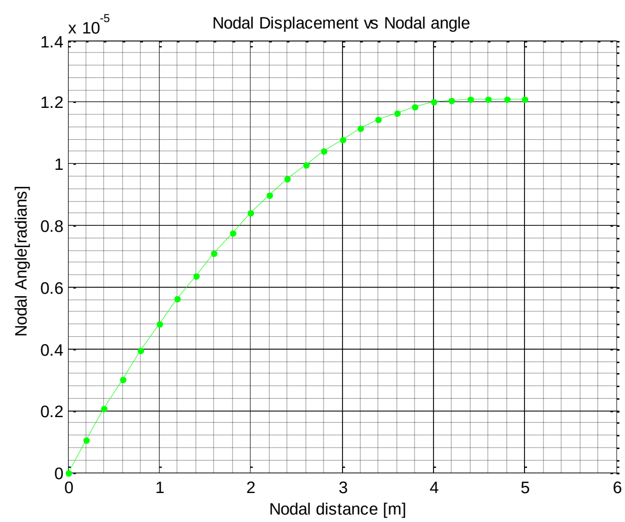


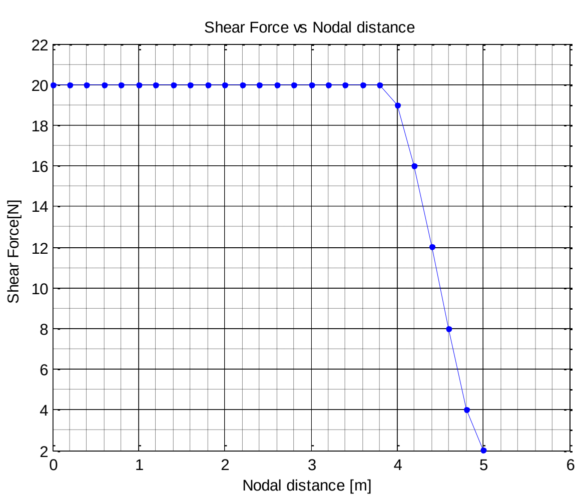


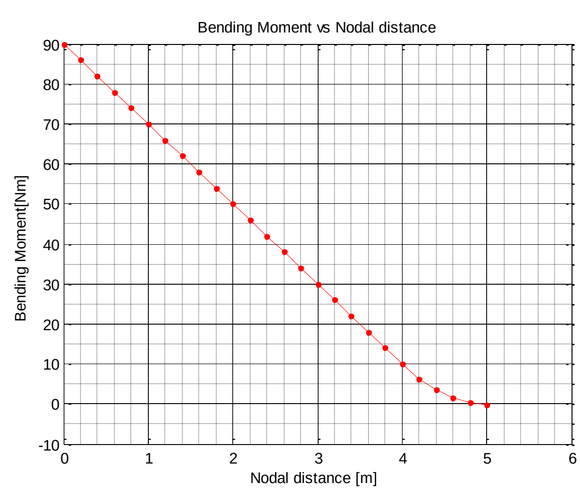


The second problem used to test the code has, instead of a concentrated load at the end, a distributed load on the last of five sections. Each section has the same length of 1m. The distributed load is 20 N/m, and the last section also has double the moment of inertia of the rest of the beam.



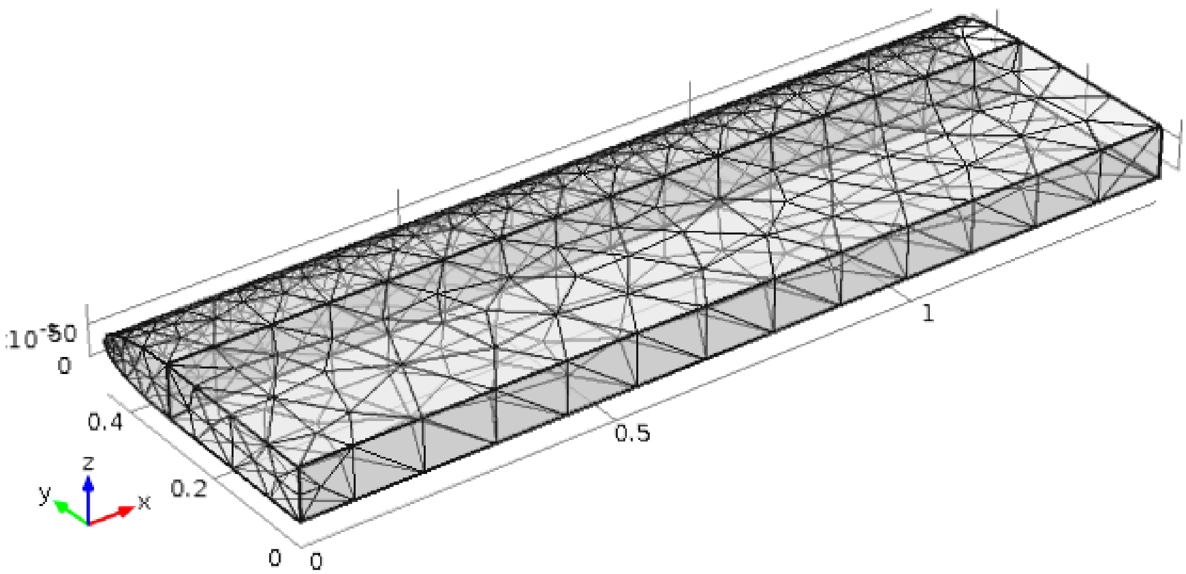


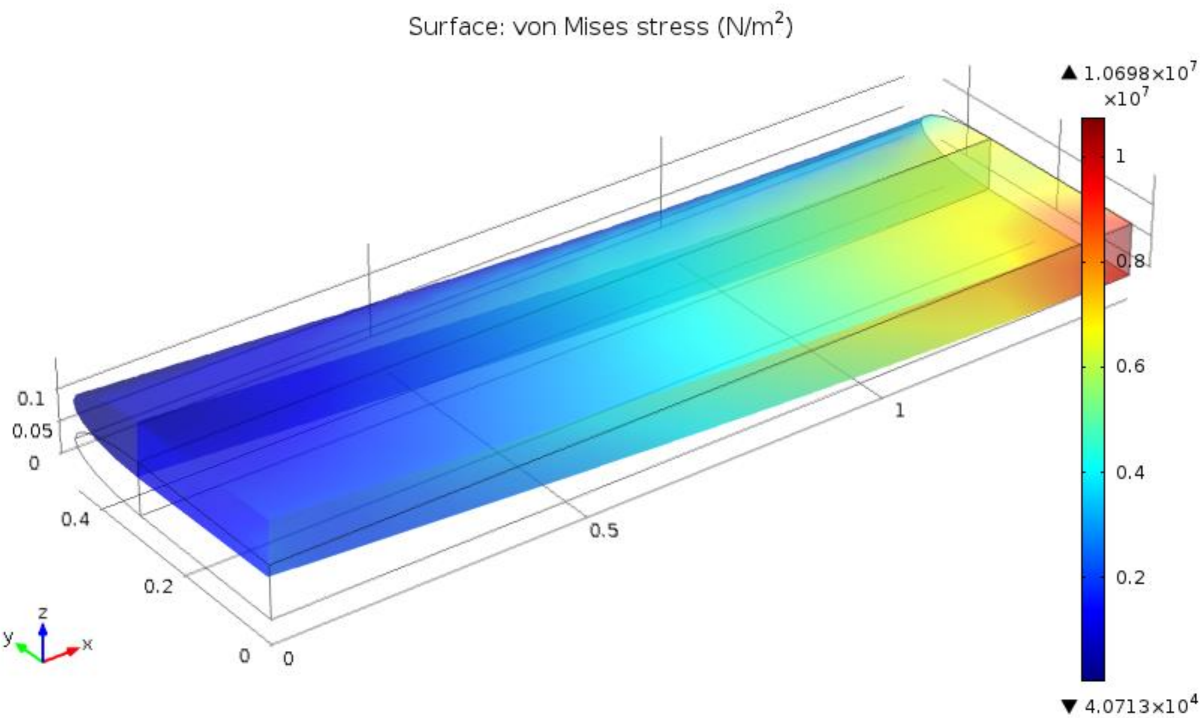


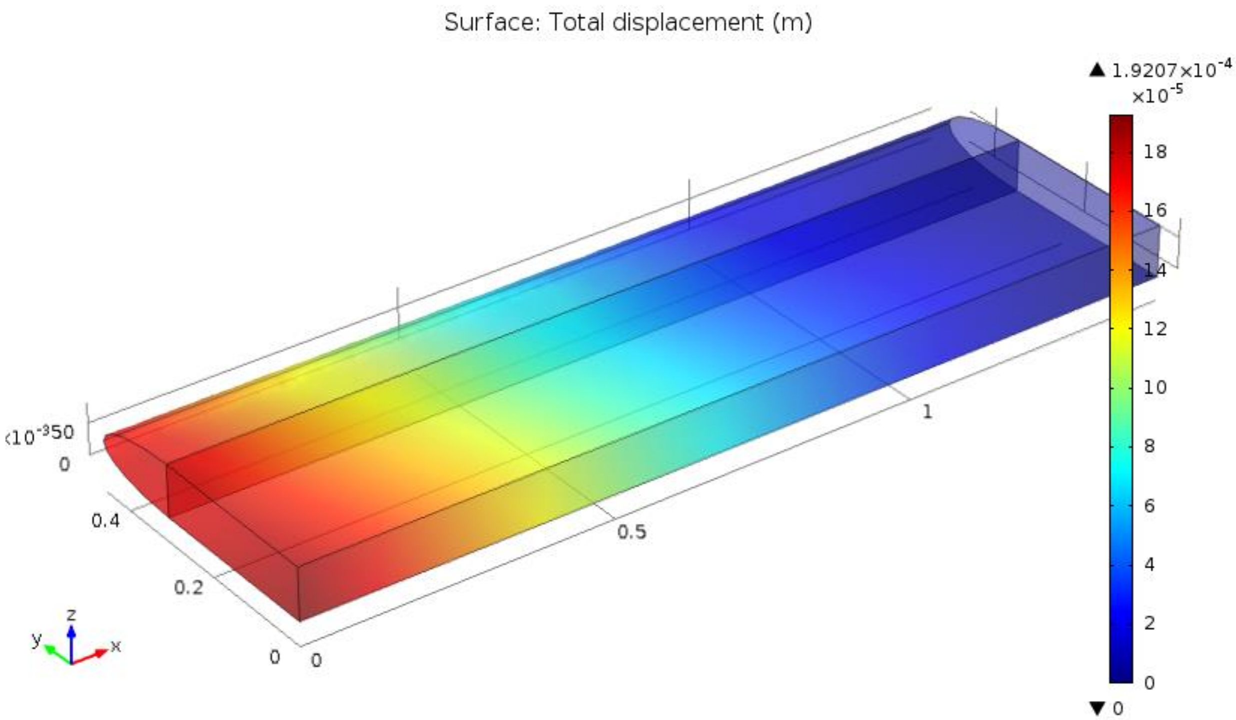


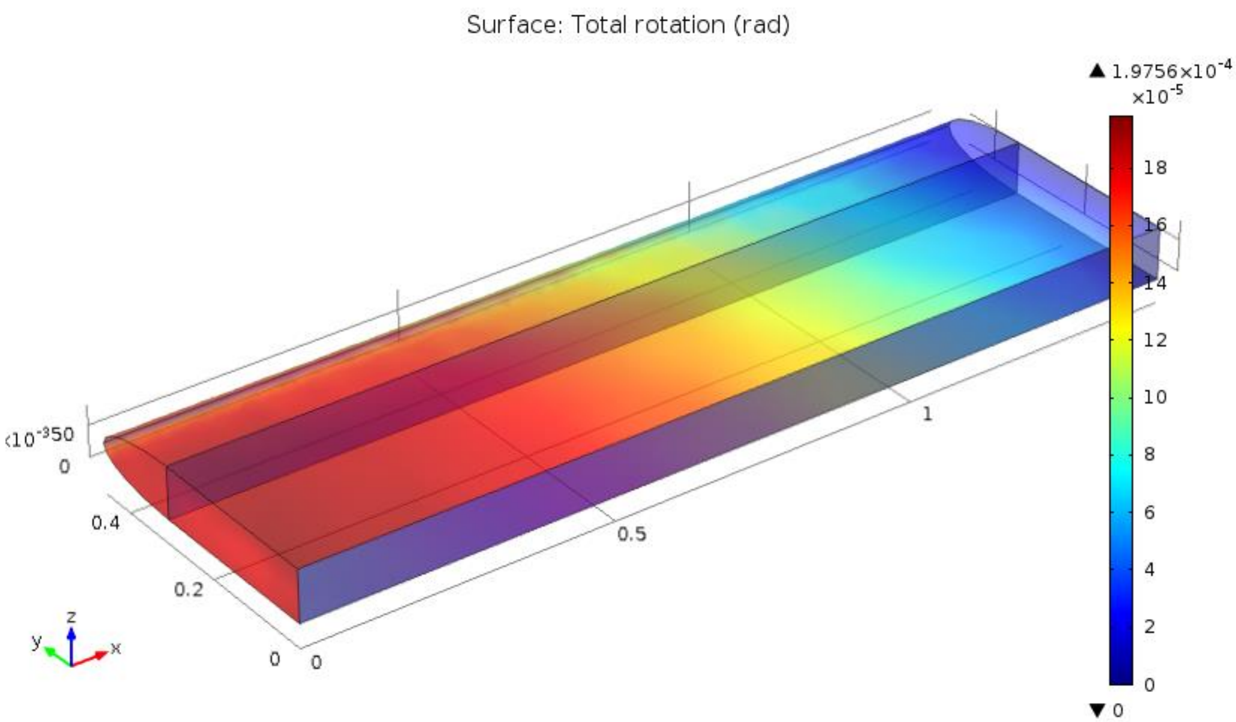
The second portion of the project is, of course, using Comsol to solve for the response of a wing under various loading conditions. The numerous figures show displacement, rotation, and von misses stress for each of loading cases. In each case, the wing is subjected to a 50lb vertical load at one of seven evenly spaced points on the end of the wing.

Point 1 Fine



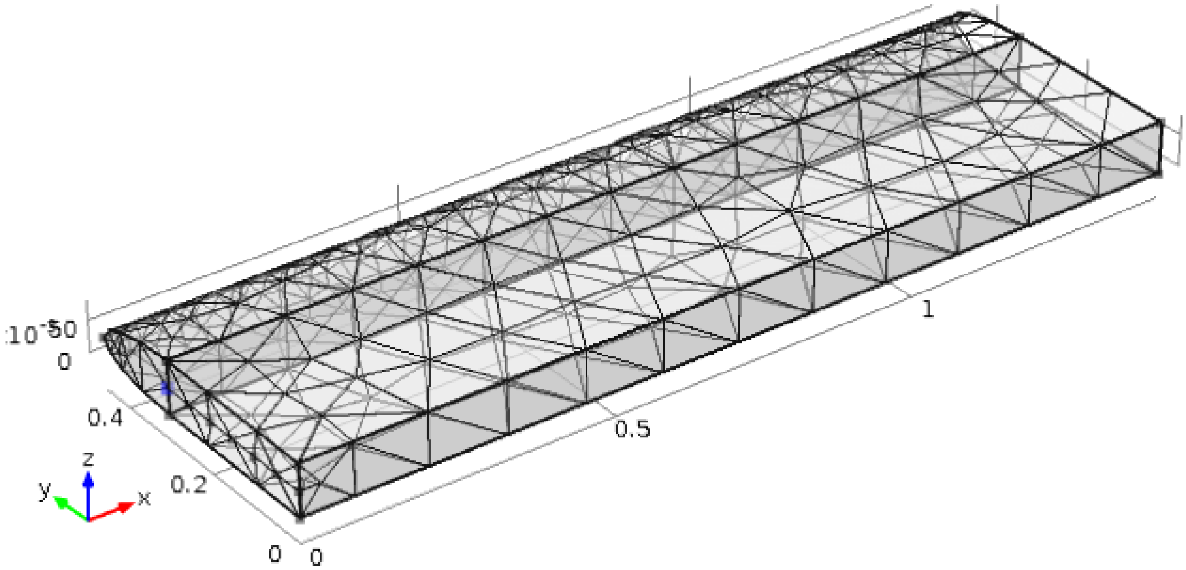


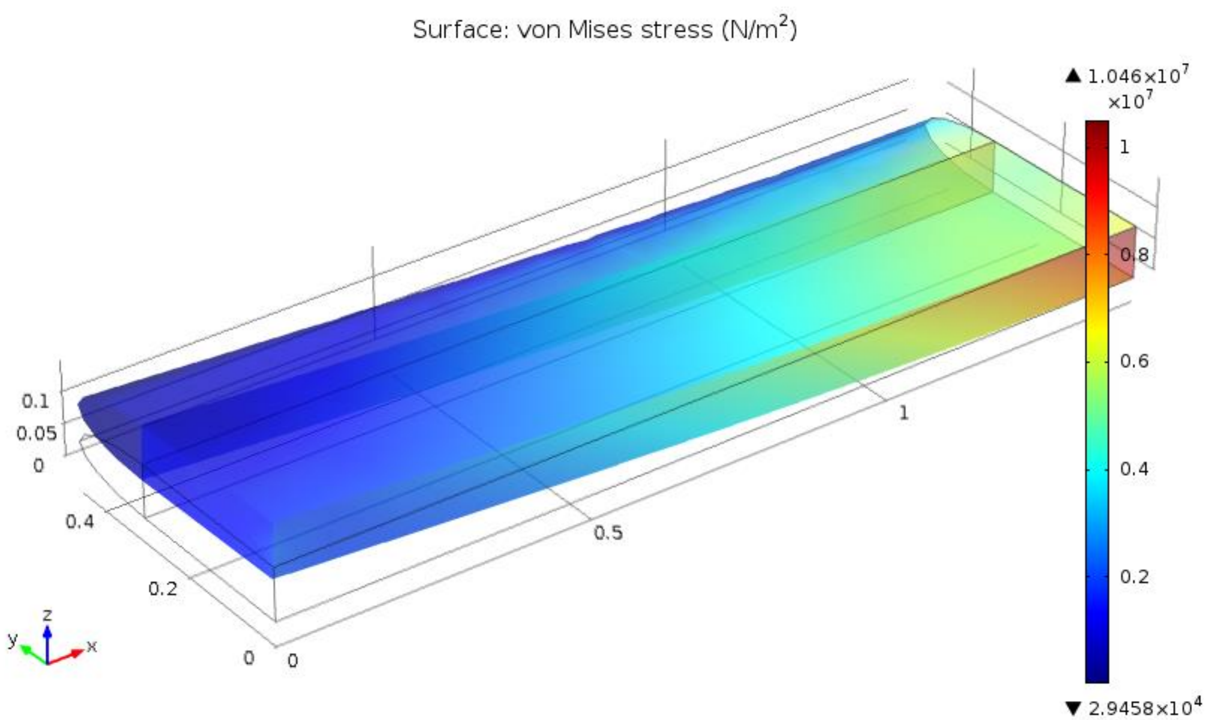


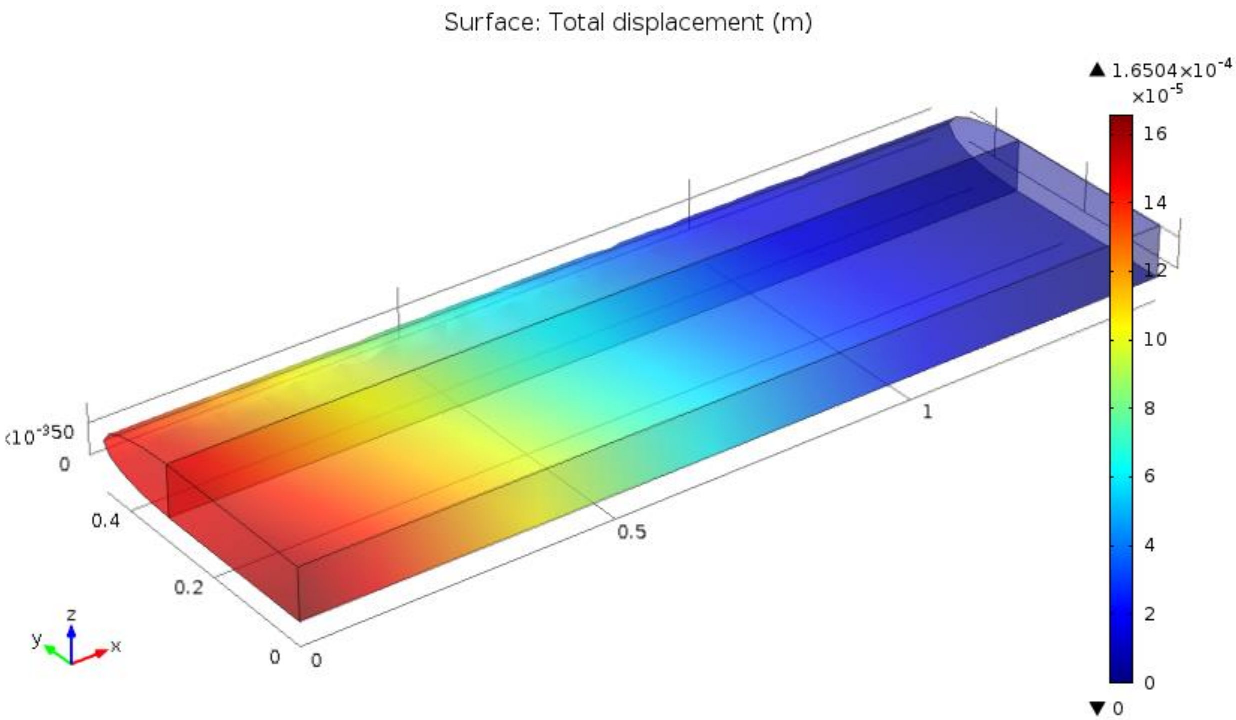


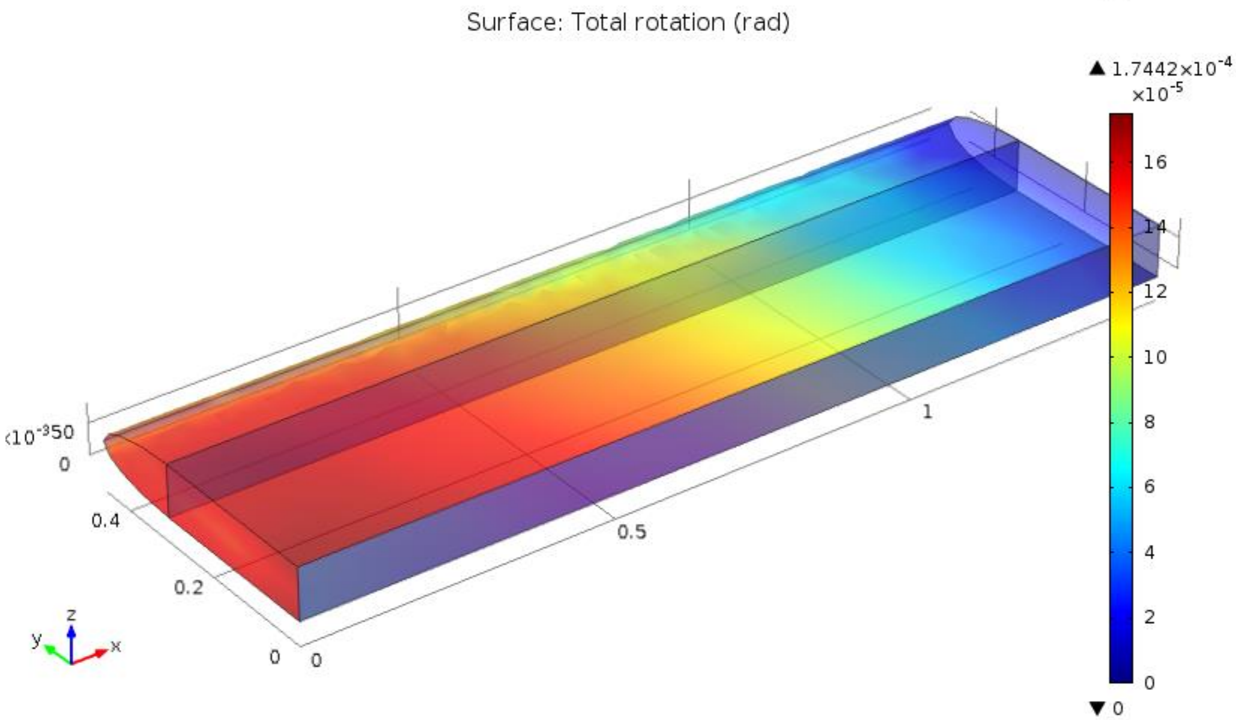
Total displacement (m): 3.84197e-5, 3.75196e-5
Strain tensor (global), xx component (1): -5.97883e-6, 6.39622e-6
Strain tensor (global), xy component (1): -1.19307e-6, 1.3938e-6

Point 1 normal



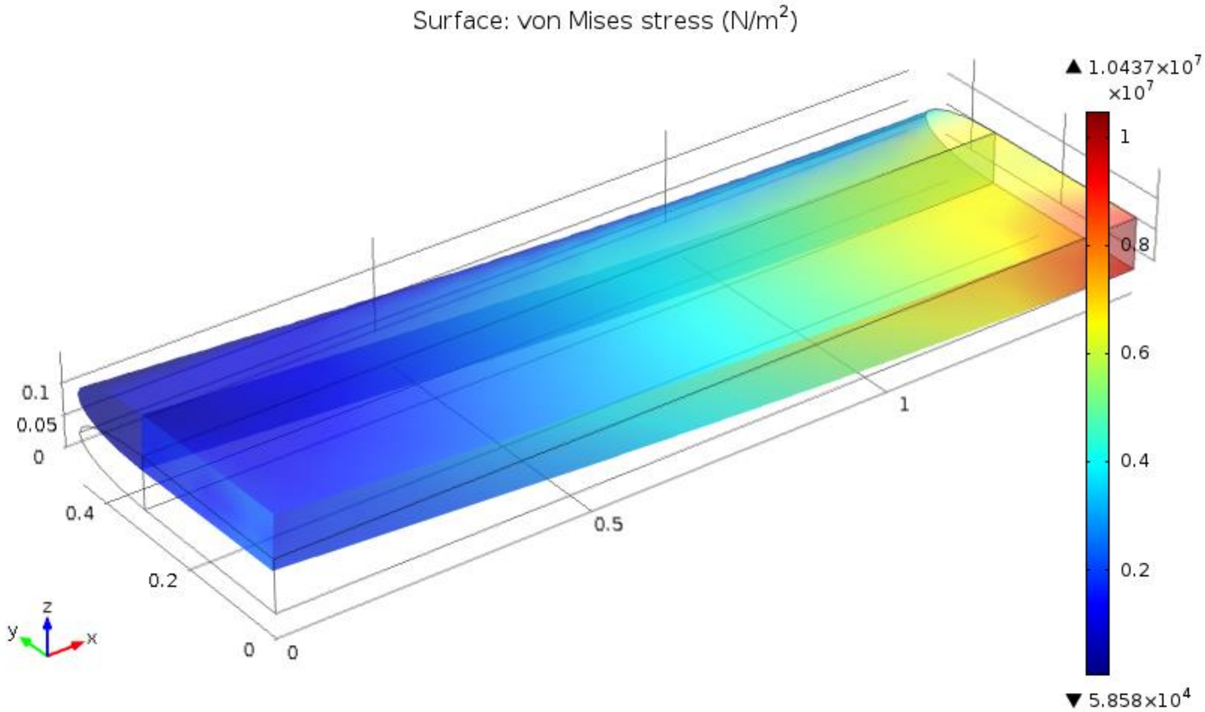




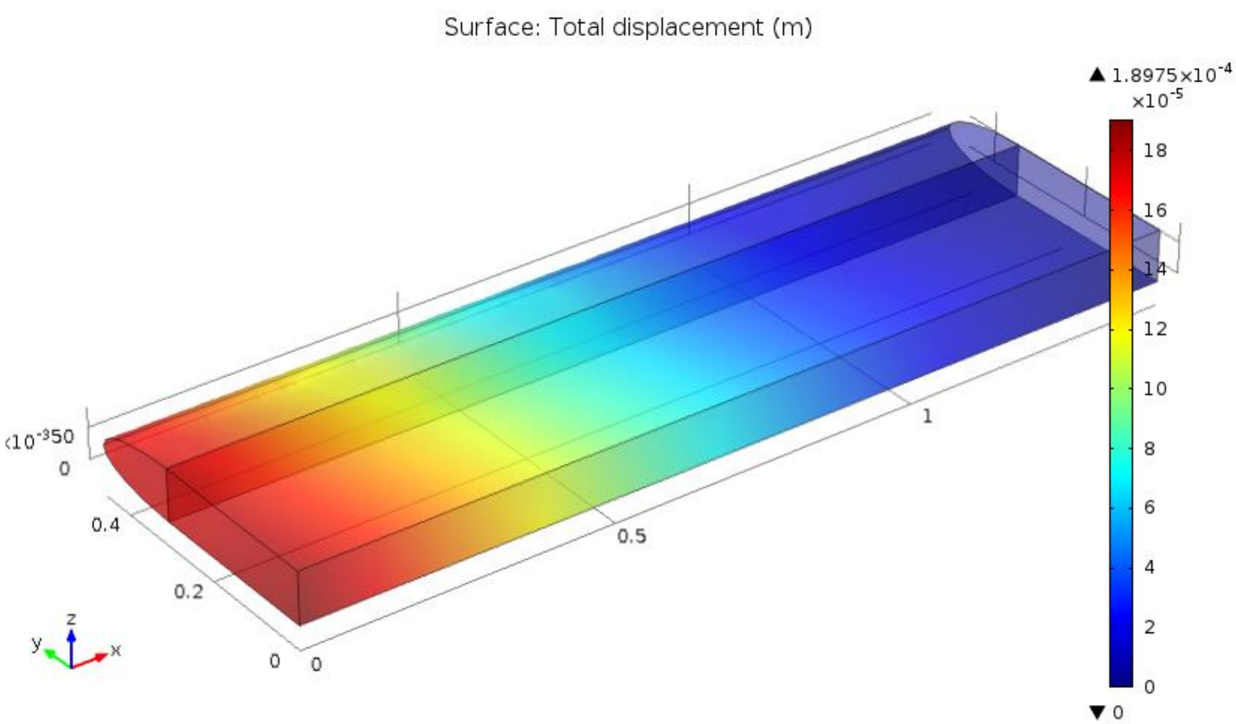


Total displacement (m): 3.19711e-5, 3.06554e-5
Strain tensor (global), xx component (1): -5.76127e-6, 5.46903e-6
Strain tensor (global), xy component (1): -9.67824e-7, 1.17867e-6

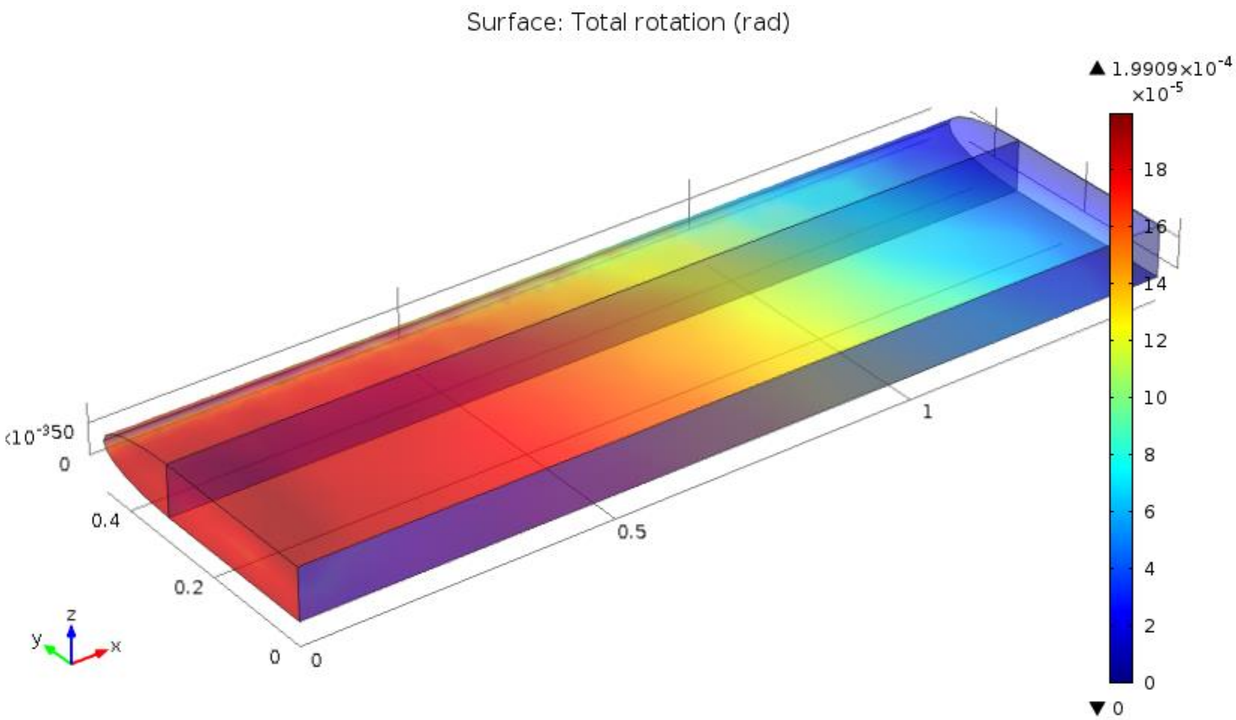
Point 2 Fine



Surface: von Mises stress (N/m^2)
▲ 1.0437×10^7
×10^7
1
0.8
0.6
0.4
0.2
▼ 5.858×10^4

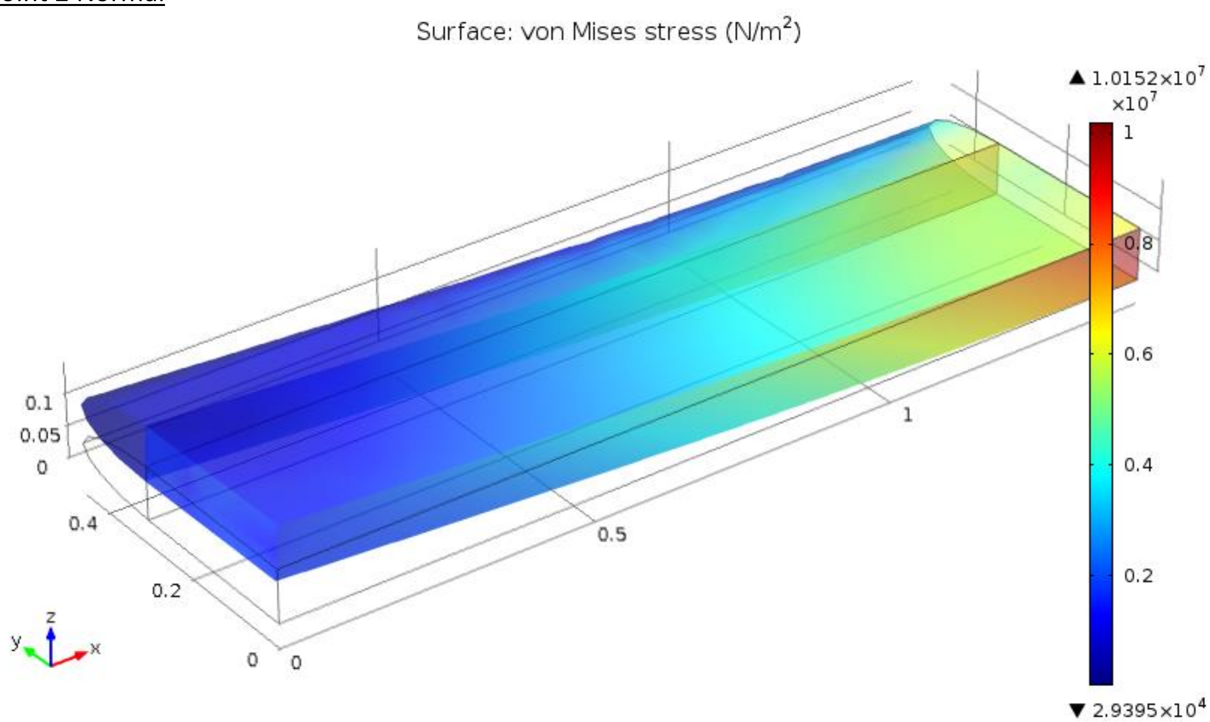


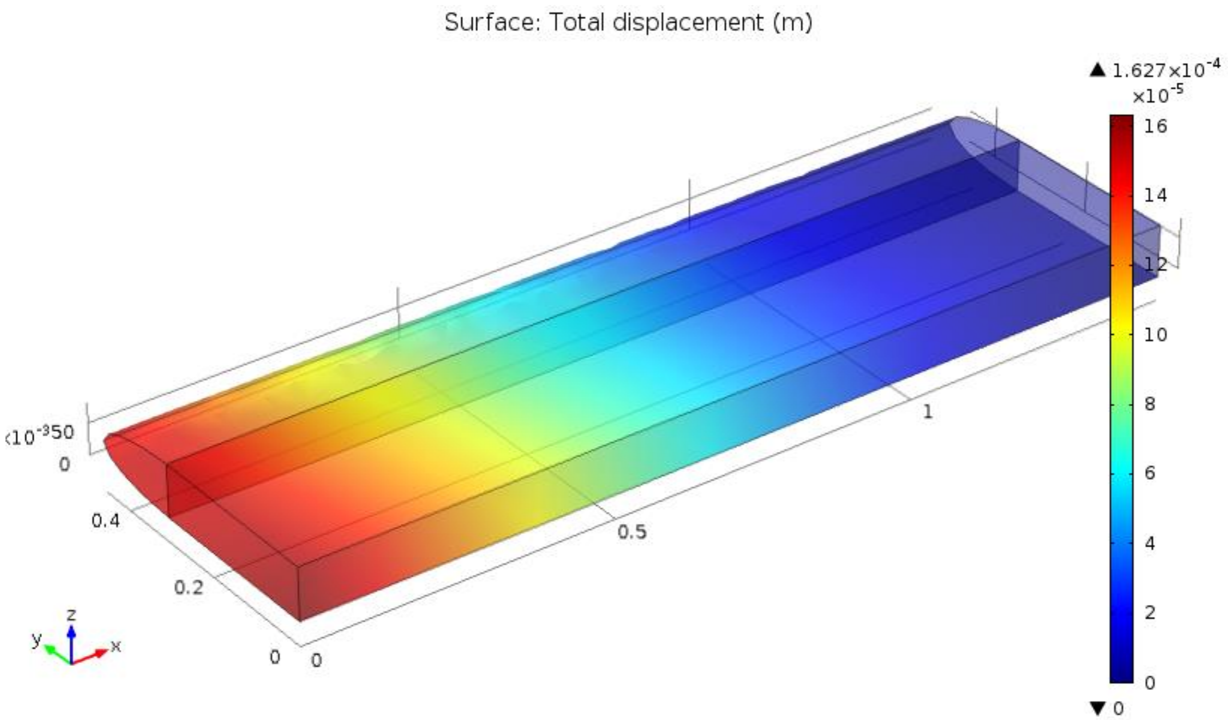
Surface: Total displacement (m)
▲ 1.8975×10^{-4}
×10^{-5}
18
16
14
12
10
8
6
4
2
0
▼ 0

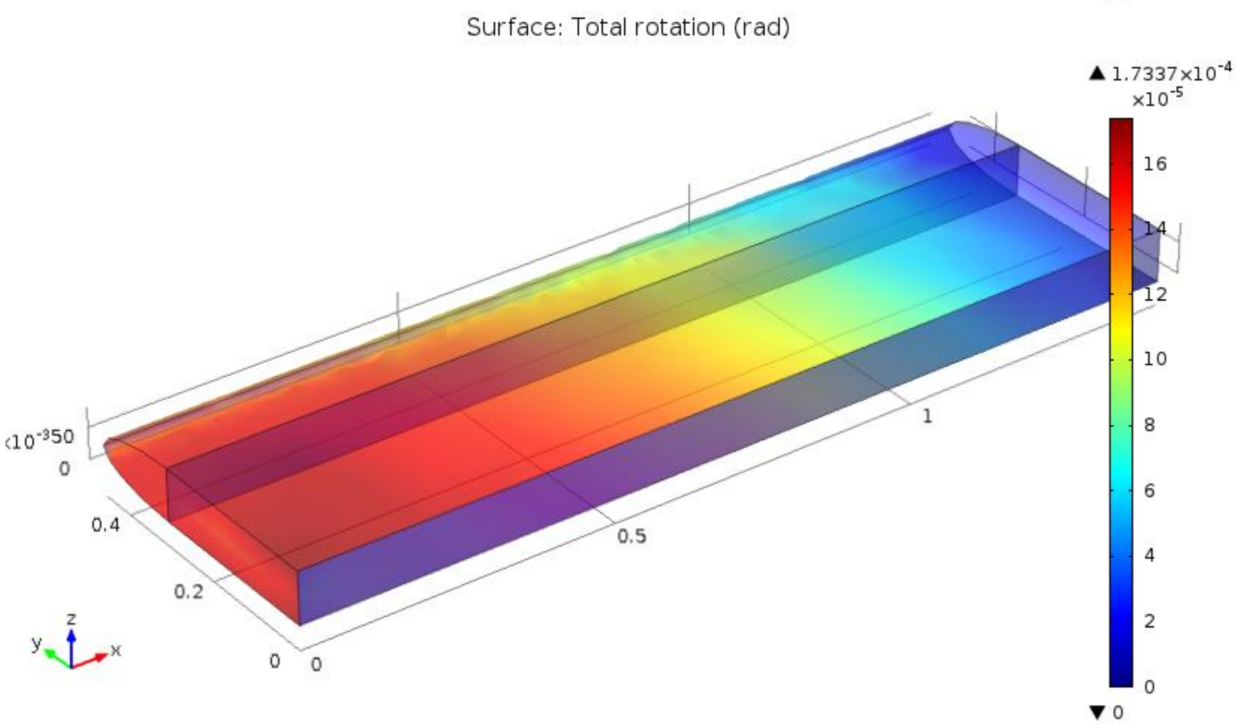


Total displacement (m): 3.82577e-5, 3.74741e-5
Strain tensor (global), xx component (1): -5.97544e-6, 6.3953e-6
Strain tensor (global), xy component (1): -8.89007e-7, 1.089e-6

<u>Point 2 Normal</u>

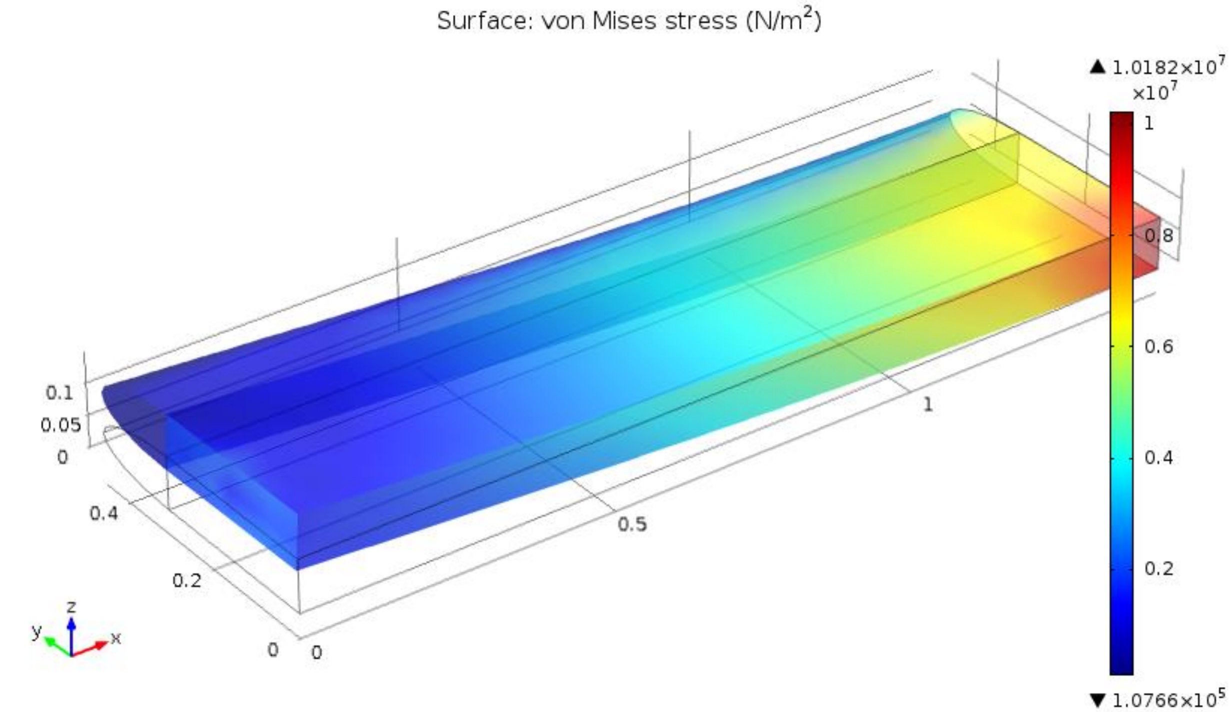


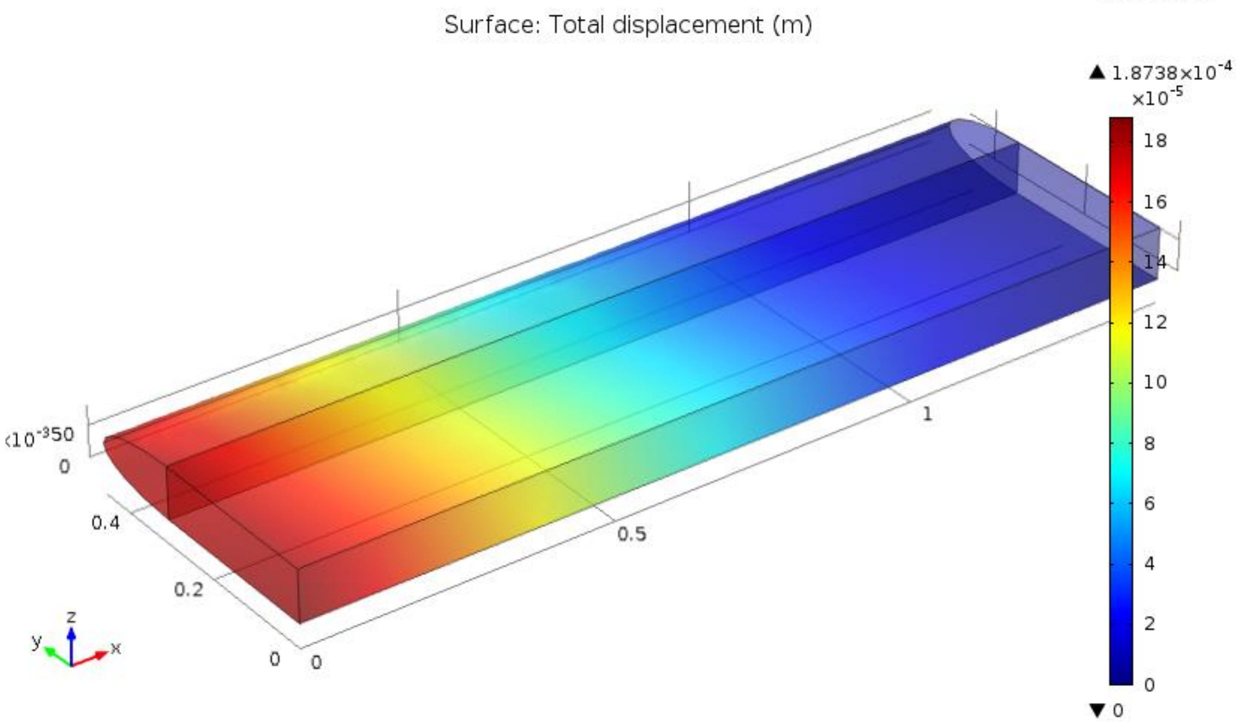


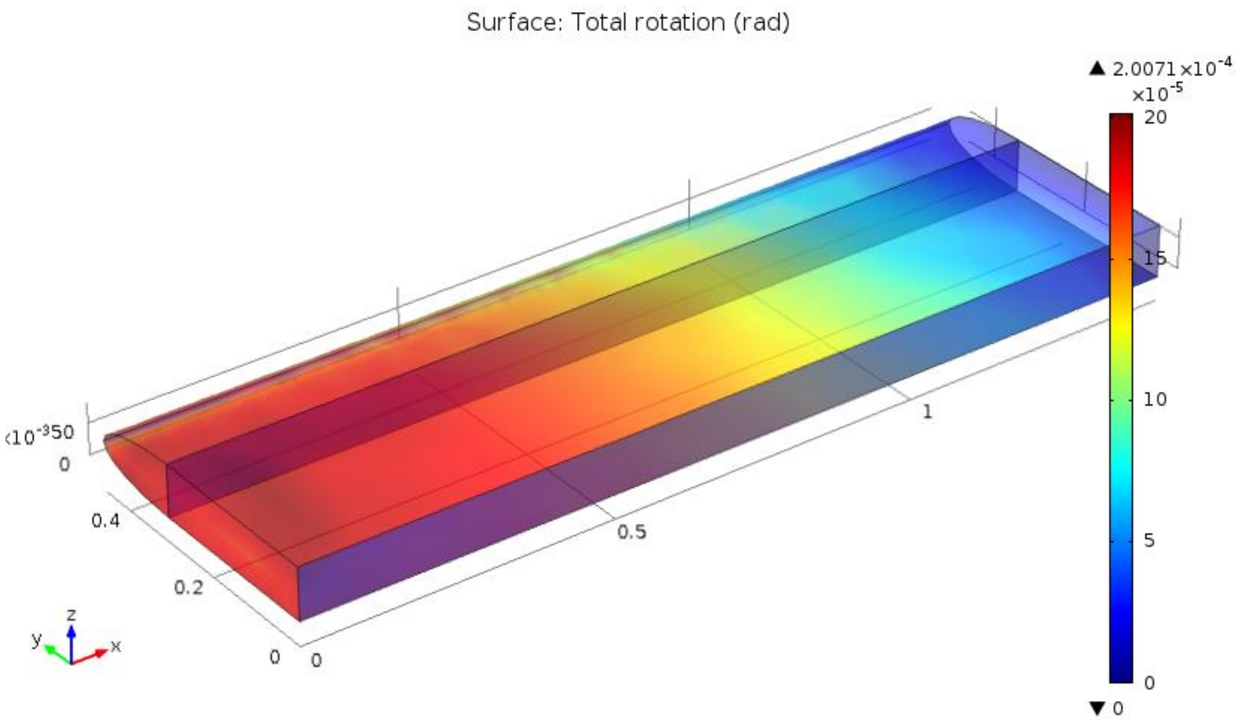


Total displacement (m): 3.17938e-5, 3.0524e-5
Strain tensor (global), xx component (1): -5.77188e-6, 5.47604e-6
Strain tensor (global), xy component (1): -6.68779e-7, 8.79419e-7

Point 3 Fine

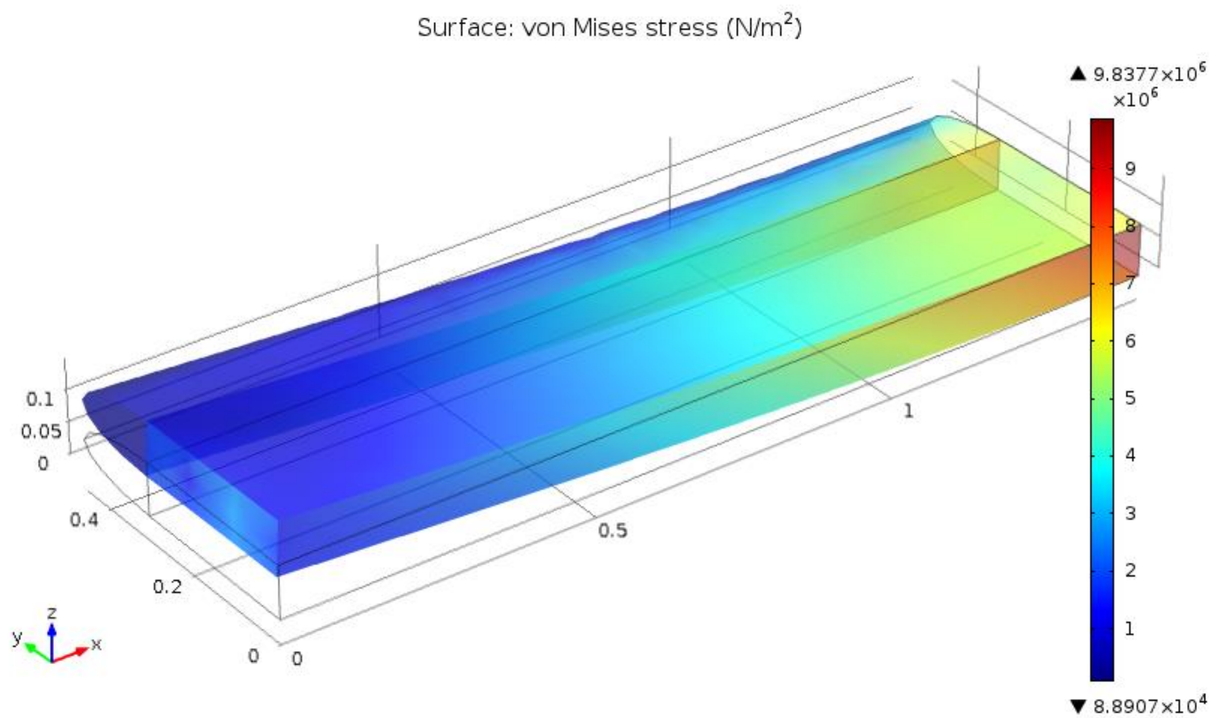

Surface: von Mises stress (N/m²)
▲ 1.0182×10⁷
×10⁷
1
0.8
0.6
0.4
0.2
▼ 1.0766×10⁵
0.1
0.05
0
0.4
0.2
0
0
0.5
1
y
z
x

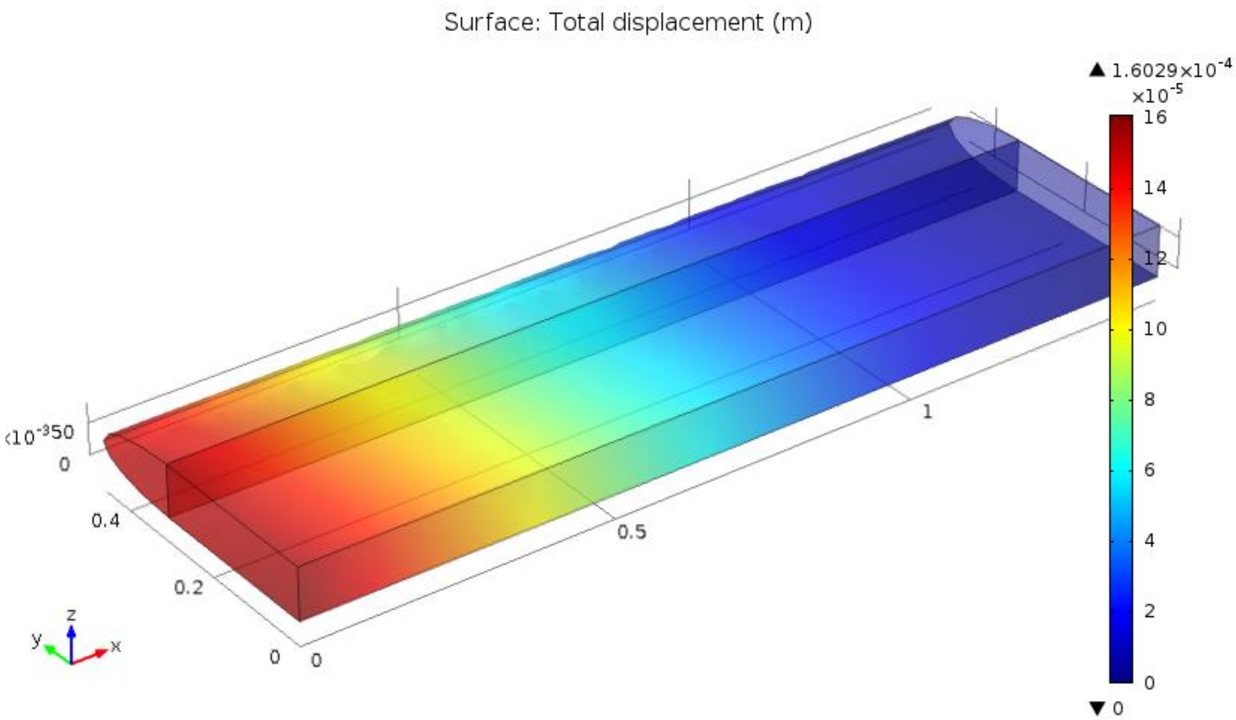

Surface: Total displacement (m)
▲ 1.8738×10⁻⁴
×10⁻⁵
18
16
14
12
10
8
6
4
2
0
▼ 0
0.4
0.2
0
0
0.5
1
y
z
x

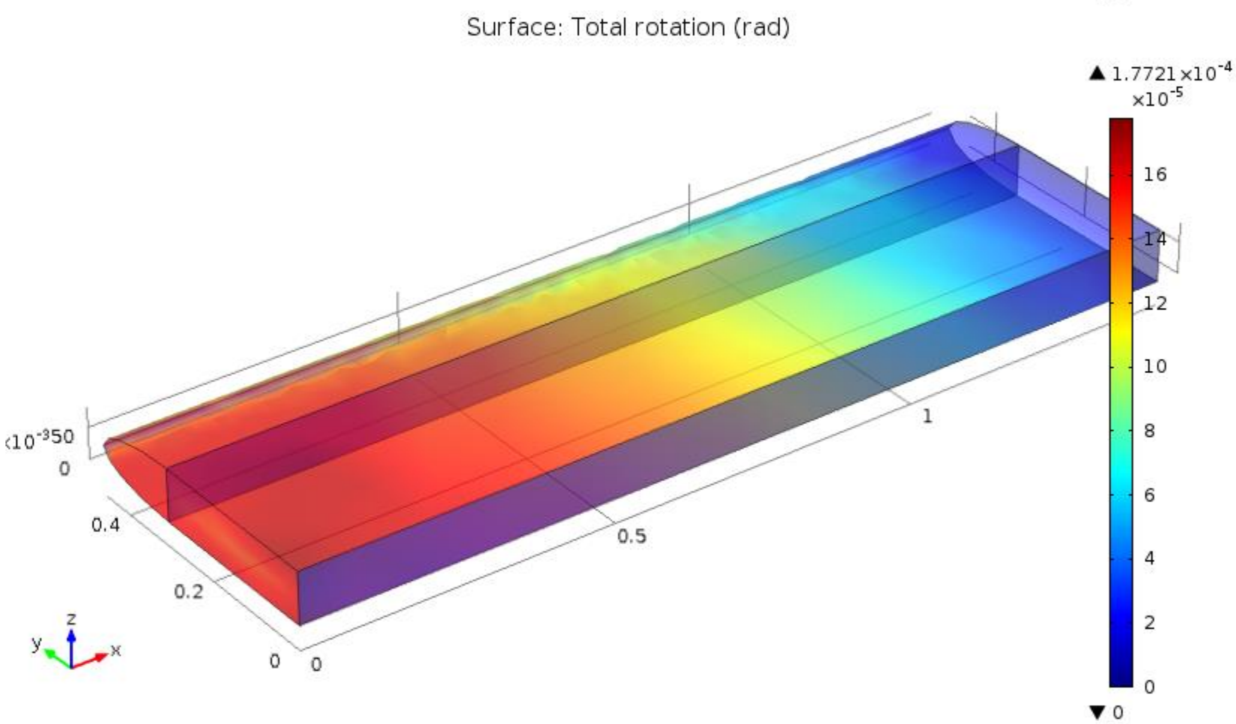


Total displacement (m): 3.17938e-5, 3.0524e-5
Strain tensor (global), xx component (1): -5.77188e-6, 5.47604e-6
Strain tensor (global), xy component (1): -6.68779e-7, 8.79419e-7

Point 3 normal

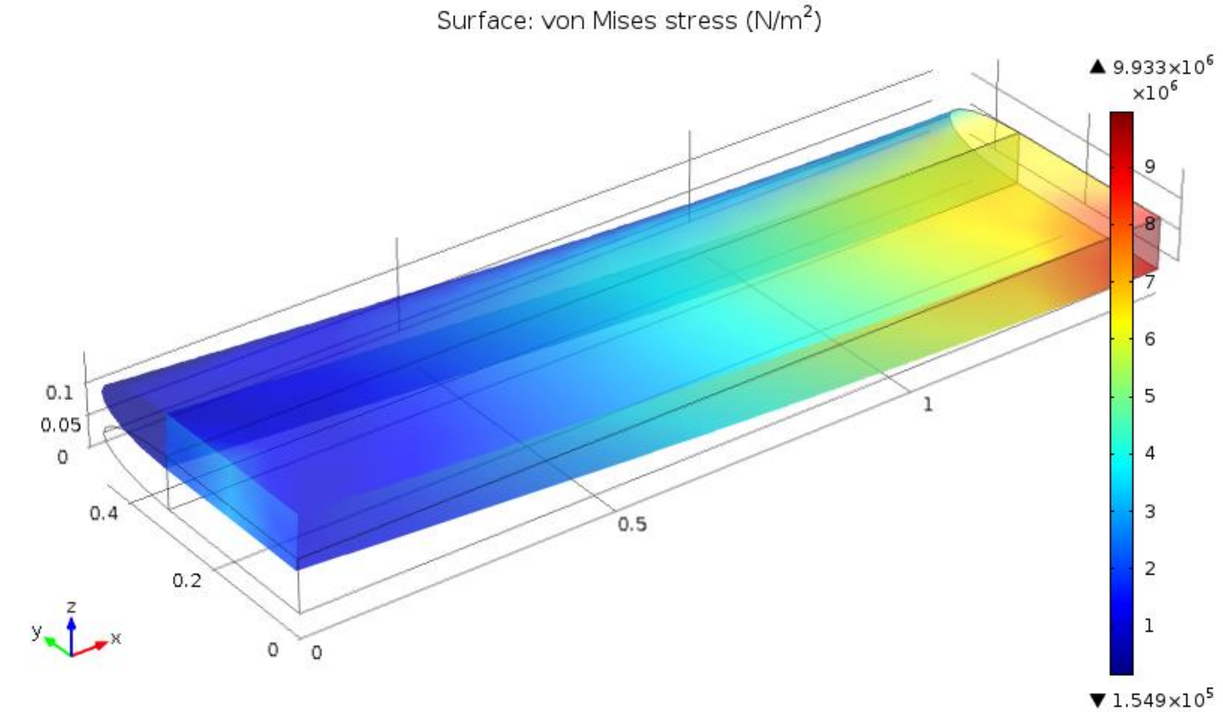


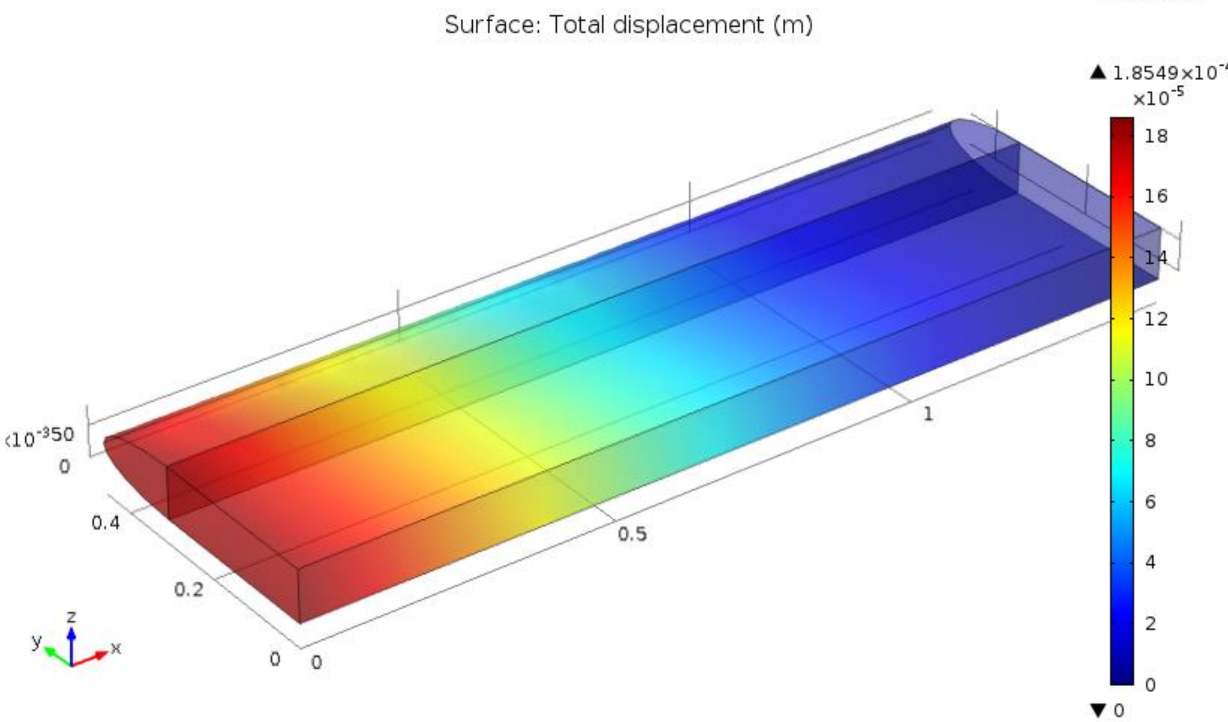


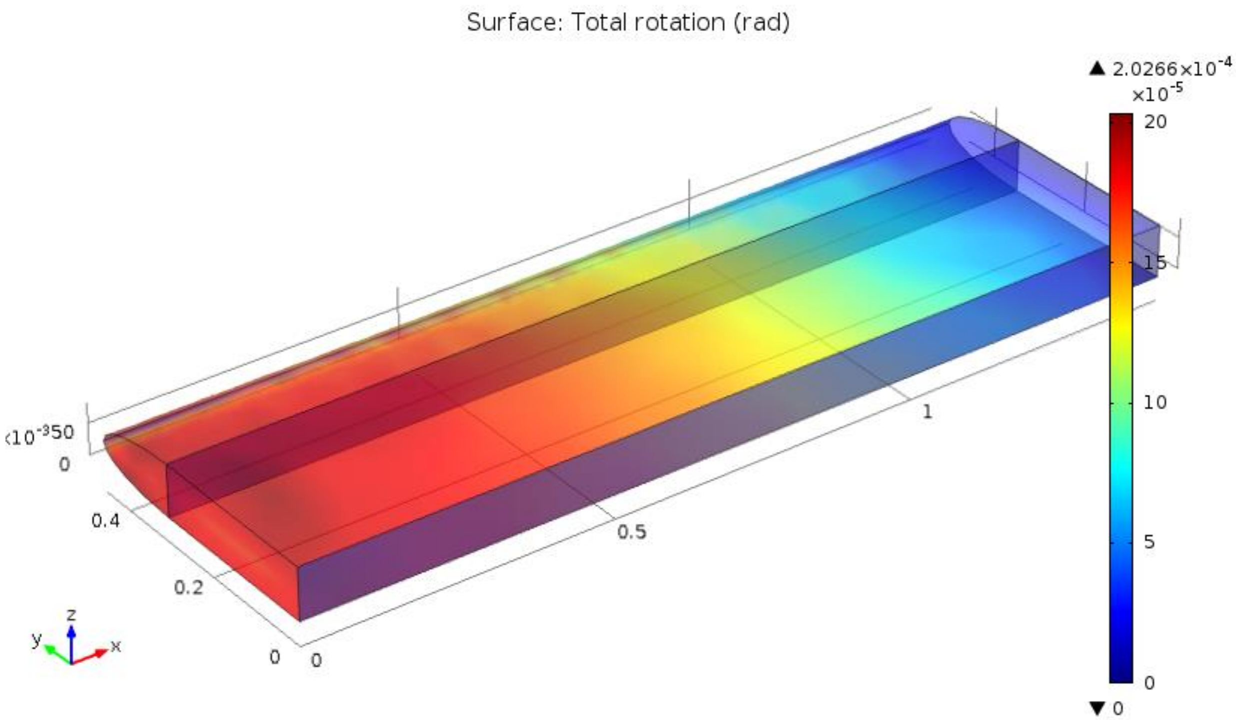


Total displacement (m): 3.16107e-5, 3.04e-5
Strain tensor (global), xx component (1): -5.78263e-6, 5.48229e-6
Strain tensor (global), xy component (1): -3.70139e-7, 5.80354e-7

Point 4 Fine

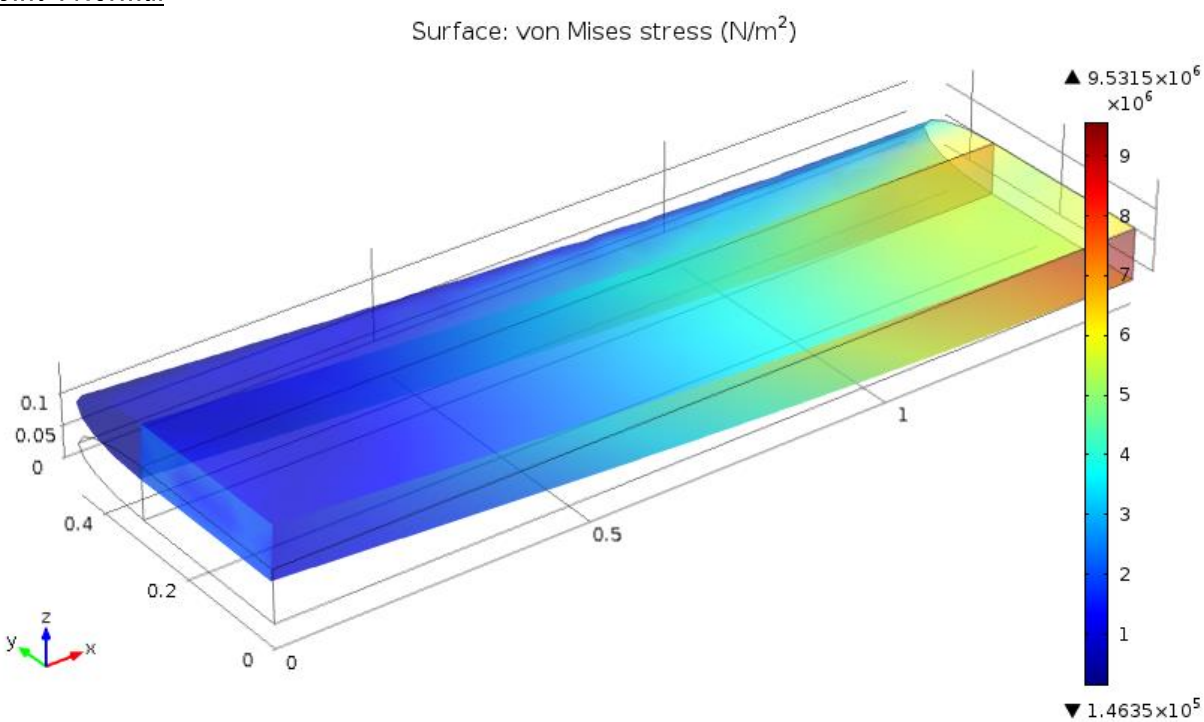

Surface: von Mises stress (N/m2)
▲ 9.933×10^6
×10^6
9
8
7
6
5
4
3
2
1
▼ 1.549×10^5
0.1
0.05
0
0.4
0.2
0
0
0.5
1
y
z
x

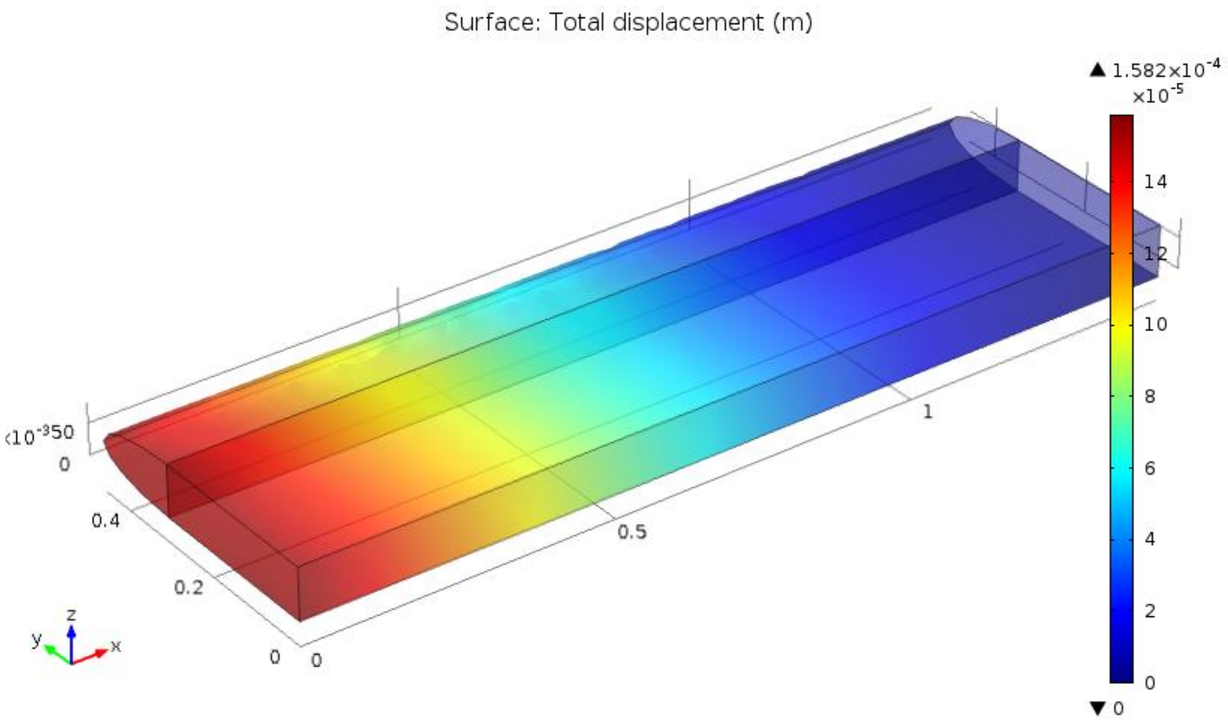

Surface: Total displacement (m)
▲ 1.8549×10^-4
×10^-5
18
16
14
12
10
8
6
4
2
0
▼ 0
0
0.4
0.2
0
0
0.5
1
y
z
x

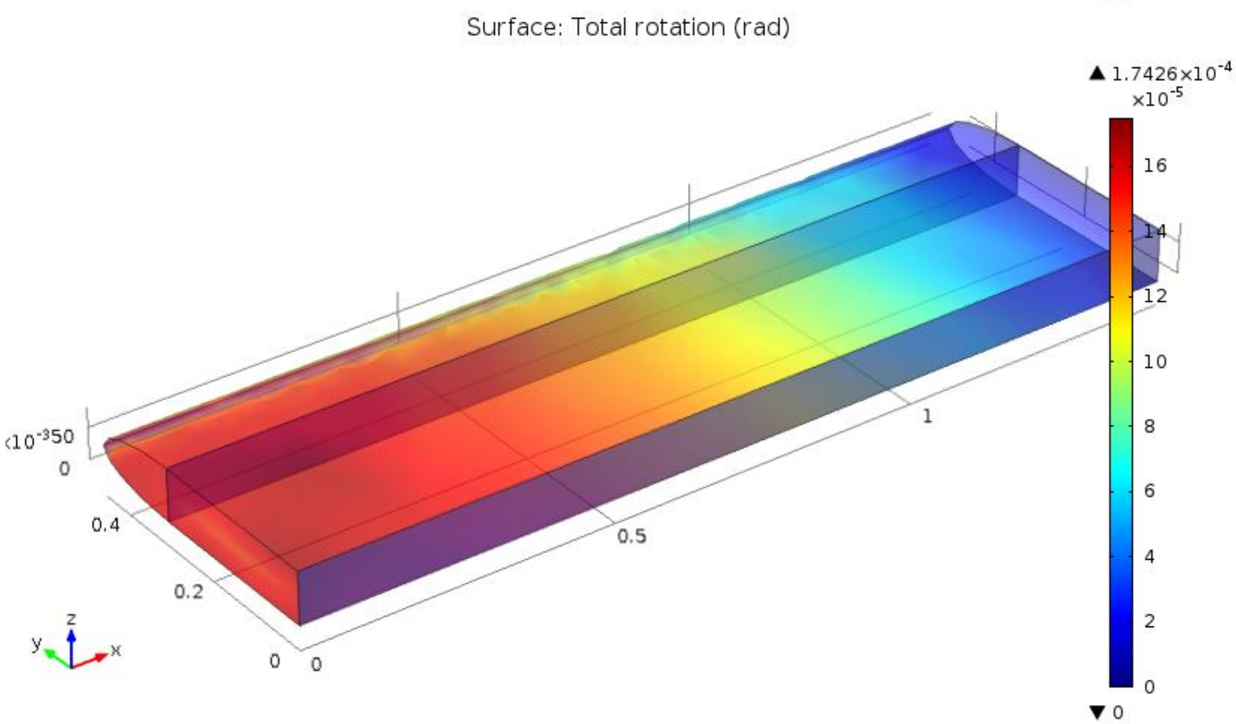


Total displacement (m): 3.80335e-5, 3.73504e-5
Strain tensor (global), xx component (1): -5.97811e-6, 6.39446e-6
Strain tensor (global), xy component (1): -2.80133e-7, 4.82898e-7

<u>Point 4 Normal</u>

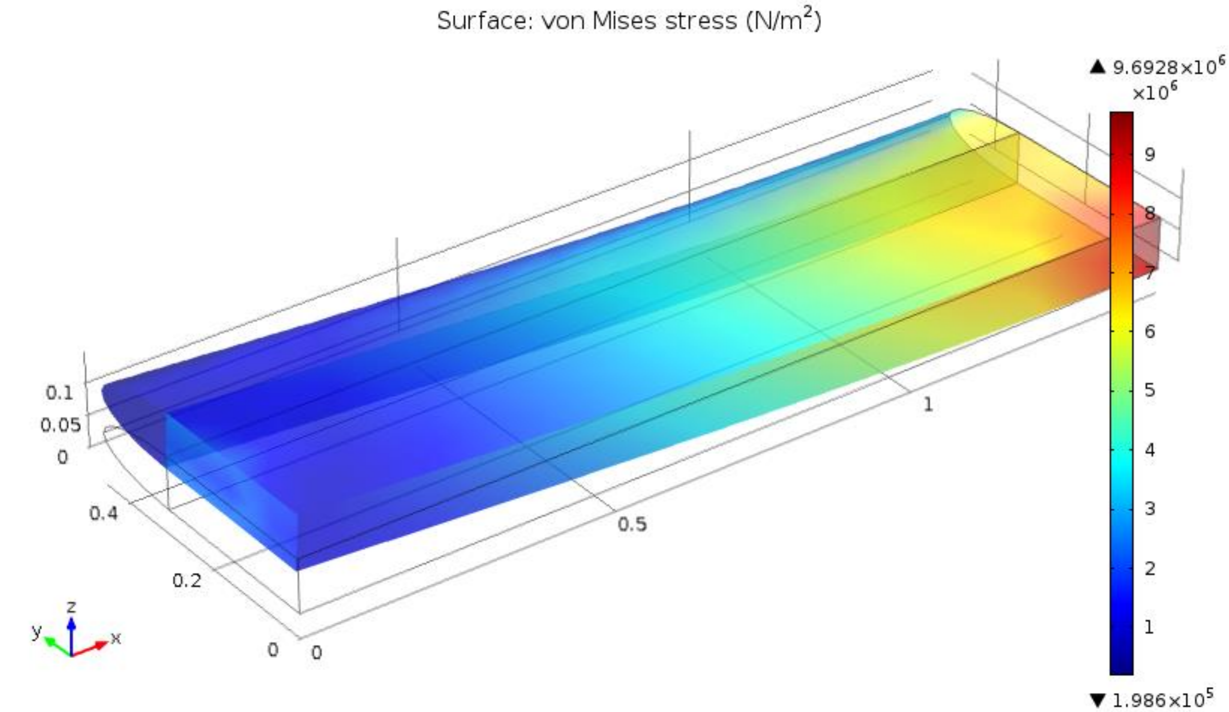




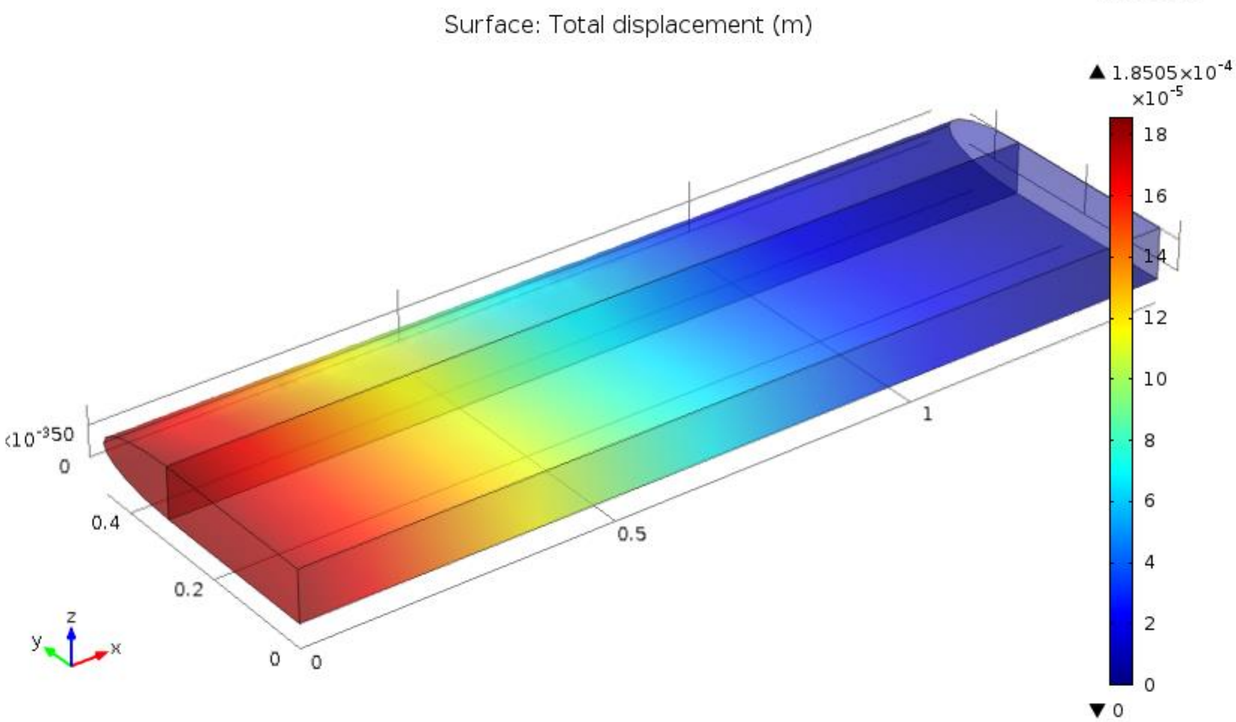


Total displacement (m): 3.14376e-5, 3.02757e-5
Strain tensor (global), xx component (1): -5.79569e-6, 5.48693e-6
Strain tensor (global), xy component (1): -7.03952e-8, 2.80856e-7

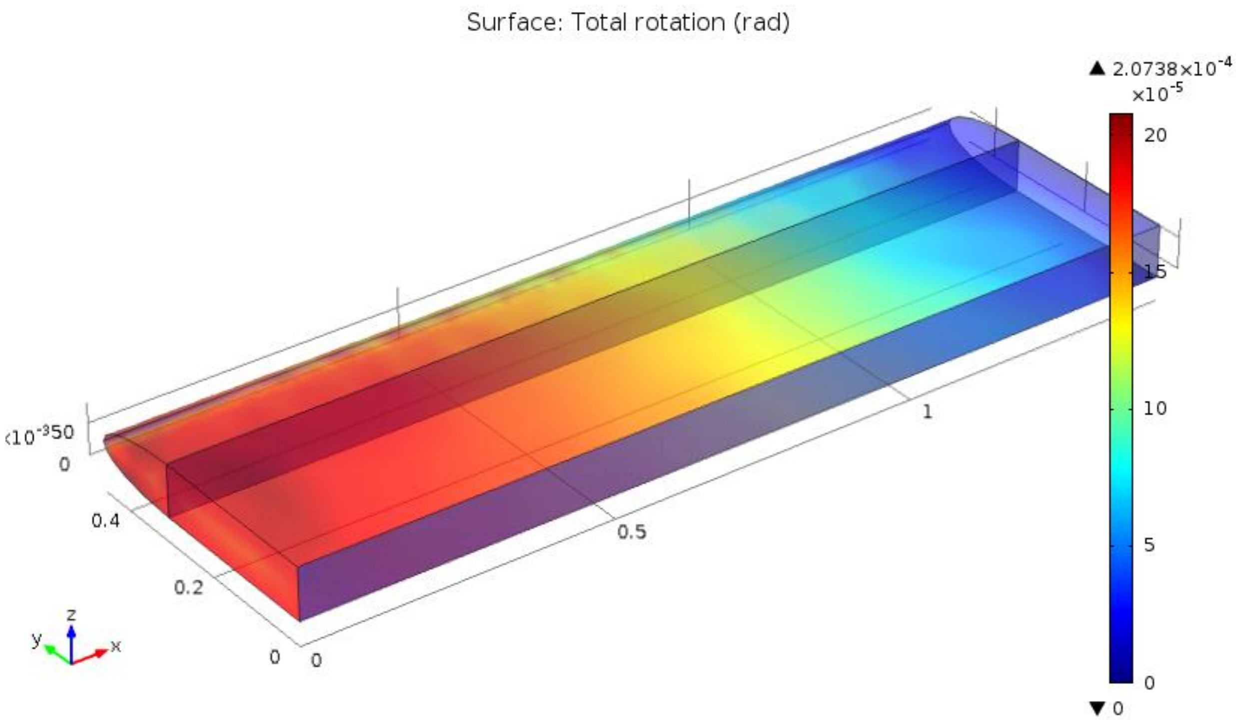
Point 5 Fine



Surface: von Mises stress (N/m²)
▲ 9.6928×10⁶
×10⁶
9
8
7
6
5
4
3
2
1
▼ 1.986×10⁵
0.1
0.05
0
0.4
0.2
0
0
0.5
1

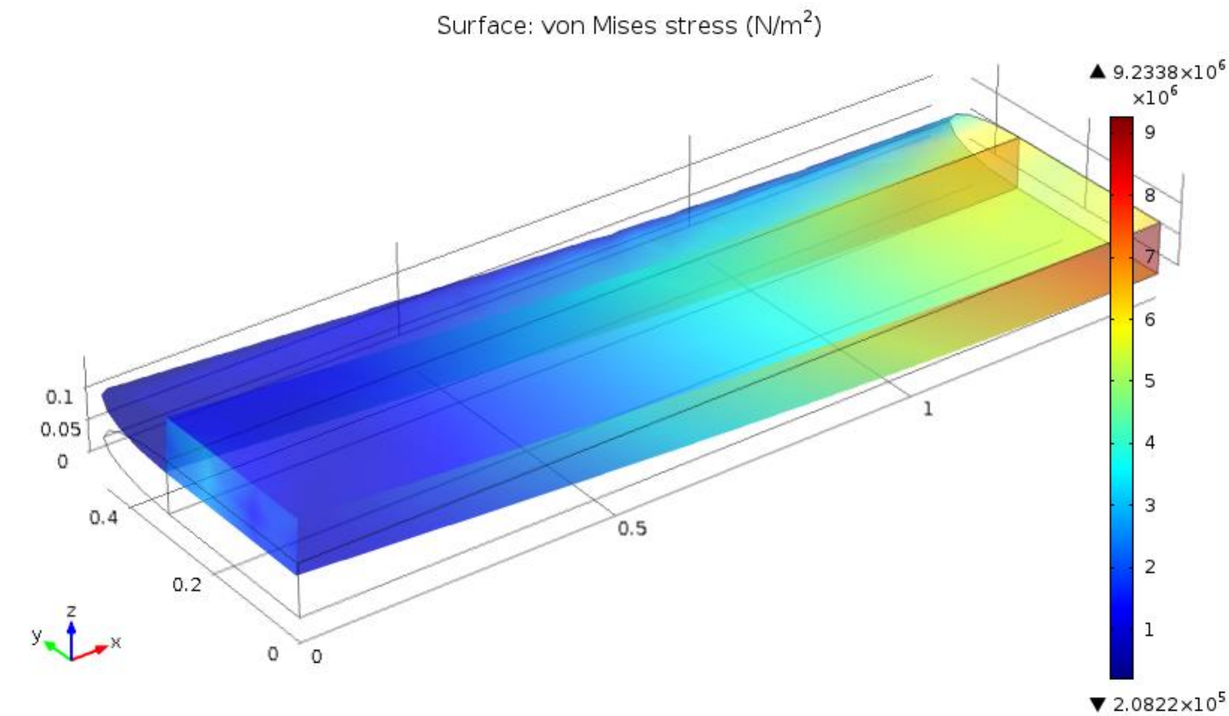


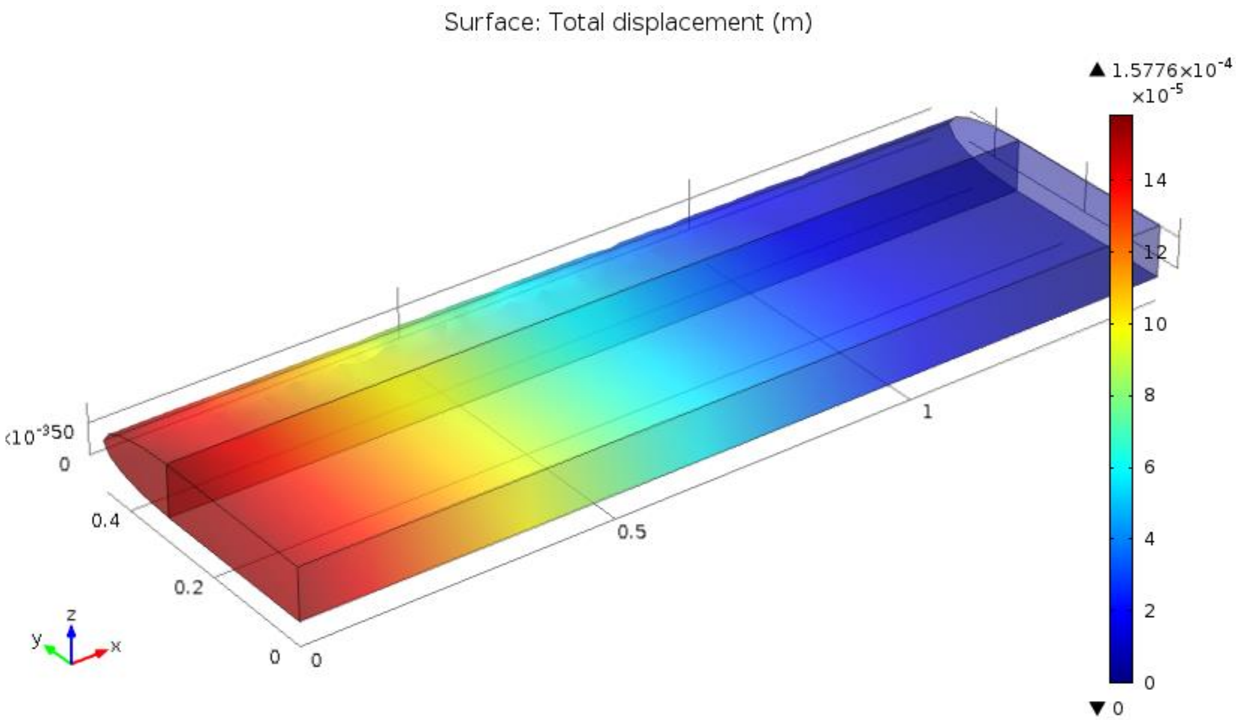
Surface: Total displacement (m)
▲ 1.8505×10⁻⁴
×10⁻⁵
18
16
14
12
10
8
6
4
2
0
▼ 0
0.4
0.2
0
0
0.5
1

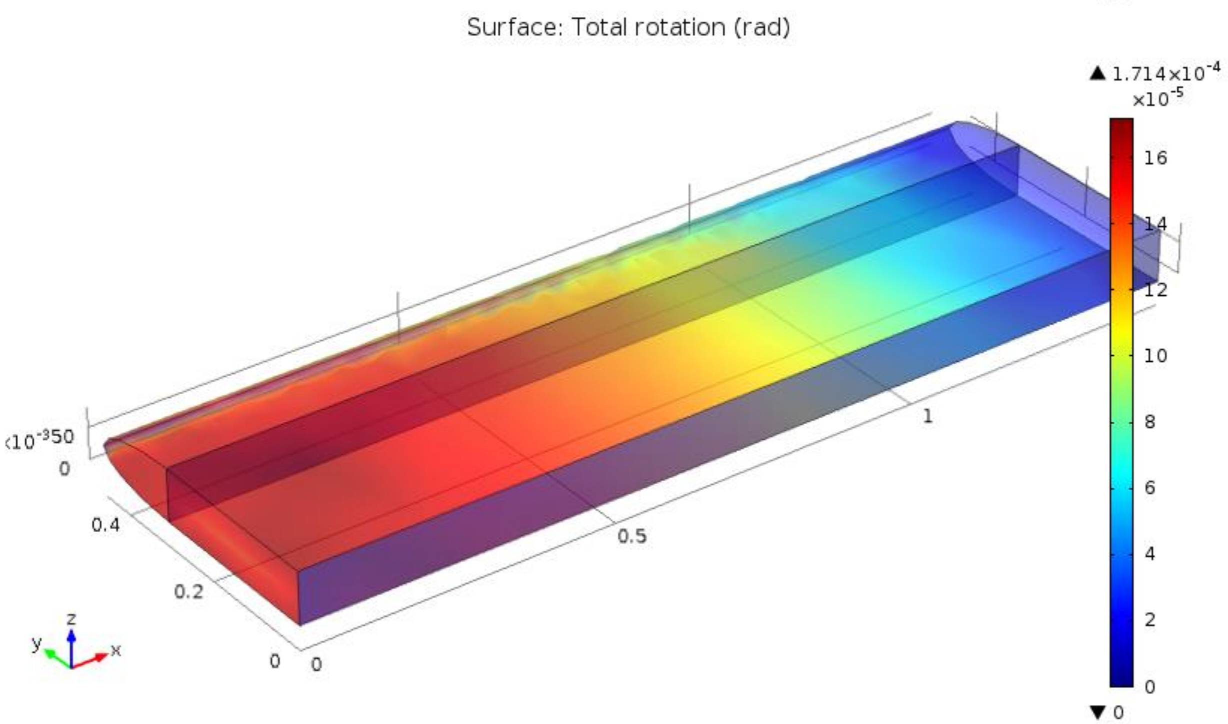


Total displacement (m): 3.78313e-5, 3.72558e-5
Strain tensor (global), xx component (1): -5.9802e-6, 6.39588e-6
Strain tensor (global), xy component (1): 1.78257e-8, 1.80655e-7

Point 5 Normal





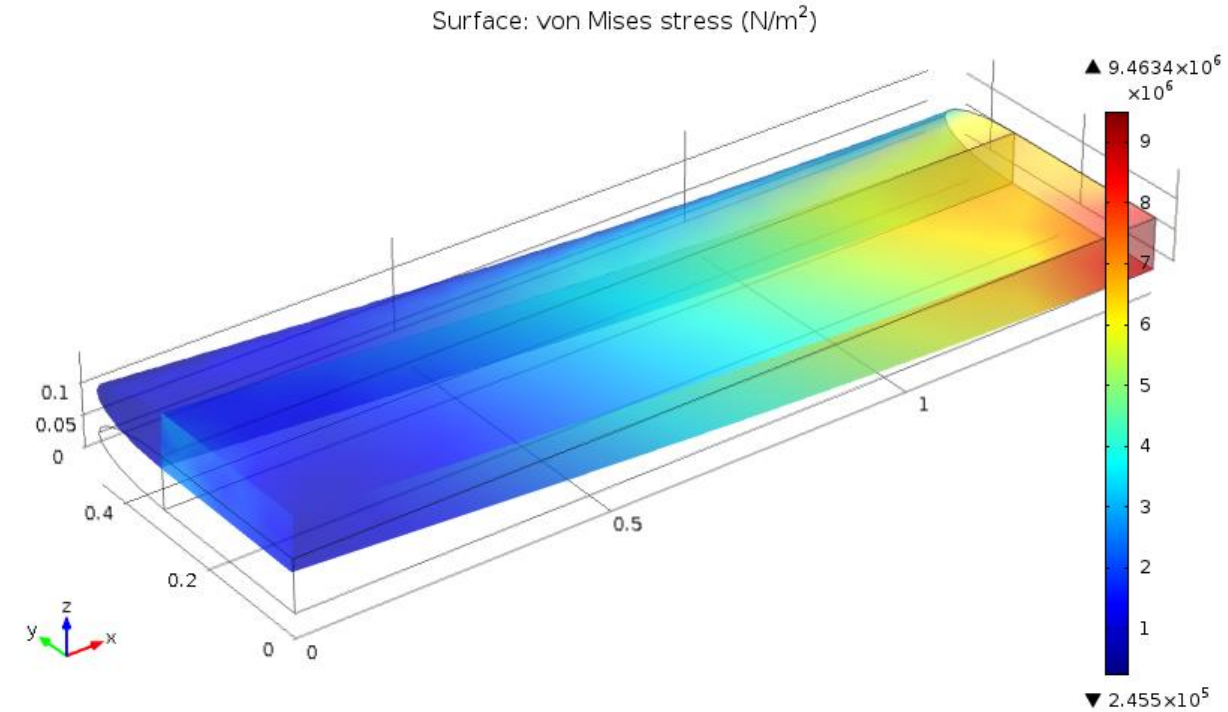


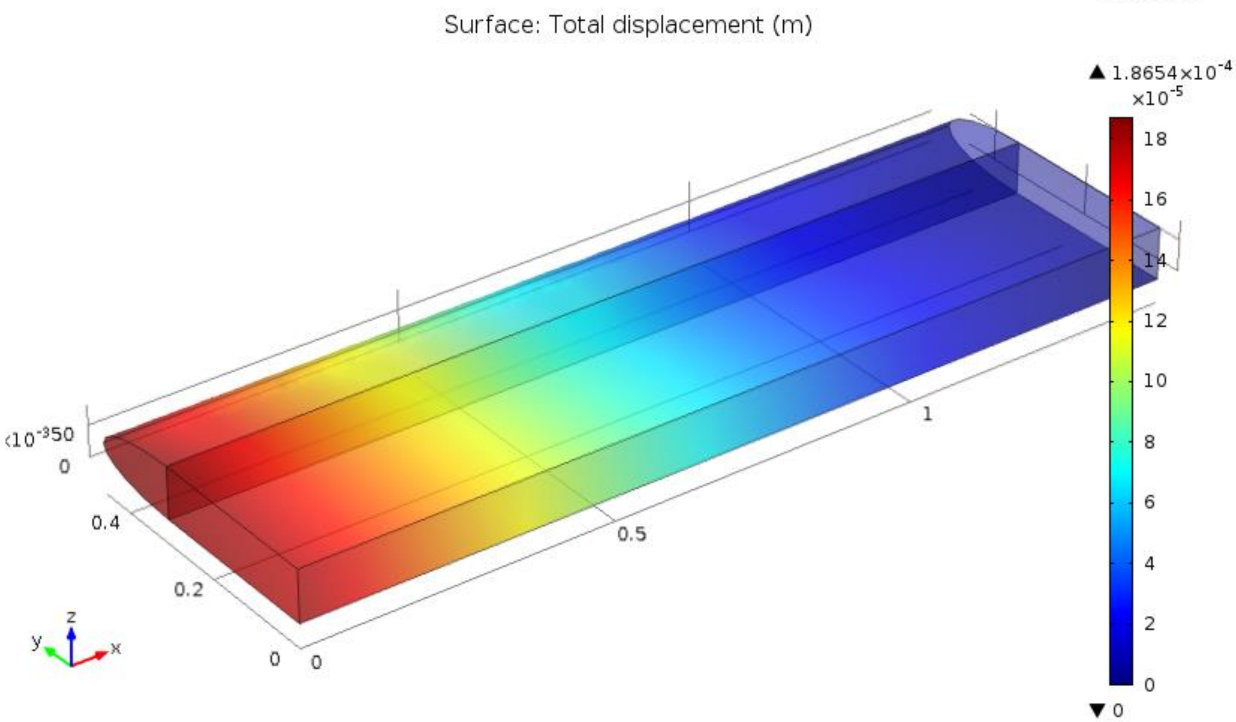
Total displacement (m): 3.12634e-5, 3.0148e-5

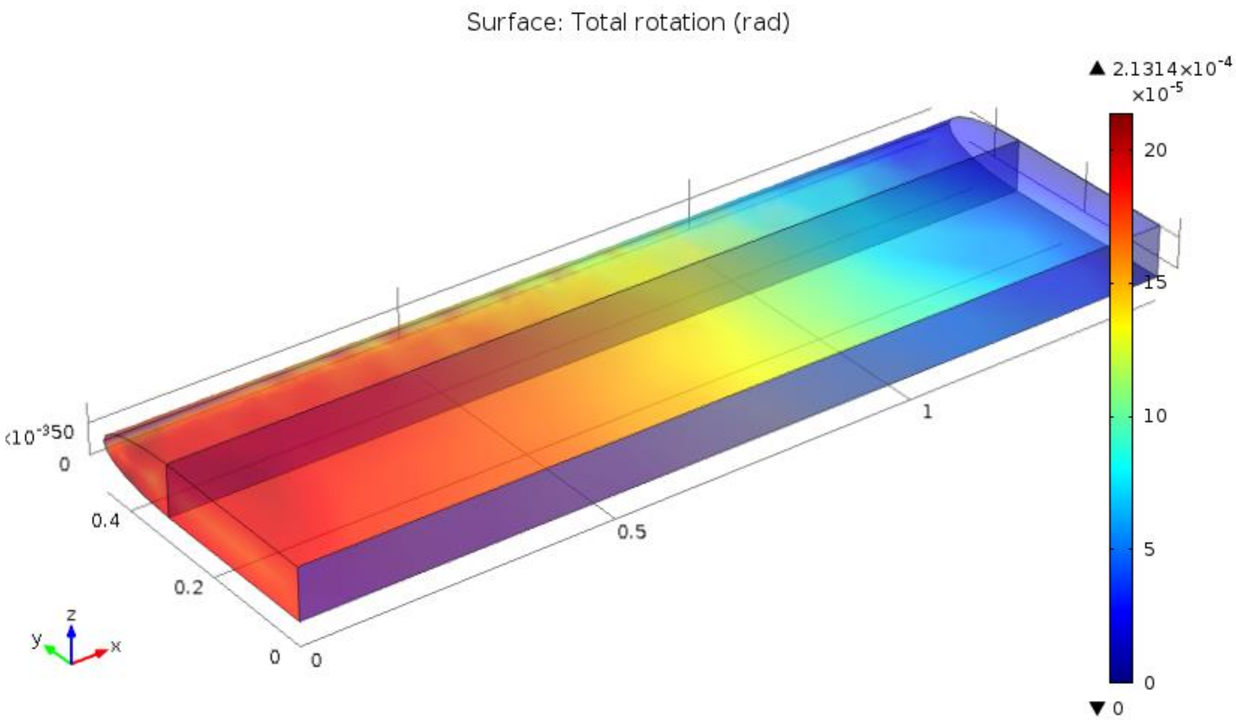
Strain tensor (global), xx component (1): -5.80885e-6, 5.49184e-6

Strain tensor (global), xy component (1): 2.28841e-7, -1.8784e-8

Point 6 Fine

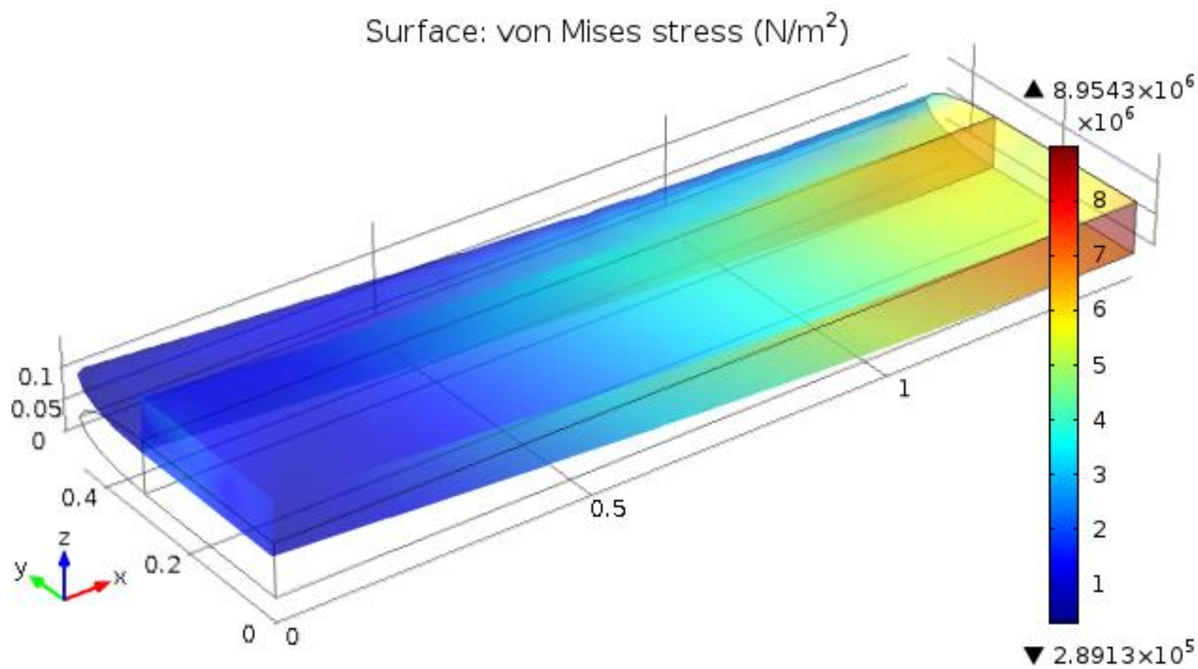

Surface: von Mises stress (N/m²)
▲ 9.4634×10⁶
×10⁶
9
8
7
6
5
4
3
2
1
▼ 2.455×10⁵
0.1
0.05
0
0.4
0.2
0
0
0.5
1
z
y
x

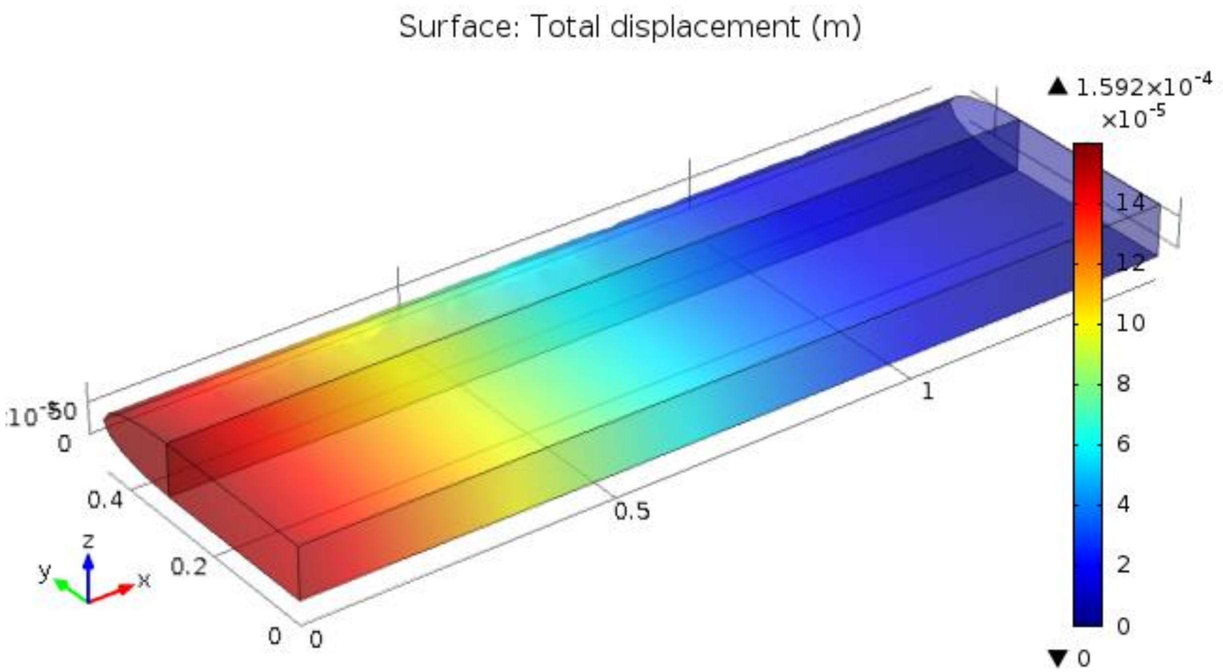

Surface: Total displacement (m)
▲ 1.8654×10⁻⁴
×10⁻⁵
18
16
14
12
10
8
6
4
2
0
▼ 0
0.4
0.2
0
0
0.5
1
z
y
x

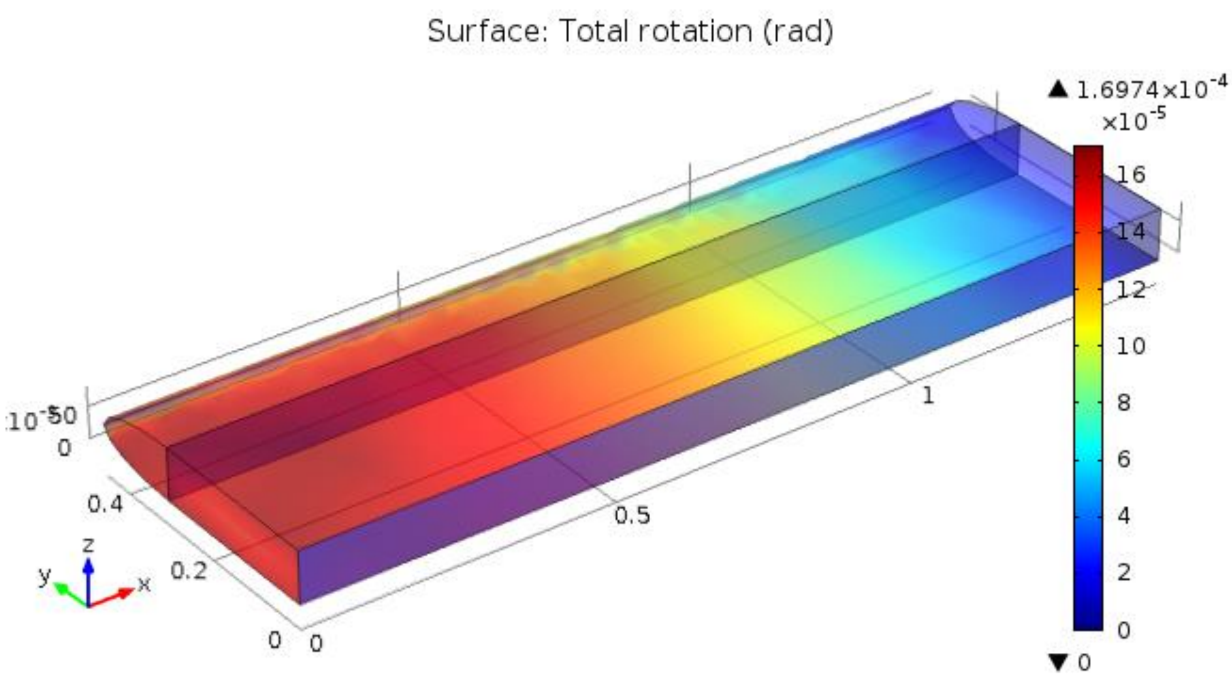


Total displacement (m): 3.76108e-5, 3.71476e-5
Strain tensor (global), xx component (1): -5.98193e-6, 6.39666e-6
Strain tensor (global), xy component (1): 3.12404e-7, -1.19302e-7

<u>Point 6 Normal</u>

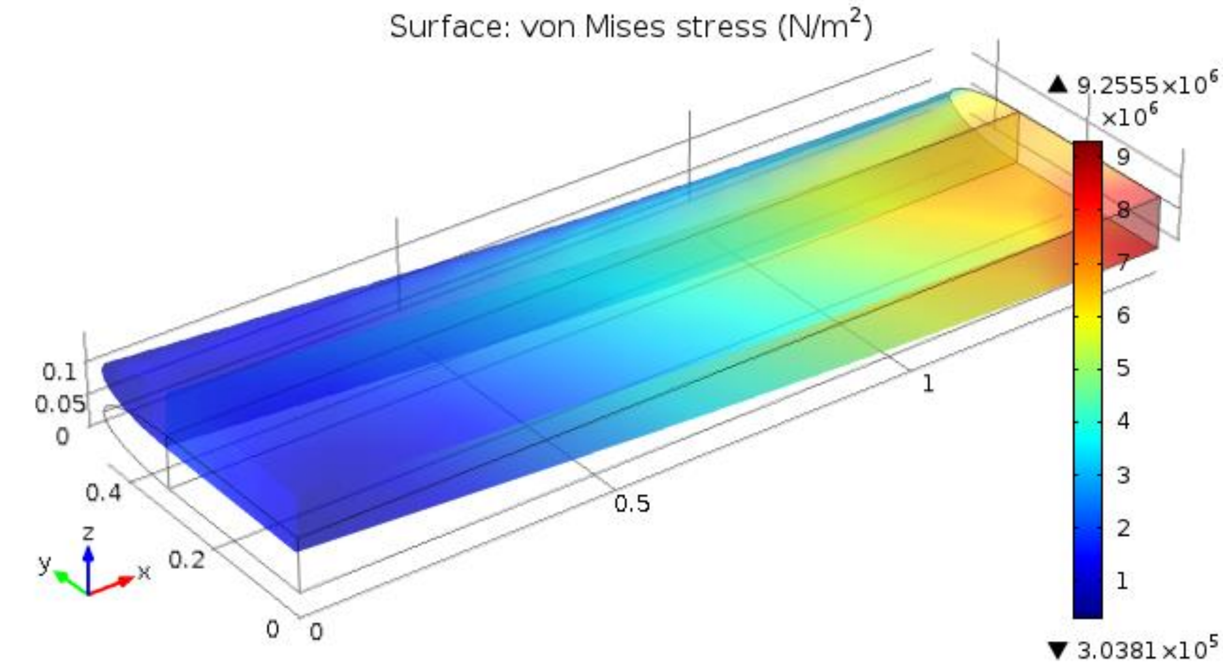


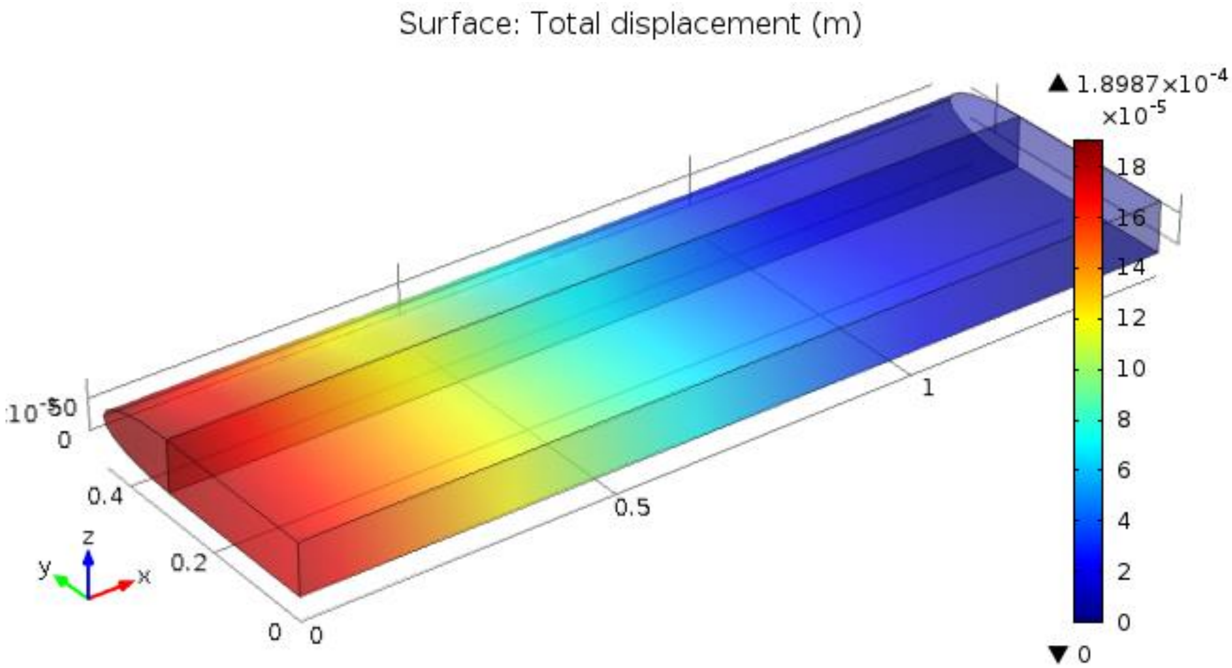


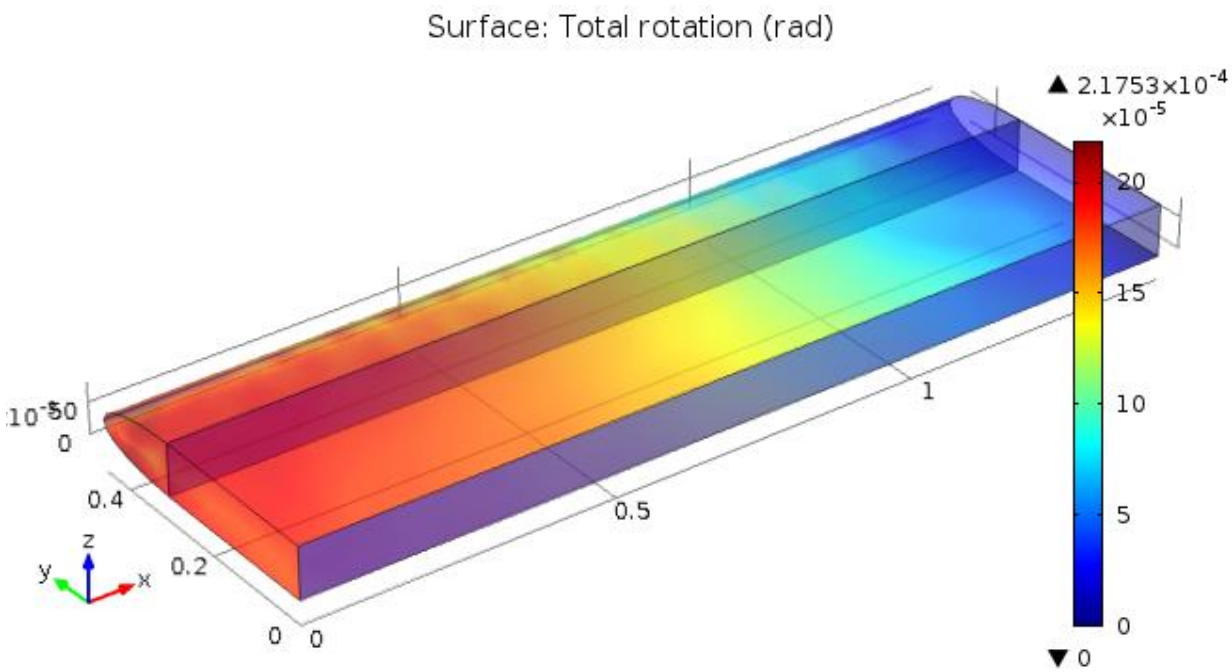


Total displacement (m): 3.10793e-5, 3.00078e-5
Strain tensor (global), xx component (1): -5.82037e-6, 5.50109e-6
Strain tensor (global), xy component (1): 5.26396e-7, -3.19371e-7

Point 7 Fine

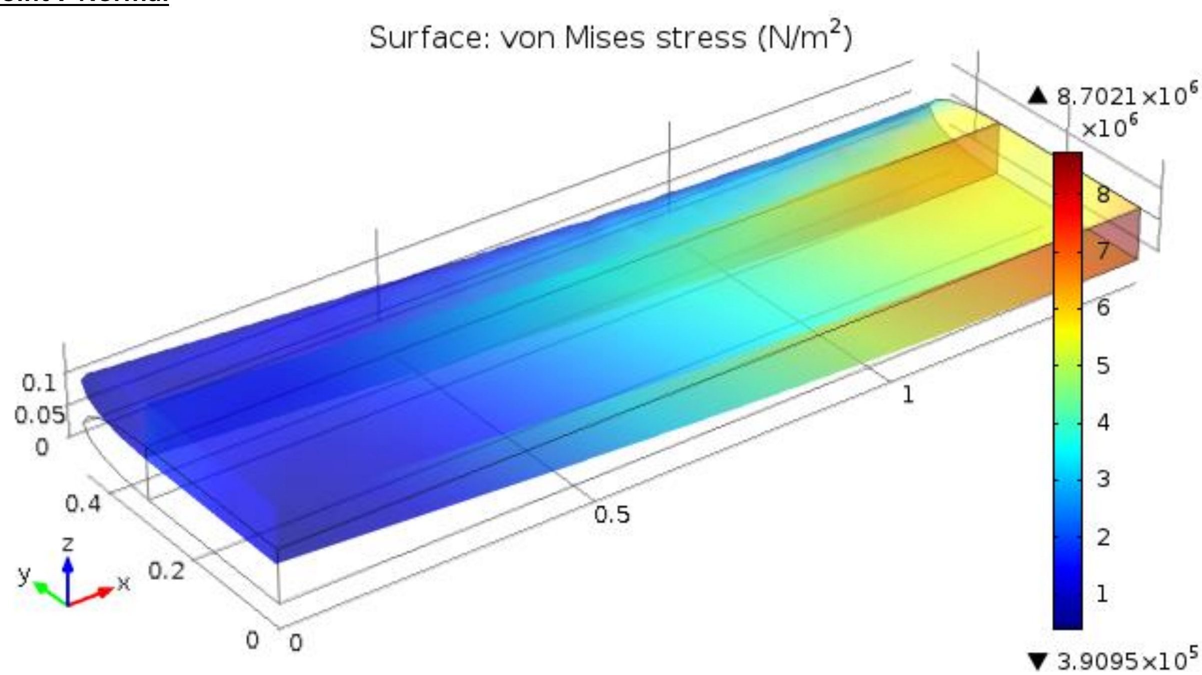

Surface: von Mises stress (N/m²)
▲ 9.2555×10⁶
×10⁶
9
8
7
6
5
4
3
2
1
▼ 3.0381×10⁵
0.1
0.05
0
0.4
0.2
0
0
0.5
1
z
y
x

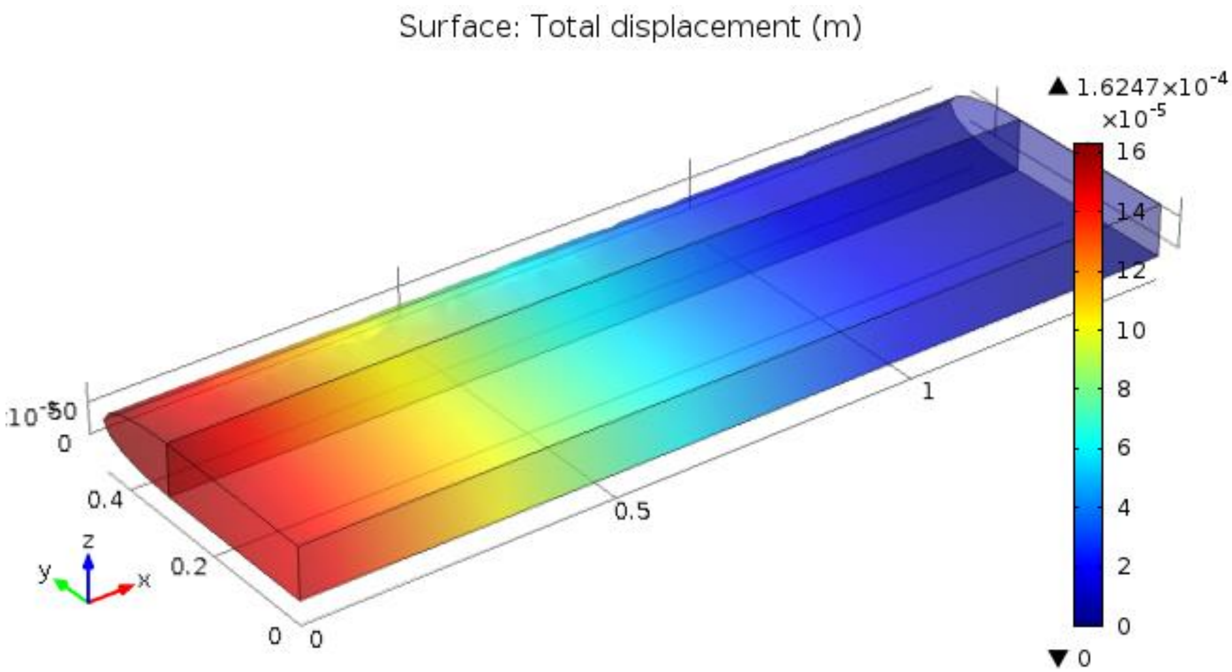

Surface: Total displacement (m)
▲ 1.8987×10⁻⁴
×10⁻⁵
18
16
14
12
10
8
6
4
2
0
▼ 0
0
0.4
0.2
0
0
0.5
1
z
y
x

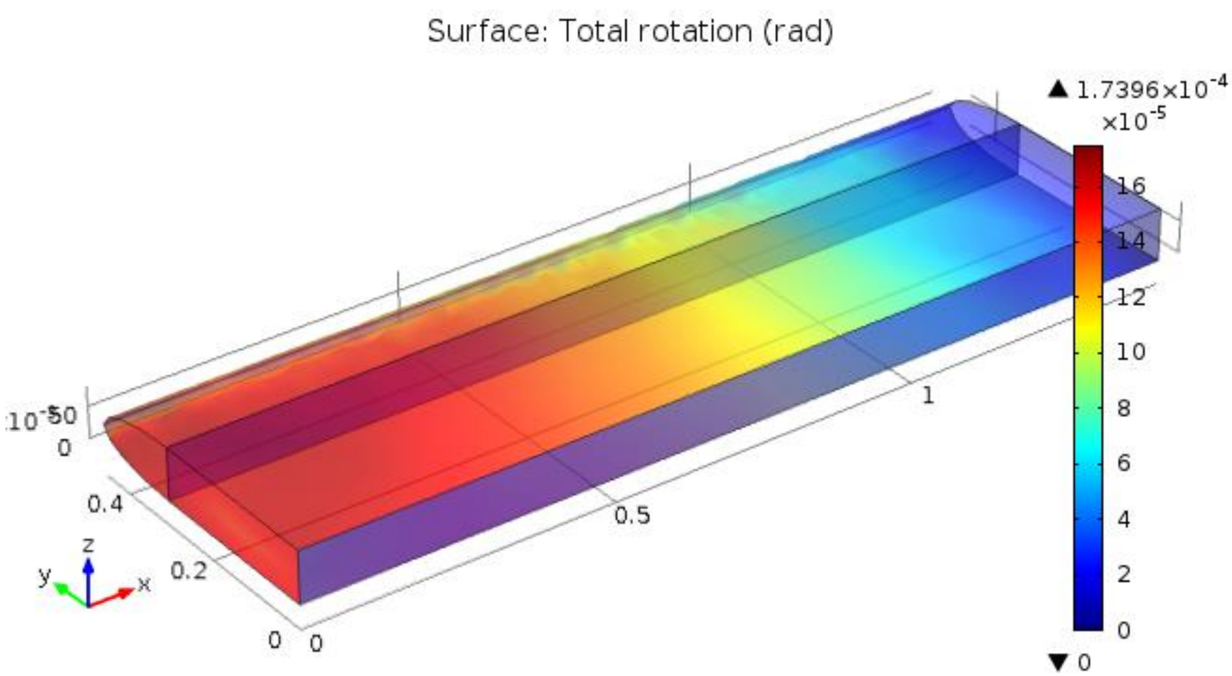


Total displacement (m): 3.74119e-5, 3.70275e-5
Strain tensor (global), xx component (1): -5.9825e-6, 6.3946e-6
Strain tensor (global), xy component (1): 6.02816e-7, -4.13951e-7

<u>Point 7 Normal</u>



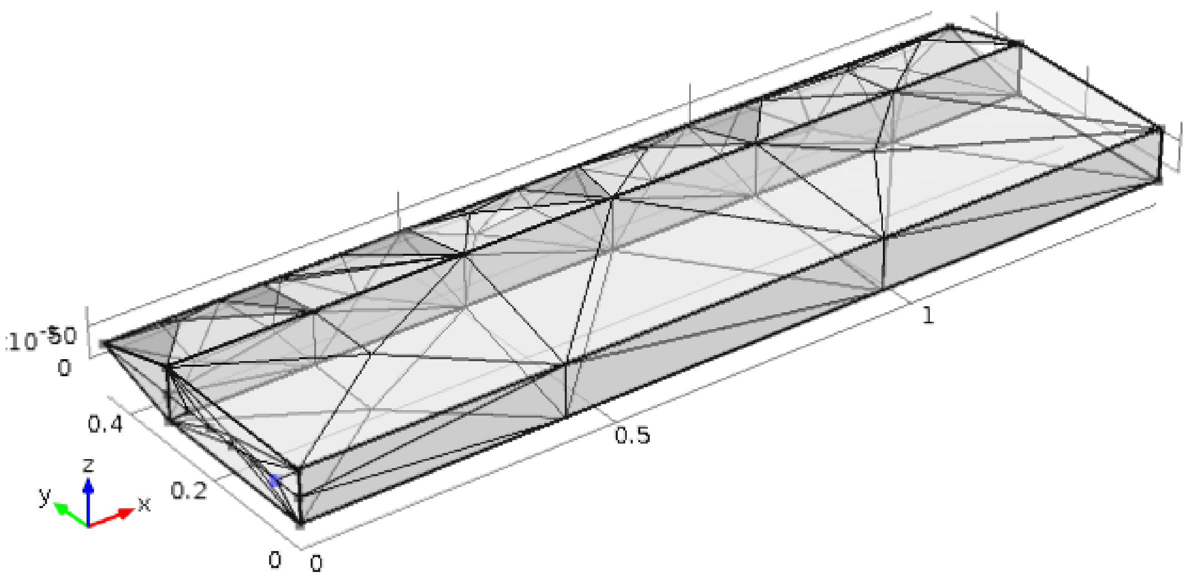


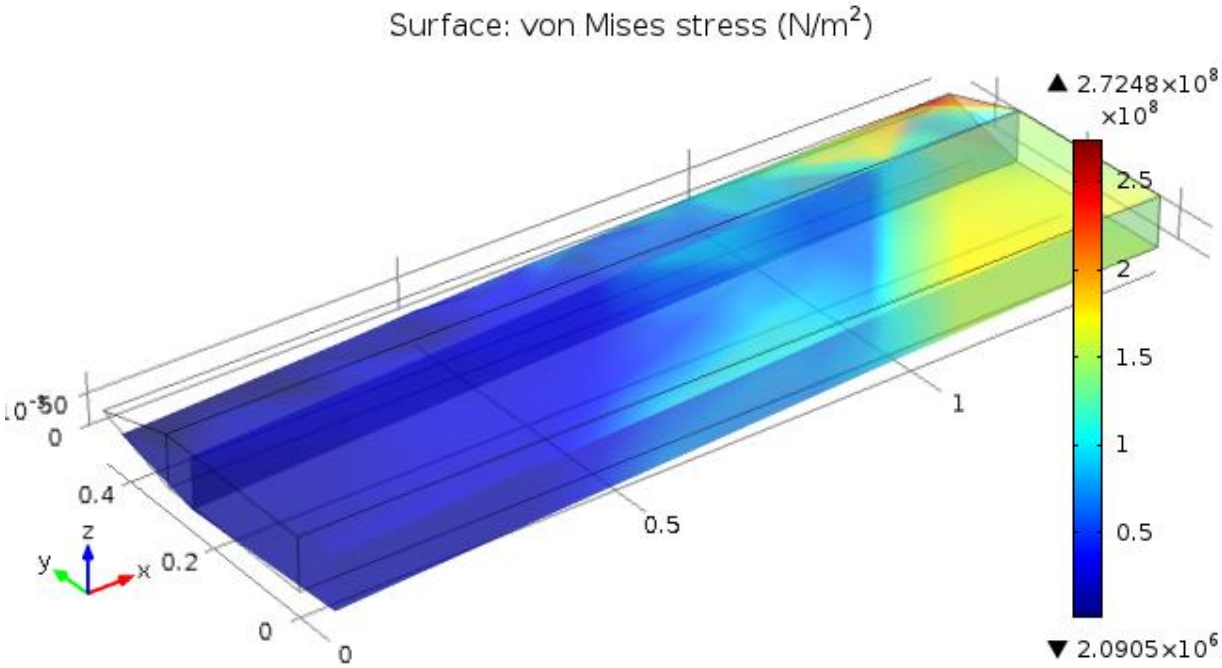


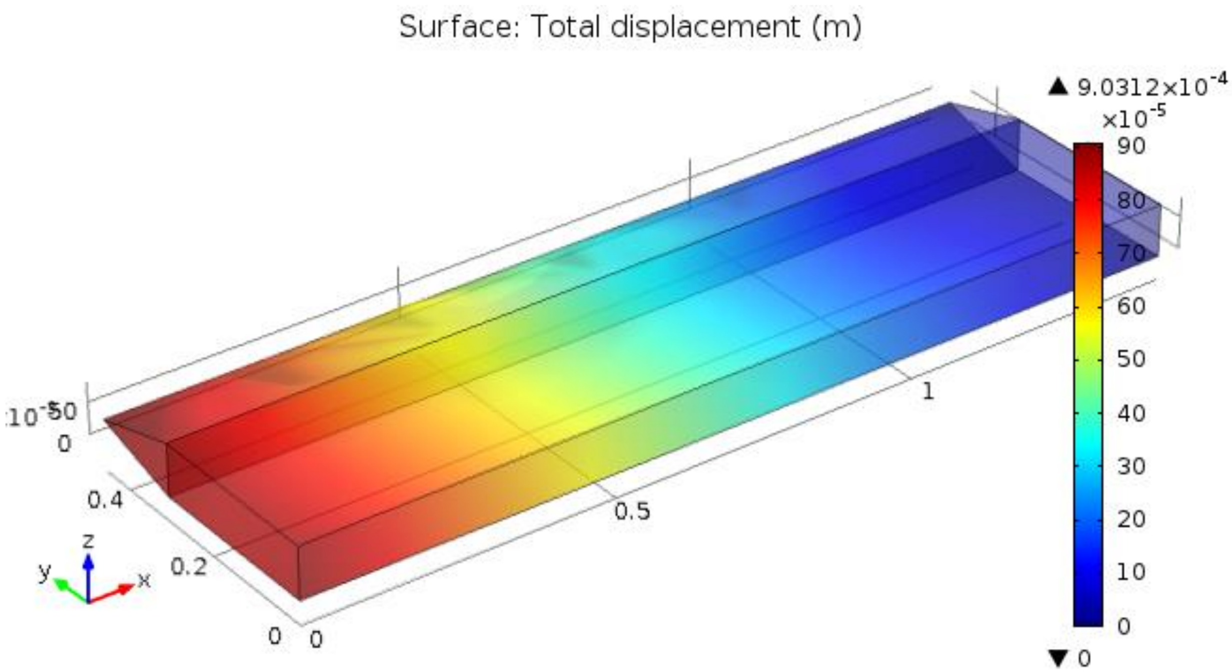
Total displacement (m): 3.08948e-5, 2.98539e-5
Strain tensor (global), xx component (1): -5.83152e-6, 5.51094e-6
Strain tensor (global), xy component (1): 8.22676e-7, -6.20049e-7

The next 12 figures show the wing under a drag load. The same analysis is done for a range of mesh densities in order to observe its effect on the data.

Faceload Ext C

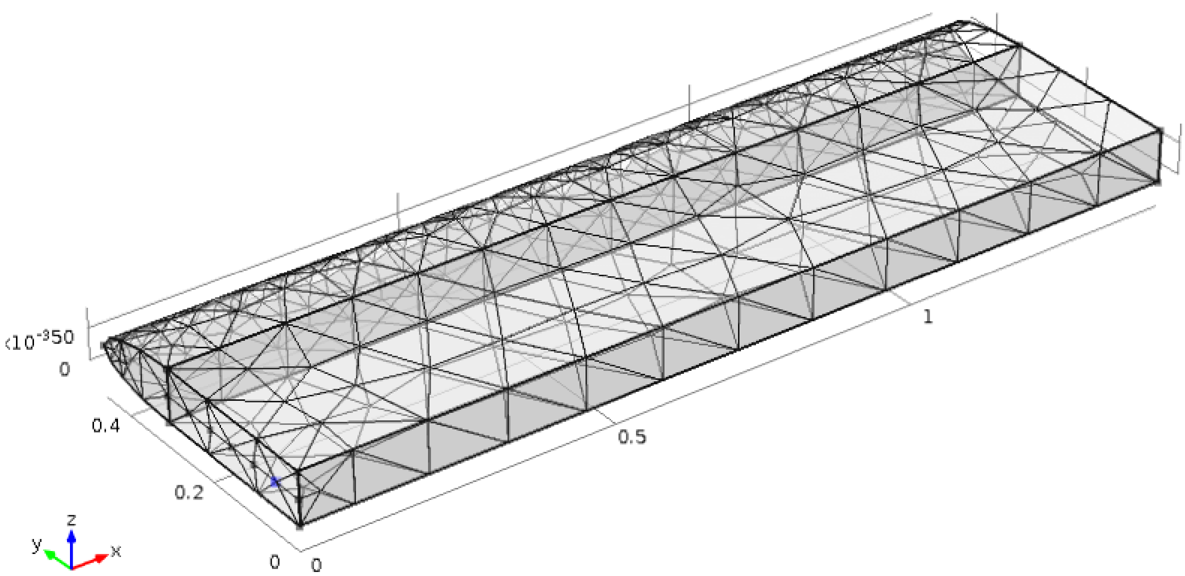


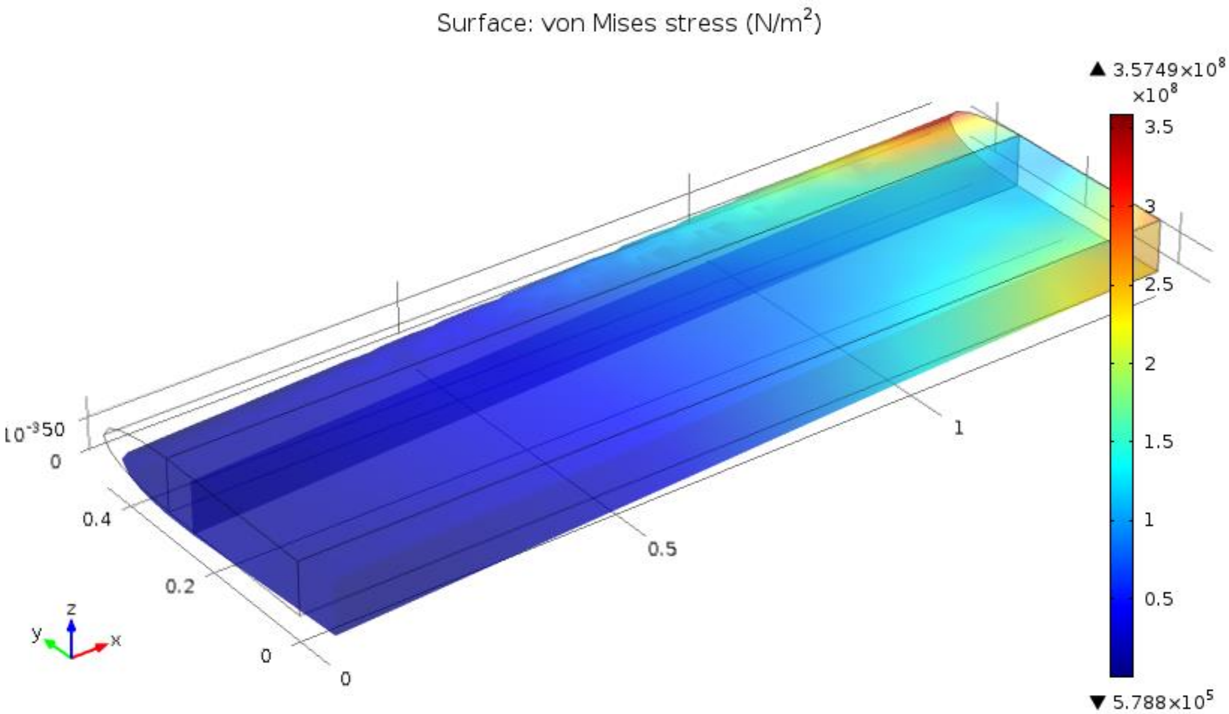


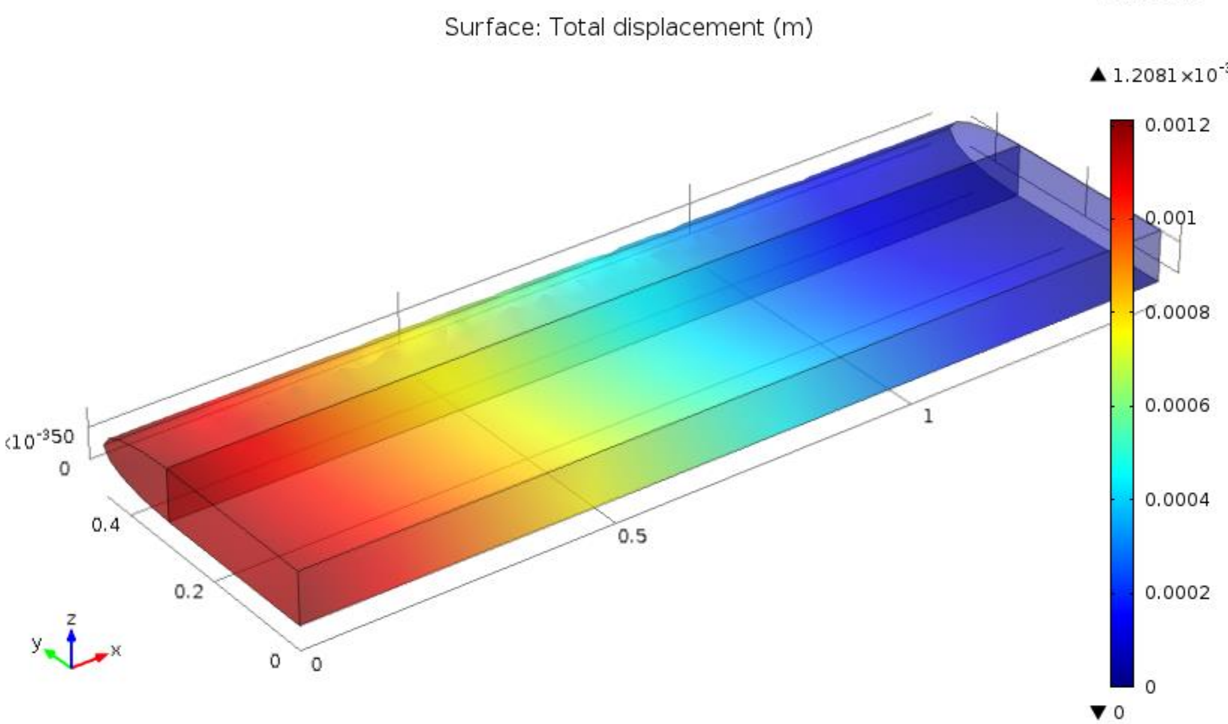


Total displacement (m): 2.47675e-4, 2.47245e-4
Strain tensor (global), xy component (1): 7.67168e-5, 7.75789e-5
Strain tensor (global), xx component (1): 7.67168e-5, 5.74752e-5

Normal

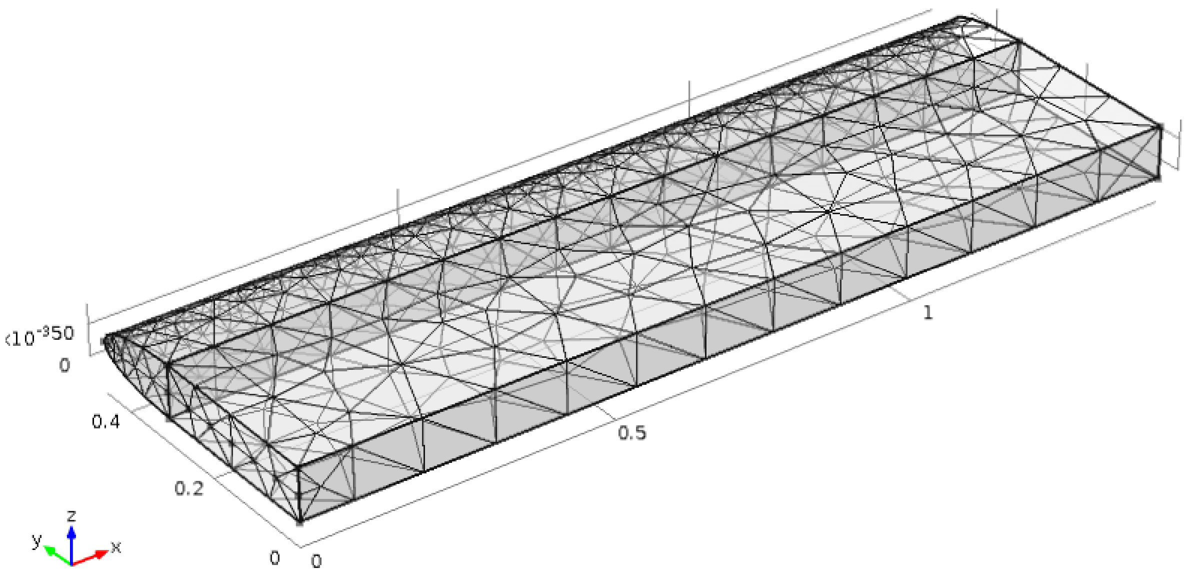




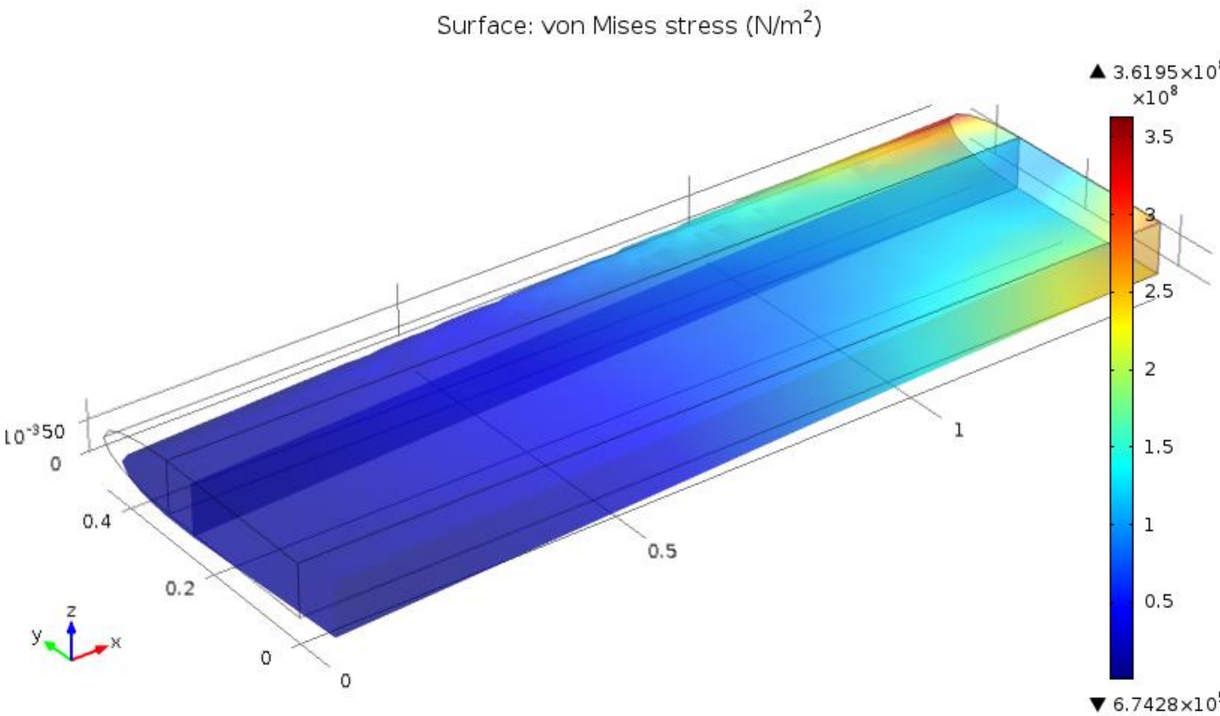


Total displacement (m): 3.1711e-4, 3.17734e-4
Strain tensor (global), xy component (1): -9.01525e-6, -9.69214e-6
Strain tensor (global), xy component (1): 7.17233e-5, 7.32779e-5

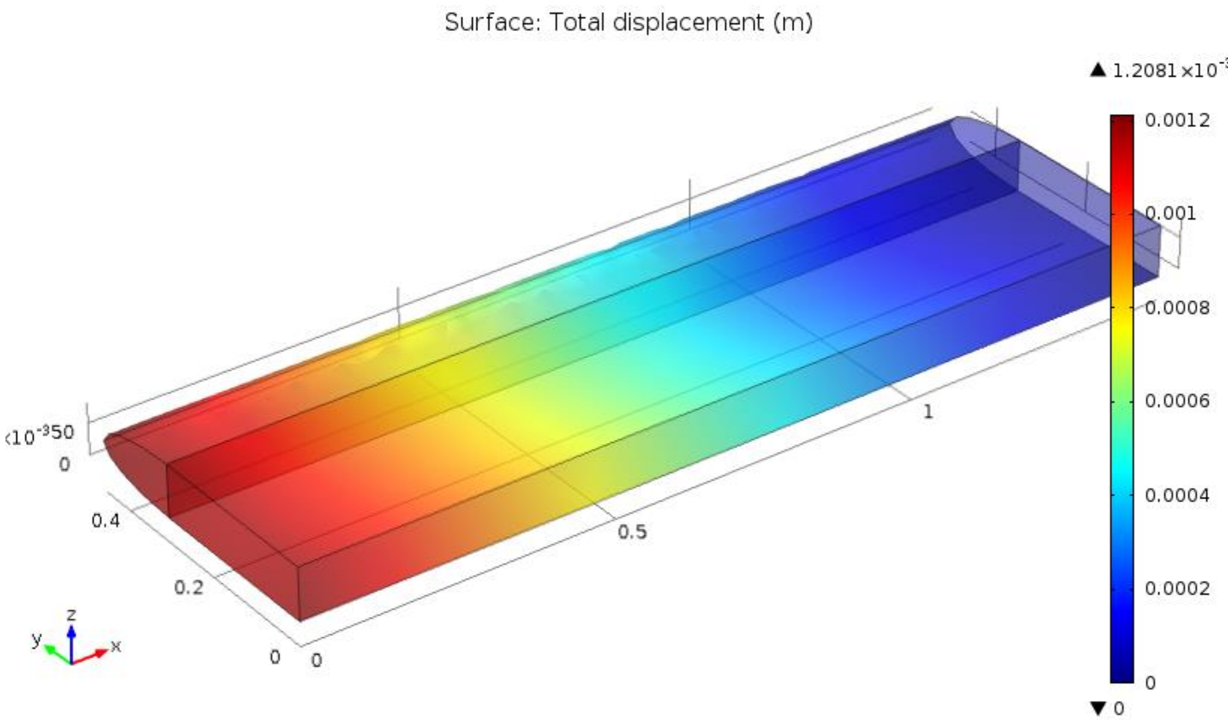
Fine



×10⁻³50
0
0.4
0.2
0
0
0.5
1
y
z
x

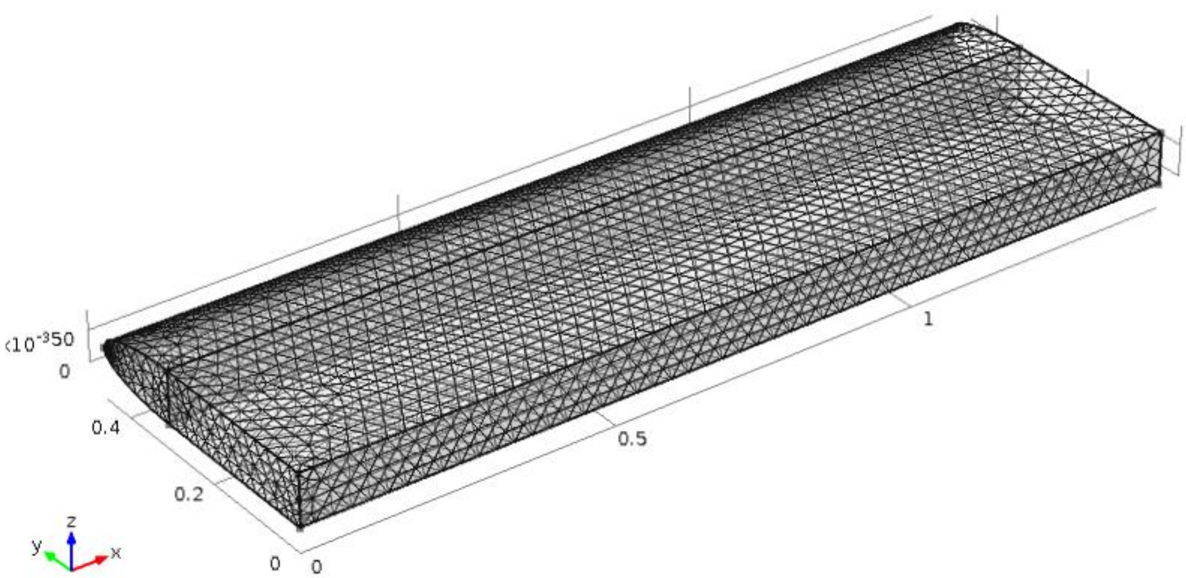


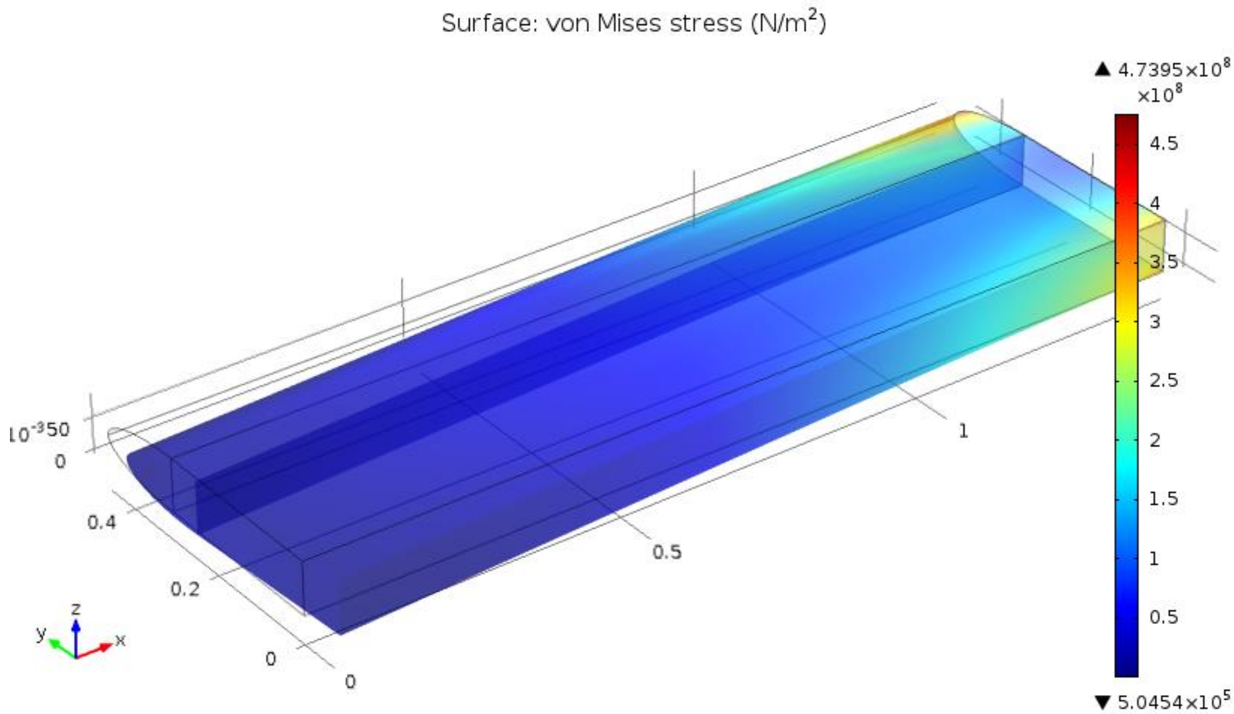
Surface: von Mises stress (N/m²)
▲ 3.6195×10⁸
×10⁸
3.5
3
2.5
2
1.5
1
0.5
▼ 6.7428×10⁵
0.4
0.2
0
0
0.5
1
y
z
x

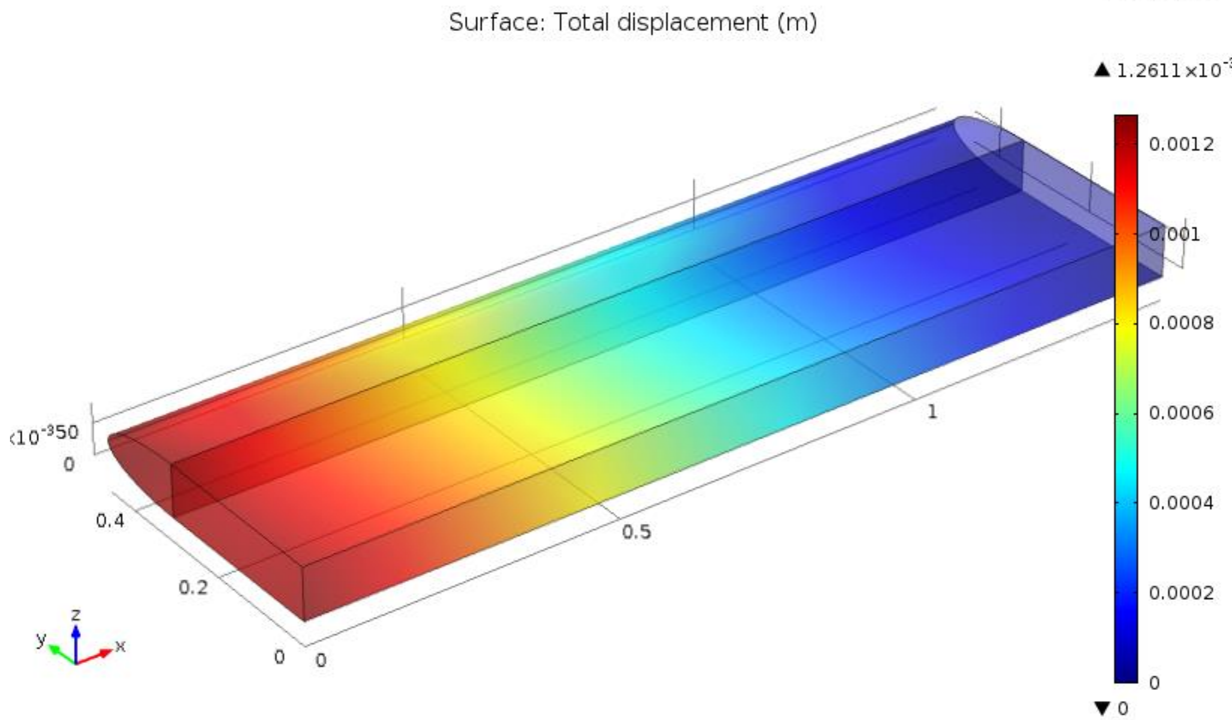


Total displacement (m): 3.1711e-4, 3.17734e-4
Strain tensor (global), xx component (1): -9.01525e-6, -9.69214e-6
Strain tensor (global), xy component (1): 7.17233e-5, 7.32779e-5

Ext Fine

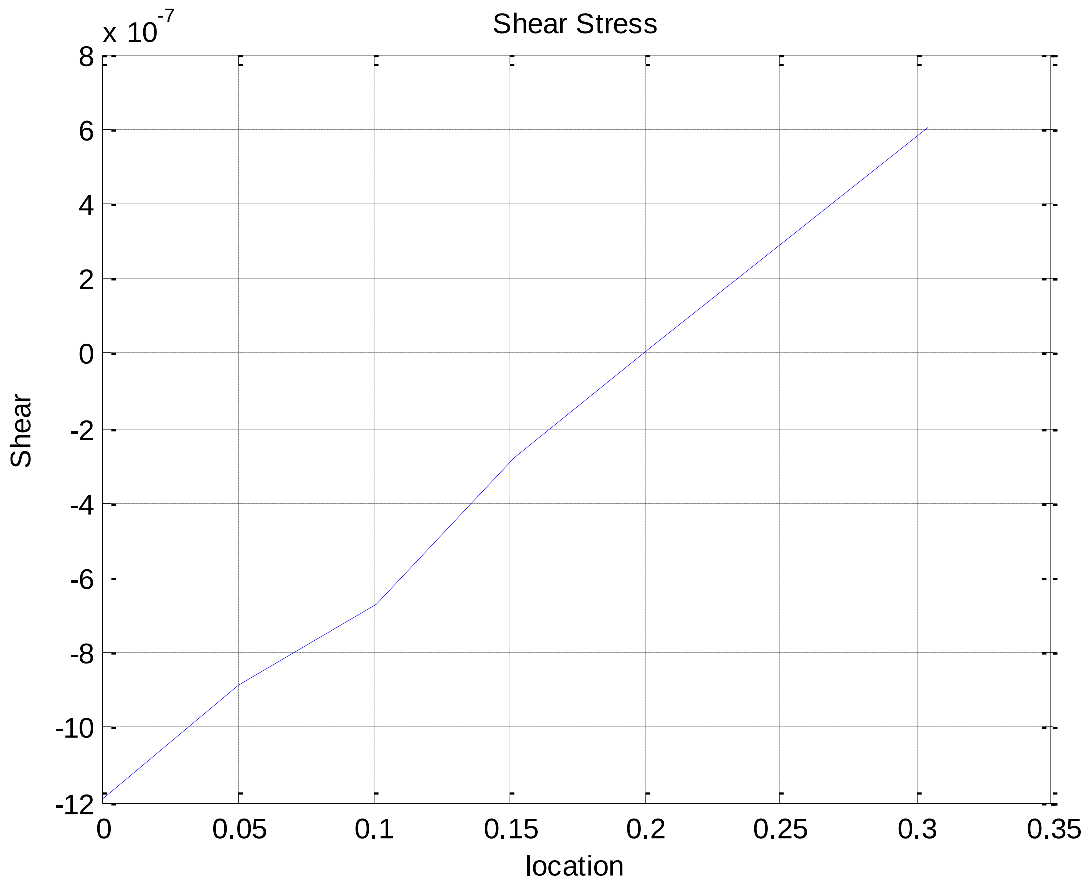




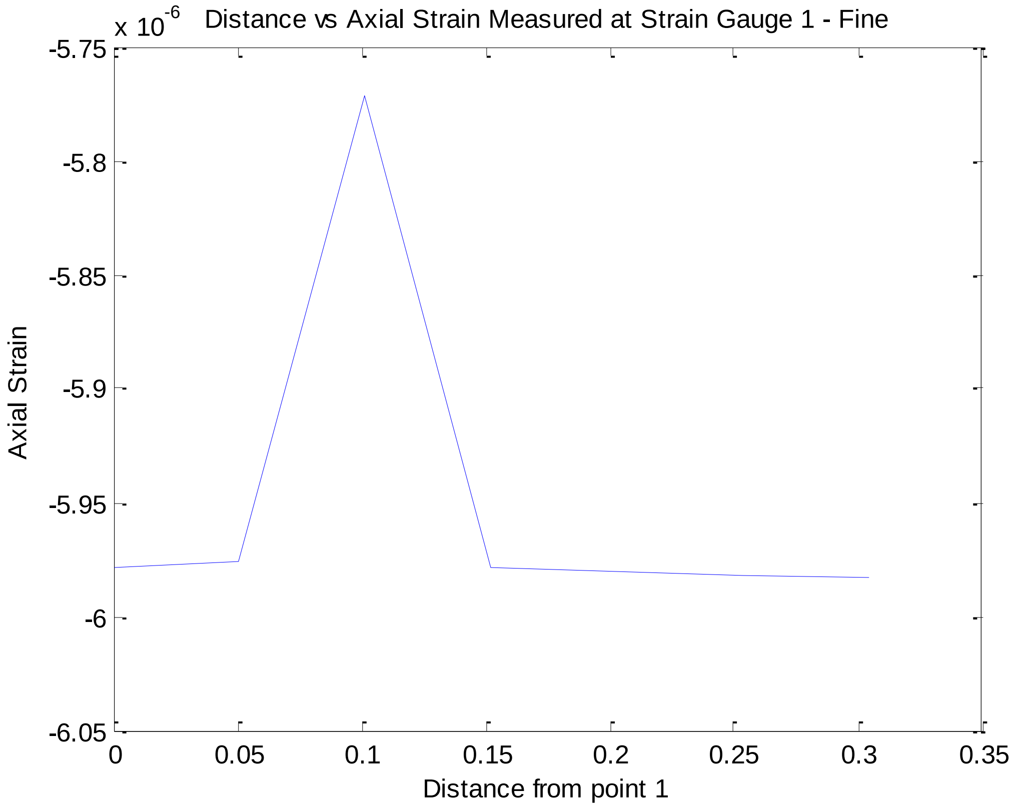


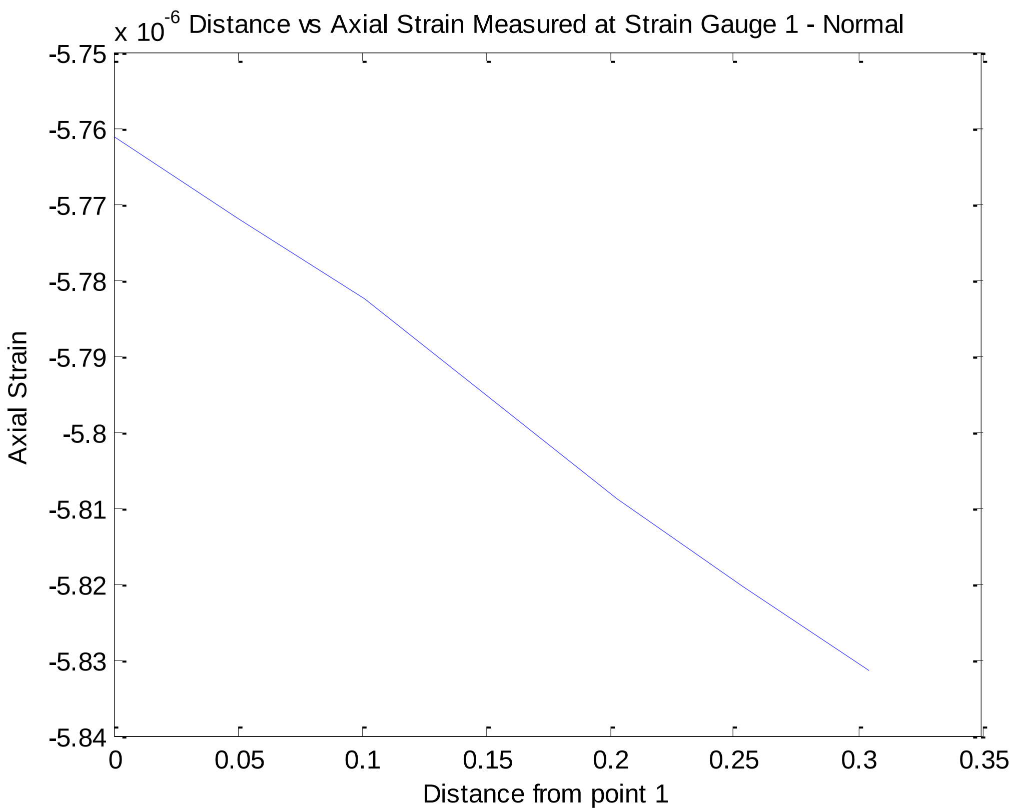
Total displacement (m): 3.36674e-4, 3.36628e-4
Strain tensor (global), xx component (1): -4.18558e-5, -5.89261e-5
Strain tensor (global), xy component (1): 8.34265e-5, 8.33791e-5

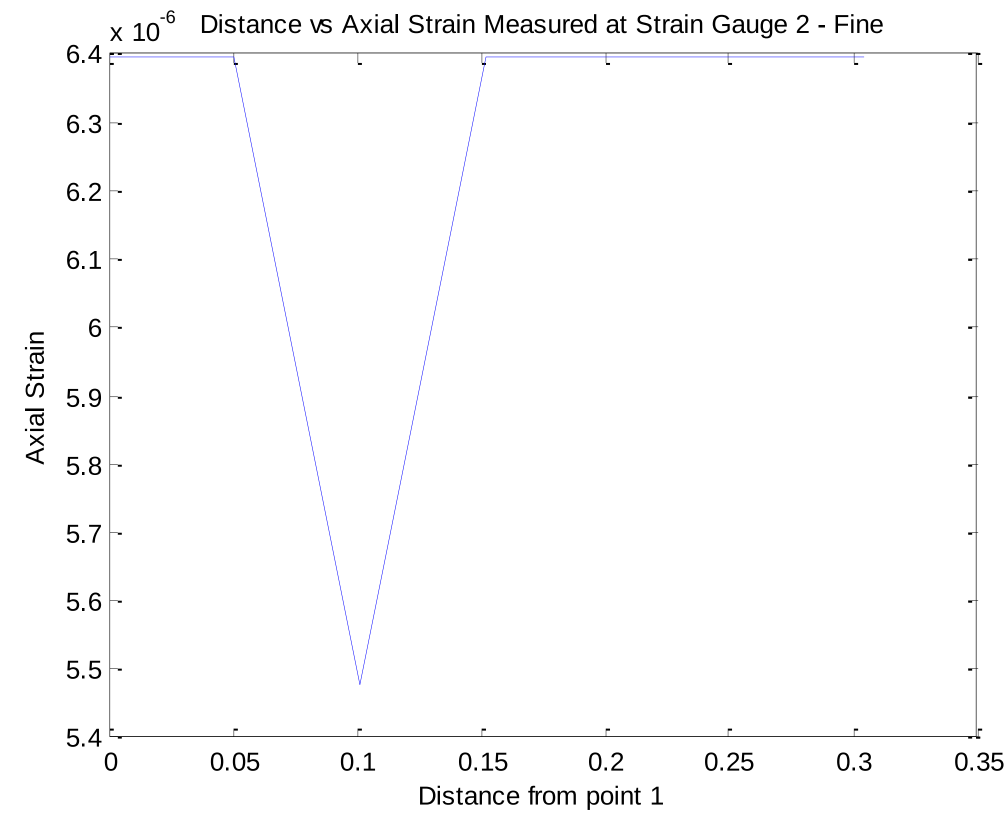
Using data from virtual strain gauges strategically placed on the model in Comsol, we can find the shear produced in each of the preceding loading cases and plot the data. This graphs shows shear stress plotted versus load location on the end of the wing. With this data, we can find the shear center of the wing simply by finding the x-intercept of this plot. The code that produced this and all of the remaining plots will be included in appendix 2.

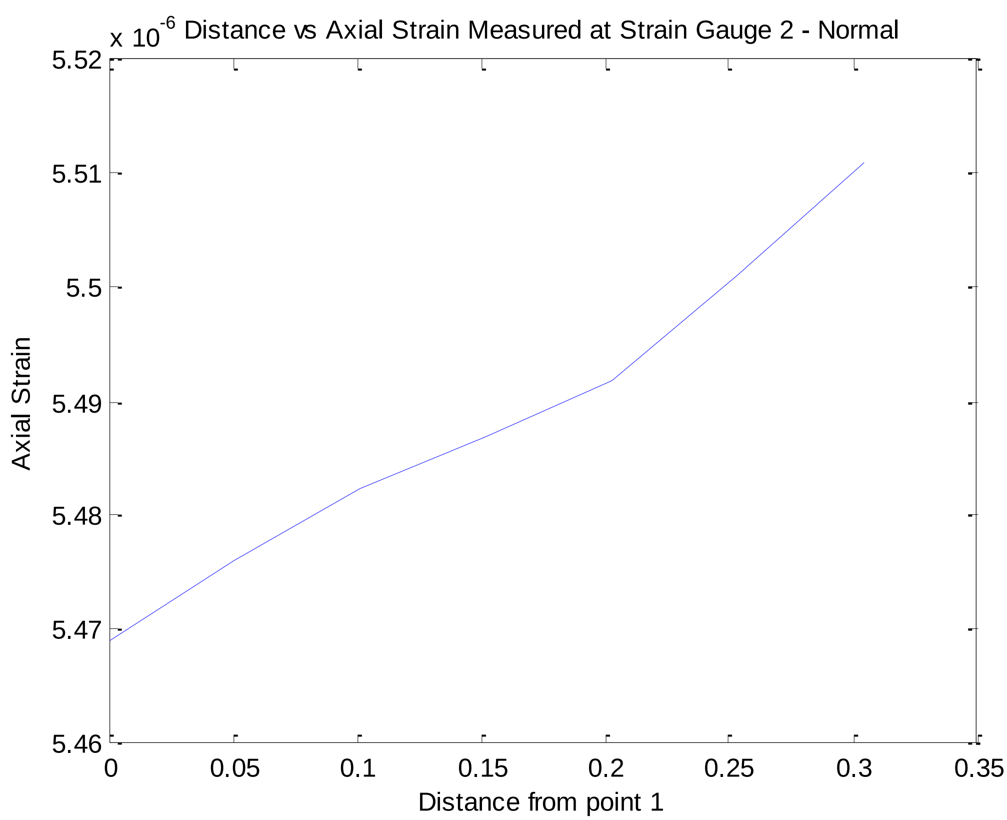


These plots show the axial stress distribution under the drag loading case for different mesh densities.

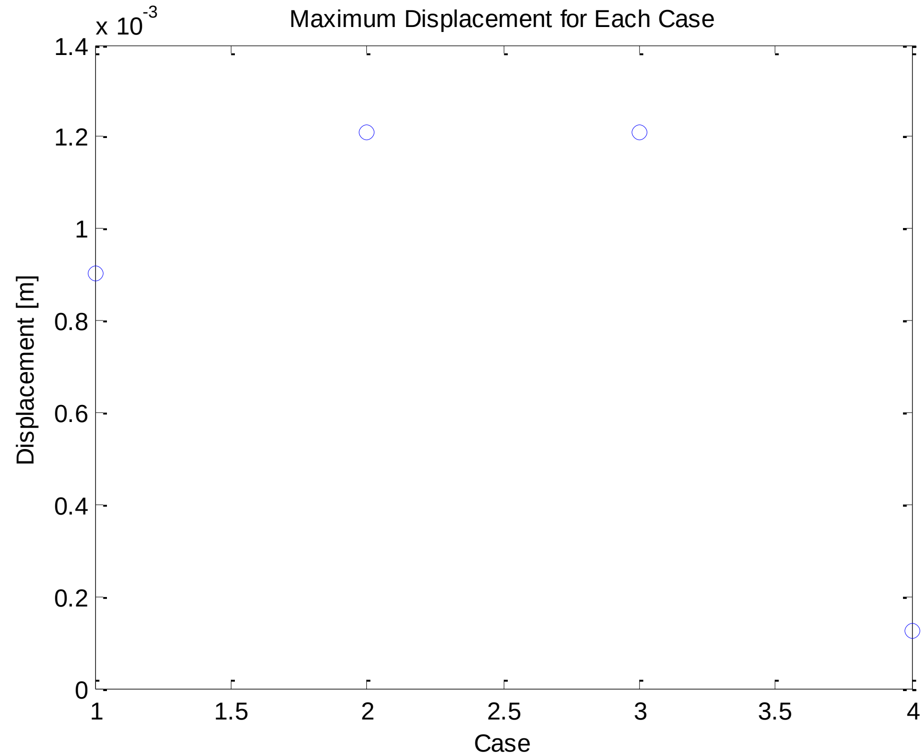


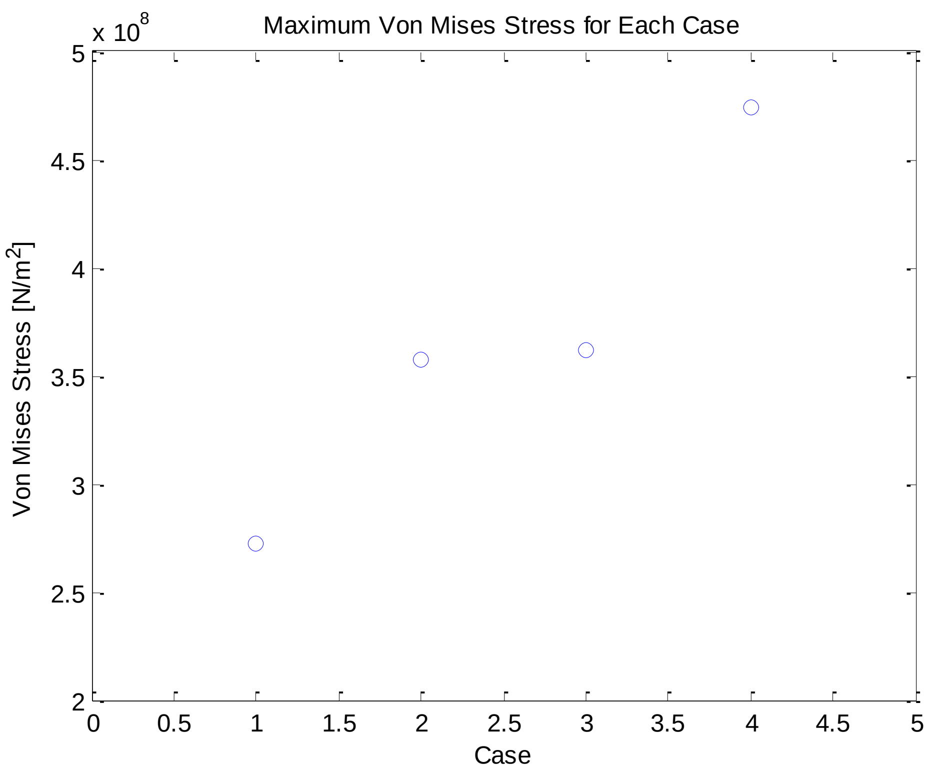



x 10^-6 Distance vs Axial Strain Measured at Strain Gauge 2 - Fine
Axial Strain
6.4
6.3
6.2
6.1
6
5.9
5.8
5.7
5.6
5.5
5.4
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Distance from point 1


x 10^-6 Distance vs Axial Strain Measured at Strain Gauge 2 - Normal
Axial Strain
5.52
5.51
5.5
5.49
5.48
5.47
5.46
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Distance from point 1

Finally, these plots show max displacement and stress for increasing mesh density. It is interesting to note how the values change with mesh density.





Discussion:

Both parts of this project, writing FEA code for beam problems and using fully developed, commercially available software to solve a problem came with their own set of challenges and nuances. The most difficult part of writing the beam code was allowing for the description of a beam in terms of sections. It was important for the code to be general enough to solve any beam problem with any number of loading conditions or boundary conditions, so the code had to be able to deal with all of that information, even at different resolutions with different numbers of sections.

Learning the basics of Comsol allowed this problem to be tackled without too many issues. The biggest challenge was correctly defining the wing geometry in the graphics window. After this was accomplished, the wing response could be calculated for any loading case. It is easy to imagine how useful FEA can be in a wide variety of structural design applications.

Conclusion:

Finite element analysis is an effective numerical method for solving differential equations for structural problems. Through writing code specifically for beam problems and using already developed, commercial level FEA software to solve a relevant problem, a greater understanding and appreciation for structural analysis, the finite element method, and computer science has been developed.

Refferences:

Kim, Nam H., and Bhavani V. Sankar. *Introduction to Finite Element Analysis and Design*. New York: John Wiley & Sons, 2009. Print.

http://en.wikipedia.org/wiki/Finite_element_method

Comsol instruction booklet

Test 2 equation sheet

Appendix I

```
%Jimmie Rush,Joshua Wheeler,Shane Chambry AE458 proj
%Author Jimmie Rush
clear;clc;close all;

%% user inputs
NumSec=5; %number of beam sections

E_Sec=[2 2 2 1 1];% E for each sections
I_Sec=[1 1 1 1 1]; %I for each section
L_Sec=[1 1 1 1 1]; %Length of each section


NDofCondAtSecPoint=[1,1,0,0,0,0,0,0,1,0,0,0]; %section point degree of freedom restrictions 1cond 0 dont: NumSecPoints*2

MagUnifLoadStart=[-5 0 0 0 0]; %Magnitude of uniform load at start of section
ConcLoad=[ 0 0 0 0 -10 0 0 20 0 0 0 0]; %concetrated loads/couples at each section point

NumElePerSec=[5,5,5,4,4]; %number of elements in eachc section
%%
NumSecPoints=NumSec+1; %number of section points
EI_Sec=E_Sec.*I_Sec;

Ne=0;
for i=1:1:NumSec
    L(i)=L_Sec(i)/NumElePerSec(i); %length of elements in section
    Ne=Ne+NumElePerSec(i); %total element count
end

SecCount=1;
EleCount=0;
for i=1:1:Ne
    EleCount=EleCount+1;
    EleSecId(i)=SecCount; %section identifier for each element
    if(EleCount==NumElePerSec(SecCount))
        SecCount=SecCount+1;
        EleCount=0;
    end
end


Nn=Ne+1; % number of total nodes;

NodeX(1)=0; %location of first node
for i=1:1:Ne;
    NodeX(i+1)=NodeX(i)+L(EleSecId(i)); %location of all nodes
end


for ele=1:1:Ne
    sec=EleSecId(ele);
    %k matrix for each element
```

```
    K{ele}=(EI_Sec(sec)/L(sec)^3)*[12 6*L(sec) -12 6*L(sec); 6*L(sec)...
    4*L(sec)^2 -(6*L(sec)) 2*L(sec)^2;-12 -6*L(sec) 12 -6*L(sec);6*L(sec)...
    2*L(sec)^2 -6*L(sec) 4*L(sec)^2];
    %distributed load for section
    p=MagUnifLoadStart(sec);
    %nodal forces for distributed load for each element
    Fdist{ele}=[p*L(sec)/2;p*L(sec)^2/12;p*L(sec)/2;-p*L(sec)^2/12];
end




Kglobal=zeros(Ne*2+2);
FdistGlobal=zeros(Ne*2+2,1);
x=1;y=1;j=1;
for j=1:1:Ne
Kglobal(x:x+3,y:y+3)=Kglobal(x:x+3,x:x+3)+K{j};

FdistGlobal(x:x+3)=FdistGlobal(x:x+3)+Fdist{j};

x=x+2;y=y+2;
end



%concentrated force on nodes
ConcForce=zeros(Nn*2,1);DofNodal=zeros(1,Nn*2);
j=1;
i=1;
for n=1:1:NumSec
    ConcForce(i:i+1,1)=ConcLoad(j:j+1); %nodal conc force matrix
    DofNodal(i:i+1)=NDofCondAtSecPoint(j:j+1); %nodal dof matrix
    i=i+NumElePerSec(n)*2;
    j=j+2;
end
ConcForce(i:i+1,1)=ConcLoad(j:j+1);
DofNodal(i:i+1)=NDofCondAtSecPoint(j:j+1);




ForceTotal=ConcForce+FdistGlobal;
%% condensing
N_RDOF=0;
count=1;
for i=1:Nn*2

    if DofNodal(i)==0
    Dof_cond(count)=i;
    count=count+1;
    end

    if DofNodal(i)==1
        N_RDOF=N_RDOF+1;
    end
end
```

```
Kcondensed=zeros(Nn*2-N_RDOF); %condensed stiffness matrix initialized
Fcondensed=zeros(Nn*2-N_RDOF,1); %condensed Force matrix initialized


for x=1:Nn*2-N_RDOF          %create force and stiffness condensed matrixes
        for y=1:Nn*2-N_RDOF
         Kcondensed(x,y)= Kglobal(Dof_cond(x),Dof_cond(y));
        end
    Fcondensed(x)=ForceTotal(Dof_cond(x));
end

Qc=Kcondensed\Fcondensed; % condensed displacement matrix

Qs=zeros(Nn*2,1);
count =1; %counter
for x=1:Nn*2   %create Structure degree of freedom matrix referencing non
condensed locations
   if ( any(x==Dof_cond(1,1:Nn*2-N_RDOF)));
   Qs(x)=Qc(count);
   count=count+1;
   end
end
%%
F_Gnodal=zeros(Ne*4,1);
i=1;j=4;
z=1;n=4;
for count=1:Ne  % create matrix of nodal sheer forces and moments for all
nodes seperatly
F_Gnodalstep(1:4,1)=K{count}*Qs(i:j,1);
F_Gnodalstep(1:2)=-F_Gnodalstep(1:2); %negate 1st 2 values
F_Gnodal(z:n,1)=F_Gnodalstep(1:4,1);
i=i+2;j=j+2;
z=z+4;n=n+4;
end

F_Gglobal(1:2,1)=F_Gnodal(1:2,1); % create global sheer force and moment
matrix for 1st and last node
F_Gglobal(Nn*2-1:Nn*2,1)=F_Gnodal(Ne*4-1:Ne*4,1);

if Ne>1 % create global sheer force and moment matrix for  nodes using
averages from previous node
    i=3;j=3;
    for count=1:Ne-1
        for count2=1:2
            F_Gglobal(j,1)=(F_Gnodal(i,1)+F_Gnodal(i+2,1))/2;
            i=i+1;
            j=j+1;
        end
        i=i+2;
    end
end

j=1;
for x=1:Nn
```

```
    displacementv(x)=Qs(j);
    theta(x)=Qs(j+1);
    shearforce(x)=F_Gglobal(j,1);
    Moment(x)=F_Gglobal(j+1,1);
    j=j+2;
end


figure
plot(NodeX,displacementv,'.-k')
title('Nodal Displacement vs Nodal distance')
xlabel('Nodal distance [m] ')
ylabel('Nodal Displacement[m] ')
grid on
set( gca, 'GridLineStyle', '-' );
grid minor

figure
plot(NodeX,theta,'.-G')
title('Nodal Displacement vs Nodal angle')
xlabel('Nodal distance [m] ')
ylabel('Nodal Angle[radians] ')
grid on
set( gca, 'GridLineStyle', '-' );
grid minor


figure
plot(NodeX,shearforce,'.-b')
title('Shear Force vs Nodal distance')
xlabel('Nodal distance [m] ')
ylabel('Shear Force[N] ')
grid on
set( gca, 'GridLineStyle', '-' );
grid minor


figure
plot(NodeX,Moment,'.-r')
title('Bending Moment vs Nodal distance')
xlabel('Nodal distance [m] ')
ylabel('Bending Moment[Nm] ')
grid on
set( gca, 'GridLineStyle', '-' );
grid minor
```

Appendix 2

```
clear;
clc;
%%Case A
Strain1=[-1.19307e-6 -8.89007e-7 -6.68779e-7 -2.80133e-7 1.78257e-8 3.12404e-
7 6.02816e-7];

Strain2=[-9.67824e-7 -6.68779e-7 -3.70139e-7 -7.03952e-8 2.28841e-7 5.26396e-
7 8.22676e-7];

w=.3048;
dist=0:(2*w/12):w;

%Shear center is where the graph crosses the x-axis

figure(1) %fine mesh
plot(dist,Strain1)
title('Shear Stress');
xlabel('location');
ylabel('Shear');
grid on


figure(2) %normal mesh
plot(dist,Strain2)
grid on


axialTop1=[-5.97883e-6 -5.97544e-6 -5.77188e-6 -5.97811e-6 -5.9802e-6 -
5.98193e-6 -5.9825e-6];

axialBottom1=[6.39622e-6 6.3953e-6 5.47604e-6 6.39446e-6 6.39588e-6 6.39666e-
6 6.3946e-6];

axialTop2=[-5.76127e-6 -5.77188e-6 -5.78263e-6 -5.79569e-6 -5.80885e-6 -
5.82037e-6 -5.83152e-6];

axialBottom2=[5.46903e-6 5.47604e-6 5.48229e-6 5.48693e-6 5.49184e-6
5.50109e-6 5.51094e-6];

figure(3)
plot(dist,axialTop1)
xlabel('Distance from point 1')
ylabel('Axial Strain')
title('Distance vs Axial Strain Measured at Strain Gauge 1 - Fine')

figure(4)
plot(dist,axialTop2)
xlabel('Distance from point 1')
ylabel('Axial Strain')
title('Distance vs Axial Strain Measured at Strain Gauge 1 - Normal')

figure(5)
```

```
plot(dist,axialBottom1)
xlabel('Distance from point 1')
ylabel('Axial Strain')
title('Distance vs Axial Strain Measured at Strain Gauge 2 - Fine')

figure(6)
plot(dist,axialBottom2)
xlabel('Distance from point 1')
ylabel('Axial Strain')
title('Distance vs Axial Strain Measured at Strain Gauge 2 - Normal')

%%Case B

cases=[1:1:4];
disp=[9.0312e-4 1.20814e-3 1.2081e-3 1.2611e-4];
mis=[2.7248e8 3.5749e8 3.6195e8 4.7395e8];

figure(7)
plot(cases,disp,'bo')
xlabel('Case')
ylabel('Displacement [m]')
title('Maximum Displacement for Each Case')

figure(8)
plot(cases,mis,'bo')
xlabel('Case')
ylabel('Von Mises Stress [N/m^2]')
title('Maximum Von Mises Stress for Each Case')
axis ([0 5 2e8 5e8])
```