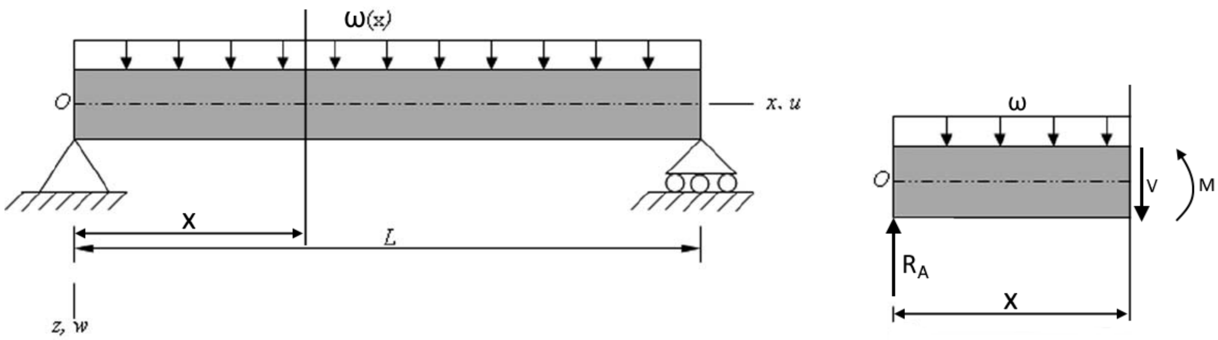
# Finite Element Methods for Solids and Structures 4 Project: Modelling of Shallow and Deep Beams

Alasdair Christison Gray*
*University of Edinburgh*

## I. Question 1

Using elementary beam theory the bending moment can be derived by taking a cut through the beam and applying translational and rotational equilibrium, as shown in figure 1.




**a) Simply supported beam with uniformly distributed load**

**b) Cut through beam at x showing shear force and bending moment**

**Figure 1. Calculating the shear force and bending moment distributions**

It can easily be shown that the vertical reactions are $R_A = R_B = \frac{\omega L}{2}$. Balancing the vertical forces on the beam section gives:

$$R_A - \omega x - V = 0 \rightarrow V = R_A - \omega x = \omega\left(\frac{L}{2} - x\right) \tag{1}$$

And equation moments around A gives:

$$M - Vx - \frac{\omega x^2}{2} = 0 \rightarrow M = \frac{x}{2}(2V + \omega x) = \frac{\omega x}{2}(L - x) \tag{2}$$

Figure 2 shows the shear force and bending moment distributions for the thick and thin beam problems.

## II. Question 2

Euler Bernoulli beam theory allows us to calculate the deflected shape of a beam using the relationship:

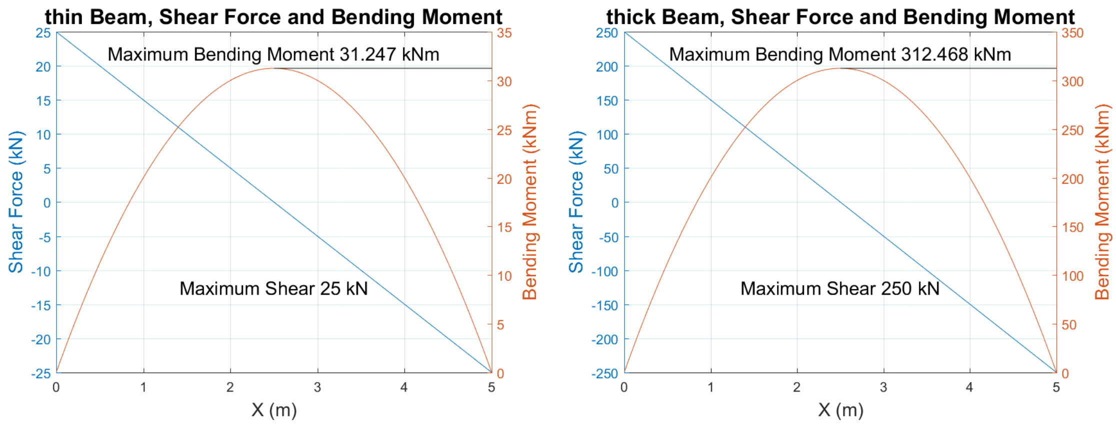
$$M = EI\frac{d\phi}{dx} \rightarrow \frac{d^2\delta}{dx^2} = \frac{M}{EI} \tag{3}$$

Integrating twice and applying the boundary conditions $\delta(0) = \delta(L) = 0$ gives the deflected shape:

$$\delta(x) = \frac{\omega}{24EI}(2Lx^2 - x^3 - L^3) \tag{4}$$

$$\delta_{\text{Max}} = \delta\left(\frac{L}{2}\right) = \frac{5\omega L^4}{384EI} \tag{5}$$

*MEng Student, Department of Mechanical Engineering, A.Gray-17@sms.ed.ac.uk




**a) Thin beam** **b) Thick beam**

**Figure 2. Shear force and bending moment diagrams for the thin and thick beams**

To account for shear deformations we use the Timoshenko beam equations:

$$M = EI\frac{d\phi}{dx} \rightarrow \frac{d\phi}{dx} = \frac{M}{EI} \tag{6}$$

$$V = \frac{AG}{f_s}\left(\frac{d\delta}{dx} - \phi\right) \rightarrow \frac{d\delta}{dx} = \phi - \frac{Vf_s}{AG} \tag{7}$$

Where $f_s$ is a factor which accounts for the non-uniform distribution of shear stress across the beam section, for a rectangular beam it can be shown that:

$$f_s = \frac{12 + 11\nu}{10(1+\nu)} \approx \frac{6}{5} \tag{8}$$

Again, substituting in the expressions for $V$ and $M$ and applying the same boundary conditions as above gives:

$$\delta(x) = \frac{\omega}{24EI}(2Lx^2 - x^3 - L^3) - \frac{\omega f_s x}{2AG}(L - x) \tag{9}$$

$$\delta_{\text{Max}} = \delta\left(\frac{L}{2}\right) = \frac{5\omega L^4}{384EI}\left(1 + \frac{48f_sEI}{5GAL^2}\right) \tag{10}$$

Figure 3 shows the stress and both deflection distributions and the maximum values of each for the thin and thick beams, the material. The typical elastic properties of steel, $E = 200$ GPa, $\nu = 0.3$, are used both for these theoretical calculations and for all the finite element models presented in this project.

# III. Question 3

Given the very similar nature of the 12 simulations to be run using beam elements, a python script was written to generate the input files based on a template, the script also wrote a batch file to run all the simulations. The input file for the thick beam, ten B22 element model is included in appendix A and the python scripts in appendix B.

# IV. Question 4/5

Table 1 presents the maximum stress and deflection values for each model along with the absolute and relative errors in each value. Figure 5 shows the stress distribution for each model and figure 4 shows the deflected shapes all model values for displacement and stress are taken at nodes, because the stress and bending moment are directly proportional ($\sigma_{Max} = \frac{Md}{2I}$) the bending stress values were taken as a good indicator of the models' ability to predict bending moments.

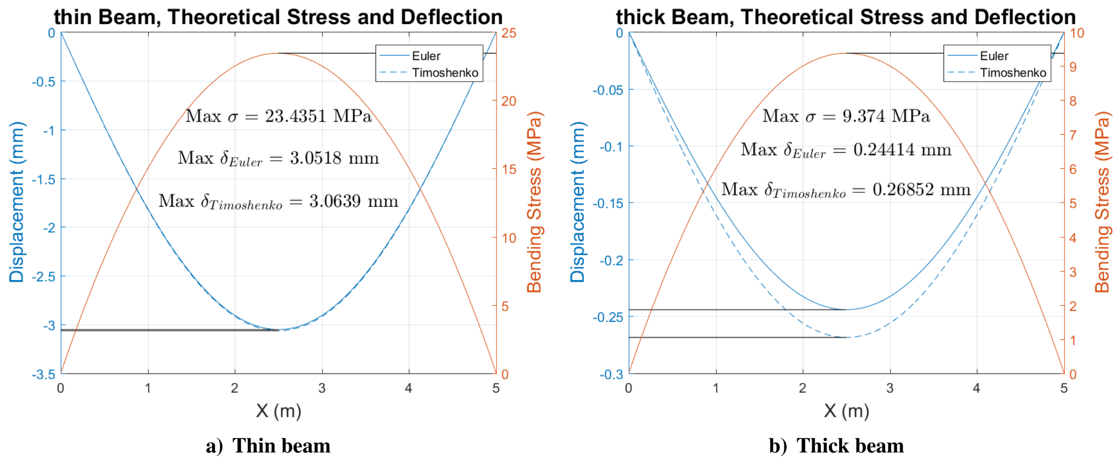



**Figure 3. Theoretical stress and deflection distributions for the thin and thick beams**

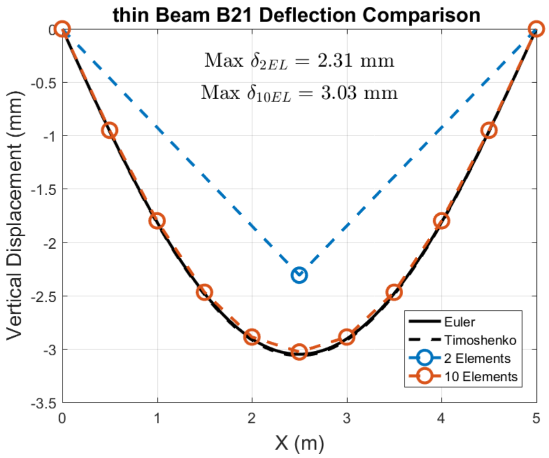
**Table 1. Stress and deflection results from beam models**

| | | **B21** | | **B22** | | **B23** | |
|---|---|---|---|---|---|---|---|
| | | **2 Elements** | **10 Elements** | **2 Elements** | **10 Elements** | **2 Elements** | **10 Elements** |
| **Thin Beam** | Max $\sigma$ (MPa) | 11.7 | 23.0 | 27.3 | 23.6 | 27.3 | 23.6 |
| | $\Delta\sigma$ | -11.7 | -0.4 | 3.9 | 0.2 | 3.9 | 0.2 |
| | $\Delta\sigma$ % | -50.07 | -1.86 | 16.49 | 0.70 | 16.49 | 0.70 |
| | Max $\delta$ (mm) | 2.31 | 3.03 | 3.06 | 3.06 | 3.05 | 3.05 |
| | $\Delta\delta_E$ | -0.742 | -0.022 | 0.008 | 0.008 | -0.002 | -0.002 |
| | $\Delta\delta_E$ % | -24.3 | -0.714 | 0.269 | 0.269 | -0.059 | -0.059 |
| | $\Delta\delta_T$ | -0.754 | -0.034 | -0.004 | -0.004 | -0.014 | -0.014 |
| | $\Delta\delta_T$ % | -24.6 | -1.11 | -0.127 | -0.127 | -0.454 | -0.454 |
| **Thick Beam** | Max $\sigma$ (MPa) | 4.69 | 9.19 | 10.90 | 9.44 | 10.90 | 9.44 |
| | $\Delta\sigma$ | -4.68 | -0.184 | 1.53 | 0.066 | 1.53 | 0.066 |
| | $\Delta\sigma$ % | -50.0 | -1.96 | 16.3 | 0.704 | 16.3 | 0.704 |
| | Max $\delta$ (mm) | 0.208 | 0.266 | 0.268 | 0.268 | 0.244 | 0.244 |
| | $\Delta\delta_E$ | -0.036 | 0.022 | 0.024 | 0.024 | 0.000 | 0.000 |
| | $\Delta\delta_E$ % | -14.8 | 9.0 | 9.8 | 9.8 | -0.057 | -0.057 |
| | $\Delta\delta_T$ | -0.061 | -0.003 | -0.001 | -0.001 | -0.025 | -0.025 |
| | $\Delta\delta_T$ % | -22.5 | -0.938 | -0.194 | -0.194 | -9.13 | -9.13 |

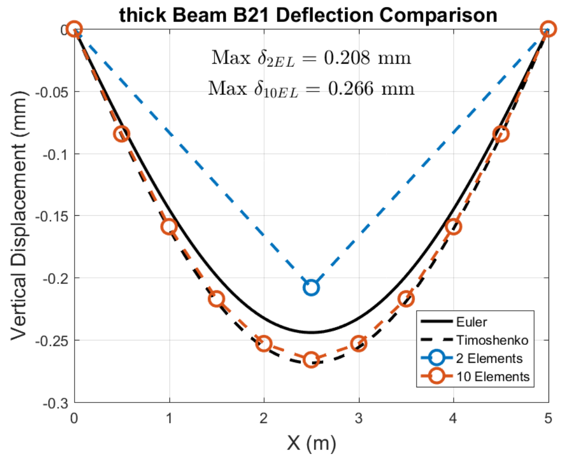
### A. Predicting Displacement

In both the thick and thin beams, the two element B21 model significantly under-predicts the beam deflection. This is because the two linear elements cannot accurately capture the deflected shape of the real beam which, as shown in equation 9, is cubic. The calculated linear variation of deflection and slope over each semi-span of the beam leads to a much stiffer deflection mode and thus a smaller deflection. Whilst the ten element model is much closer to the correct value it is still under the theoretical value by approximately 1% in both the thick and thin cases. Because the B21 is a Timoshenko element it correctly predicts the increased deflection in the thick beam compared to the Euler-Bernoulli theory.

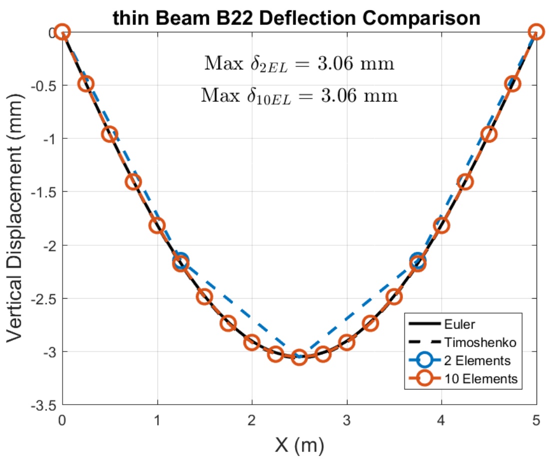
The B23 element does a much better job at predicting the deflection of the thin beam, due to its cubic shape functions, with the two and ten element models predicting exactly the same deflection which is within 0.1% of the



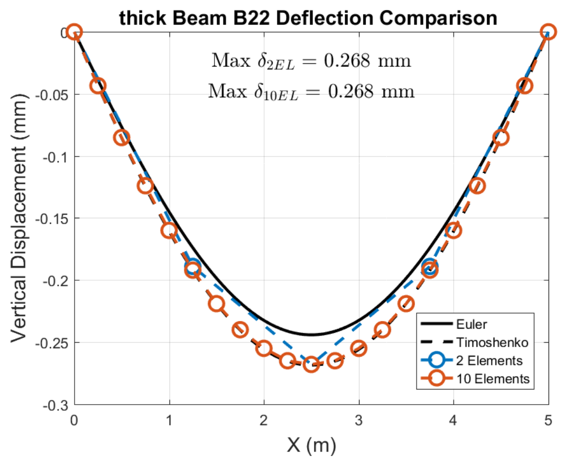

**a) Thin Beam B21 Elements**



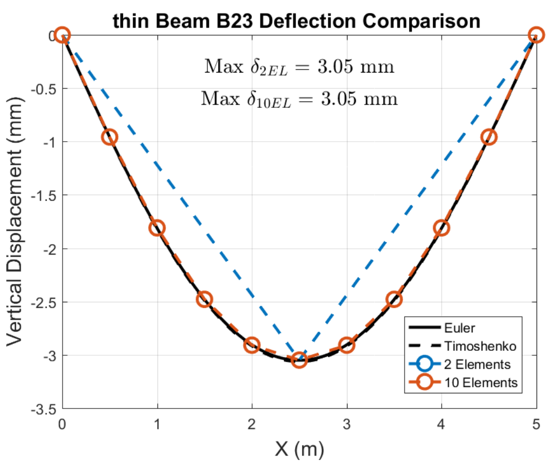

**b) Thick Beam B21 Elements**



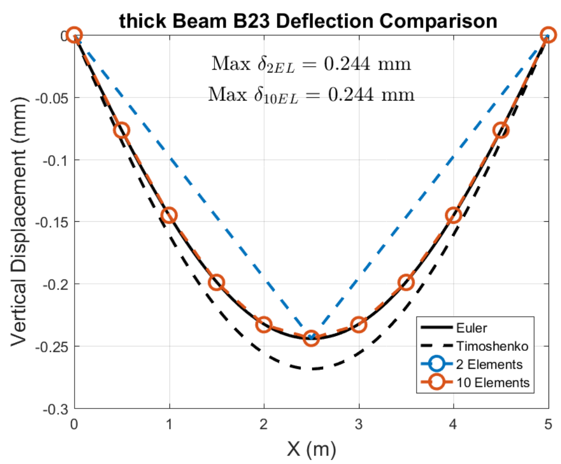

**c) Thin Beam B22 Elements**




**d) Thick Beam B22 Elements**



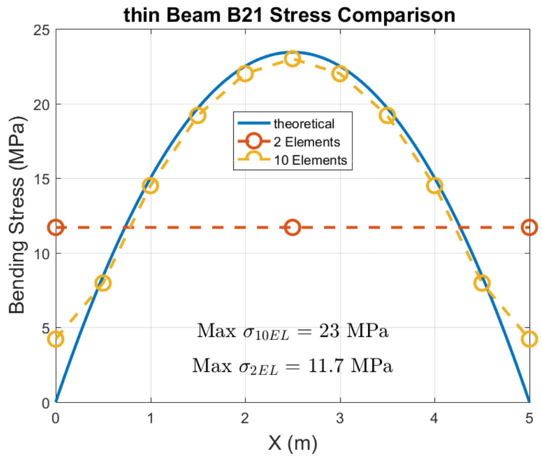

**e) Thin Beam B23 Elements**



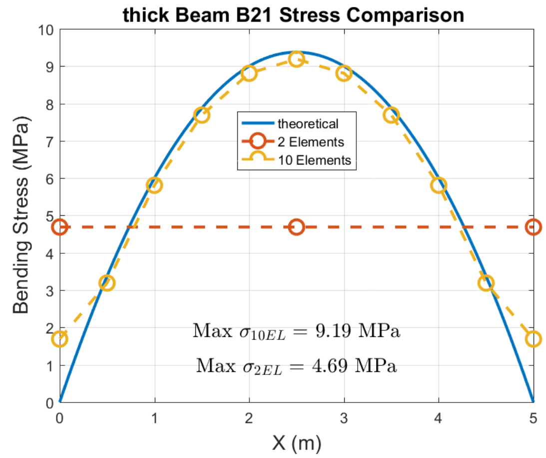

**f) Thick Beam B23 Elements**

**Figure 4. Beam Model Deflected Shapes**

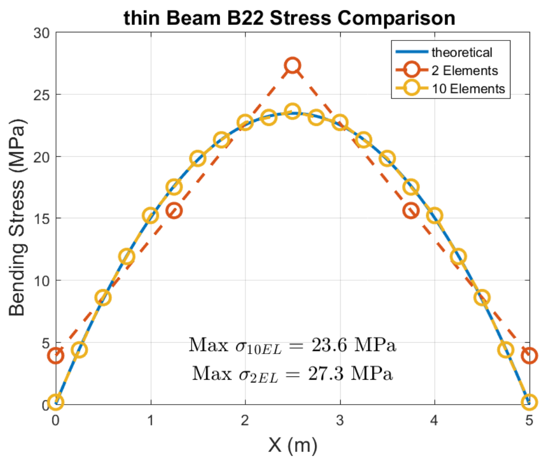
theory. However, because the element uses an Euler-Bernoulli formulation it significantly under-predicts the deflection of the thick beam. Because this error stems from the bending equations used by the element, it cannot be overcome with finer discretisation.



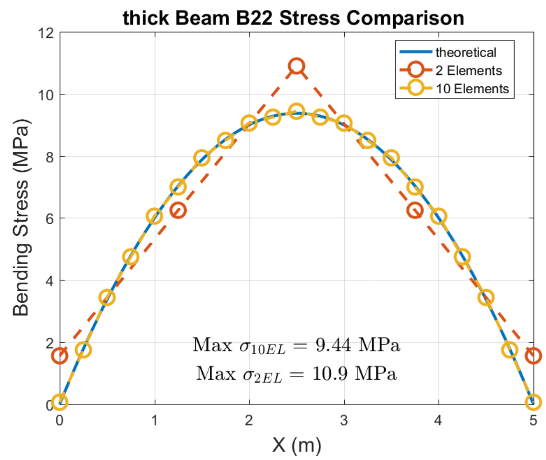

**a) Thin Beam B21 Elements**



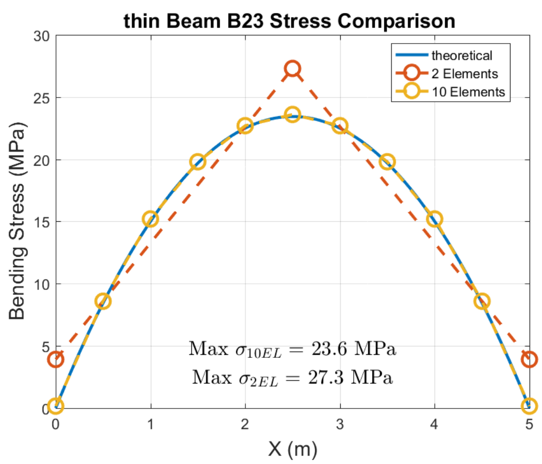

**b) Thick Beam B21 Elements**



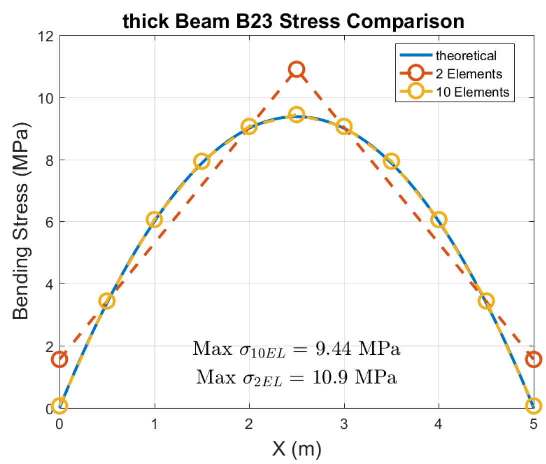

**c) Thin Beam B22 Elements**




**d) Thick Beam B22 Elements**




**e) Thin Beam B23 Elements**




**f) Thick Beam B23 Elements**

**Figure 5. Beam Model Stress Distributions**

The B22 elements are able to to accurately predict the deflection of both beams to within 0.2%. As with the B23 elements, the two and ten element meshes give the same deflection values, indicating that the calculated displacement is mesh independent for this relatively simple loading case. Similarly to the B21, the B22's Timoshenko formulation allows it to accurately predict the thick beam deflection, unlike the B23 element.

### B. Predicting Stresses

As with the deflection, the stresses calculated by the two B21 element model are significantly different from the theoretical values, in shape as well as magnitude. These errors stem from the linear nature of the element. Because displacements are interpolated linearly along the element, the strain, and therefore stress, which is proportional to the displacement derivative ($\epsilon = Lu$) is constant over the element. The stress is computed at the integration points which for the two B21 elements will be at the quarter and three quarter span of the beam, because the model is symmetric these two points have the same stress. This means that the stress values interpolated onto the nodes using the integration point values are the same across the entire beam leading to the flat lines in figures 5a and 5b. The stress at the integration points is under-predicted due to reduced displacement mentioned in the previous section. The ten element model is able to predict the stress distribution more accurately, under-predicting the peak stress by less than 2% in both the thick and thin beams. This because the finer discretisation allows the linear shape functions to better approximate the cubic displacement curve, consequently the strains and stresses computed from the nodal displacements are closer to the real values. Additionally the increased density of integration points means the stresses are better interpolated onto the nodes. The non-zero stress shown at the two ends of the beam is an artefact of this stress interpolation.

Both the B22 and B23 elements predict the bending stress with exactly the same accuracy, with two elements the models over-predict the peak stress in the thin and thick beams by 3.9 and 1.53% respectively, with ten elements the values were 0.2 and 0.066%. This indicates that, unlike displacement, the computed stress is not mesh independent for this load case. Because these elements are higher order, they do not exhibit constant stress along each element. This means that even with only two elements there is a decrease in stress towards the beam ends although the end stress is still non-zero. Even in the ten element models the end stress is non-zero but is an order of magnitude lower than the values along the rest of the beam. It is interesting to note that even though the two element B22 model has nodes at the quarter and three quarter points of the span it has still produced a linear stress curve along each half of the beam. As mentioned earlier, this is because the stress is derived from the derivative of displacement ($\sigma = D\epsilon = DLu$) and so the quadratic shape functions used for displacement in the B22 element give linear distributions of stress.

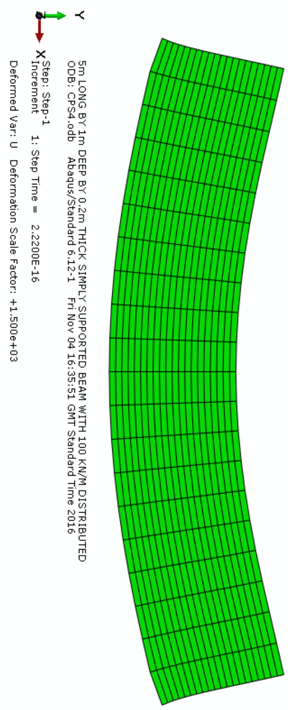
## V. Question 6

It is clear from the results above that the B21 element is not a good choice for modelling beams, it requires a much finer mesh than the other two elements to even come close to the correct values of stress and displacement. In cases where a coarse mesh is used it severely under-predicts both stress and displacement and can give an unrealistic stress distribution.

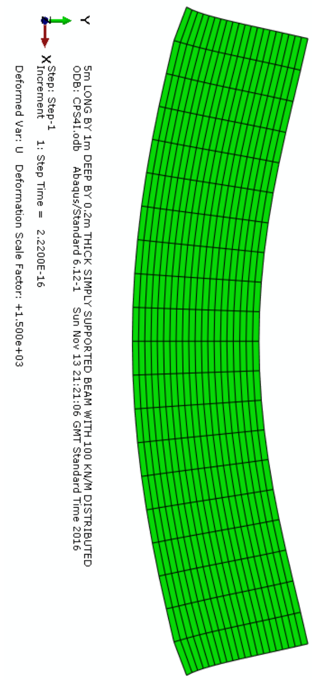
The B22 and B23 elements have both been shown to give almost perfectly accurate displacement predictions relative to their bending models, even with an extremely coarse, two element mesh. They also predict exactly the same stress distribution for the same number of elements. Given that for slender beams, the difference between the Euler-Bernoulli and Timoshenko displacements is small, there will be little difference in the results produced by B22 and B23 models with the same number of elements. However when using the B22 elements, twice as many nodes are required for the same number of elements, this means the model has double the number of degrees of freedom. The size of all computation matrices is also doubled and therefore the computational cost of solving the system of equations and the memory required to store the simulation data is also greater. The B23 therefore offers the best option for modelling in plane slender beams with minimal computational cost. In reality, the one dimensional nature of beam elements means that even complex beam models rarely take more than a few minutes to run, even on a single consumer grade processor. In cases where the aspect ratio of any beams in a model are not clearly slender ($L/d \leqslant 10$) the B22 is therefore a better choice.

## VI. Question 7

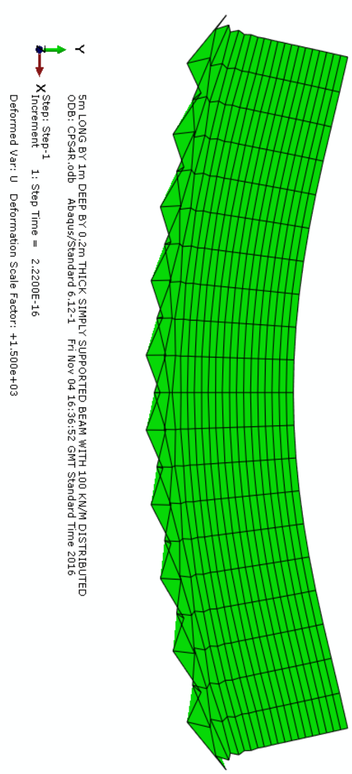
The input file for the CPS4I model is included in appendix A. Figure 6 shows the deformed shape of the four plane stress models with a scaling factor of 1500. The CPS4R model shows a clear hourglass deformation pattern along the bottom edge of the beam see figure 7a.



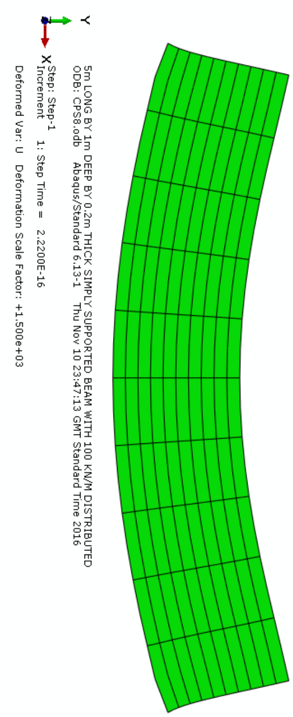

**a) CPS4**




**b) CPS4I**



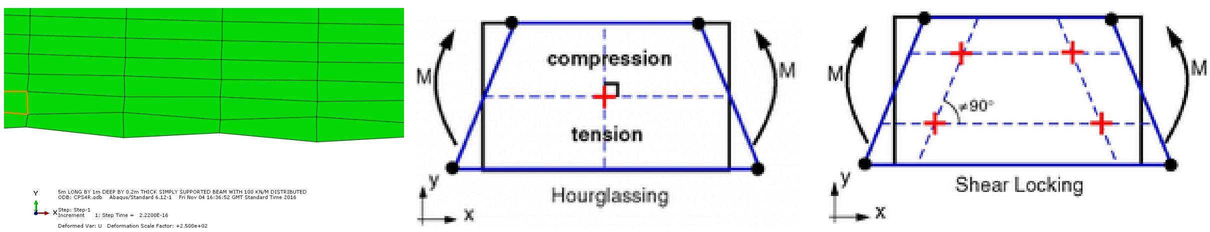

**c) CPS4R**




**d) CPS8**

**Figure 6. Deformed shape of plane stress models, scaling factor 1500**

The hourglass effect occurs because the reduced order integration in the element is unable to detect the strains associated with the hourglass deformation mode, this can be understood by noticing that the length of the dashed lines in figure 7b does not change in the hourglass deformation mode, unlike in the fully integrated element in figure 7c. These spurious deformations lead to inaccurate predictions of both displacement and stress throughout the model. Fully integrated elements exhibit a different problem known as "shear locking" in pure bending, this is because the deformed shape in this loading mode (see figure 7c) does not accurately represent the shape of a material, instead exhibiting significantly more shear strain which gives a much stiffer element. To remedy this problem it is necessary to use multiple elements across the depth of the beam, using higher order elements which can capture the curvature of the top and bottom edges of the element or to using elements with incompatible modes, such as the CPS4I. These elements incorporate additional strains from the incompatible deformation modes and thus avoid the shear locking problem.




**a) Hourglassing on the CPS4R model** **b) Hourglass deformation mode in a reduced integration element** **c) Hourglass deformation mode in a fully integrated element**

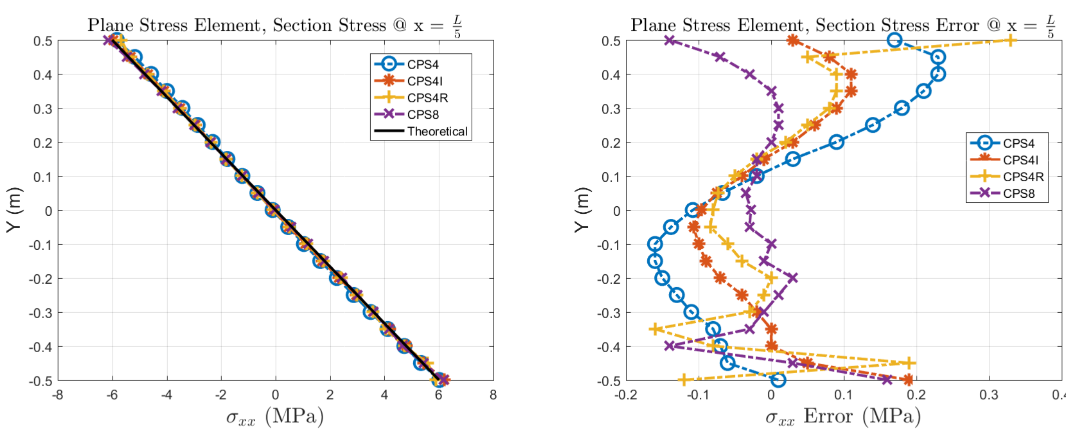
**Figure 7. The hourglass phenomenon**

# VII. Question 8

The bending stress distribution through the cut section is calculated using:

$$\sigma_{xx} = \frac{M(L/5)y}{I} \quad \text{where} \quad M(L/5) = \frac{\omega L}{10}\left(L - \frac{L}{5}\right) = \frac{2\omega L^2}{25} \tag{11}$$

Looking at figure 8a all the models appear to have the correct magnitude and linear shape. The error plot in figure 8b shows that all models are within 0.2 MPa of the theoretical distribution over the majority of the section indicating good agreement with the theory.




**a) $\sigma_{xx}$ Distributions** **b) $\sigma_{xx}$ Error**

**Figure 8. Bending stress distribution through section at $x = \frac{L}{5}$**

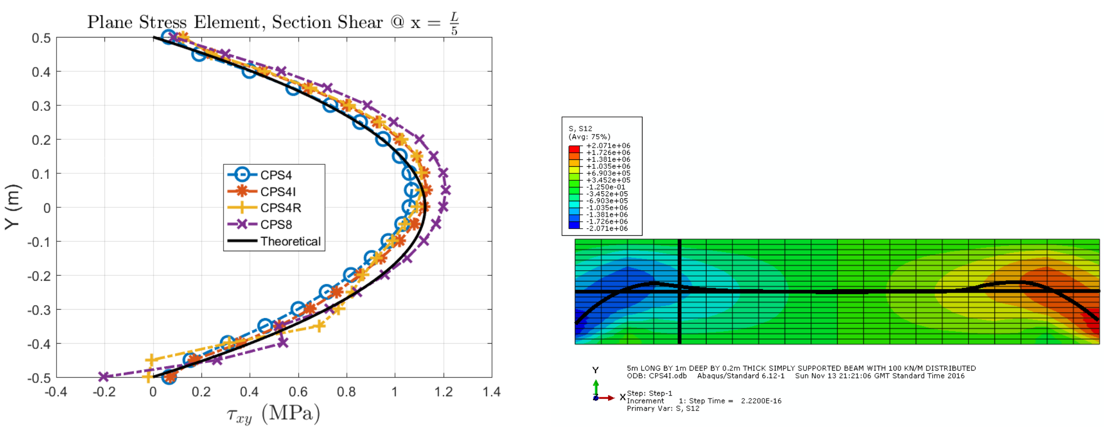
## VIII. Question 9

The shear stress distribution through the beam for a rectangular section is given by:

$$\tau_{xy}(y) = \frac{V(L/5)}{A}\left(1 - \frac{y^2}{(d/2)^2}\right) \text{ where, } V\left(\frac{L}{5}\right) = \omega\left(\frac{L}{2} - \frac{L}{5}\right) = \frac{3\omega L}{10} \tag{12}$$

**Table 2. Shear stress distribution results**

| | Theoretical | CPS4 | CPS4I | CPS4R | CPS8 |
|---|---|---|---|---|---|
| $\tau_{mean}$ | 0.742 | 0.668 | 0.719 | 0.717 | 0.780 |
| $\Delta\tau_{mean}$ | | -0.0746 | -0.0232 | -0.0258 | 0.0374 |
| $\tau_{max}$ | 1.113 | 1.07 | 1.13 | 1.11 | 1.21 |
| $\Delta\tau_{max}$ | | -0.0436 | 0.0163 | -0.0036 | 0.0963 |
| $\frac{\tau_{max}}{\tau_{mean}}$ | 1.5 | 1.602 | 1.571 | 1.549 | 1.552 |
| $\Delta\frac{\tau_{max}}{\tau_{mean}}$ | | 0.102 | 0.071 | 0.049 | 0.052 |




**a) Shear stress distribution through section at $x = \frac{L}{5}$**

**b) Shear stress distribution in the CPS4I model**

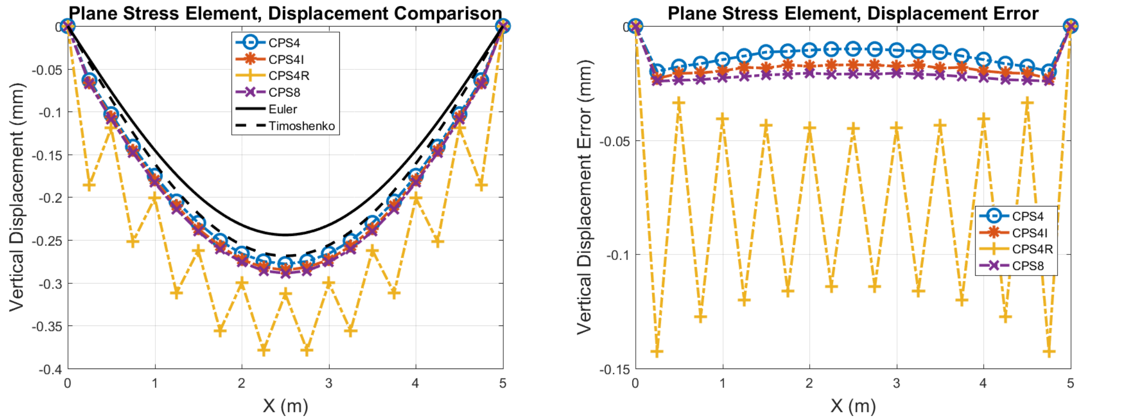
**Figure 9. Shear stress distribution in plane stress model**

As shown in table 2 and figure 9a all the models show the correct parabolic shape in the shear stress distribution but the distributions also show an upwards shift with the peak values occurring above the section midpoint and have non-zero shear stresses at the top and bottom of the section. The has the highest mean and peak shear stress most likely because it is quadratic and so does not exhibit shear locking.

Figure 9b shows how the approximate path of the peak shear stress through the beam section along the beam length along with the location of the plotted section and the neutral axis, at the supports the peak value occurs close to the fixed nodes at the bottom of the section before moving above the neutral axis at the cut section and returning the the mid plane towards the middle of the span.

## IX. Question 10

The vertical displacement along the bottom edge of the plane stress beam models and the discrepancies between the simulation and Timoshenko theory values are shown in figure 10. All of the models predict greater vertical displacements than the Theory by approximately 10%, the effect of hourglassing on the CPS4R displacement values is clearly visible by the jagged deflection curve. The discrepancy distribution in figure 10b shows that the majority of the discrepancy growth occurs over the first element at either side of the beam, these sections see the highest shear forces which leads the beam section to warp (see figure 11). Timoshenko beam theory assumes that beam sections remain planar and so does not predict the additional deflection due to this section shear warping. This may explain why the CPS8 model which showed the highest shear stresses also shows the greatest displacement values (ignoring the CPS4R hourglassing values).




**a) Deflection along bottom edge of plane stress beam models** **b) Difference between Timoshenko and simulation deflection values**

**Figure 10. Plane stress element beam displacement**

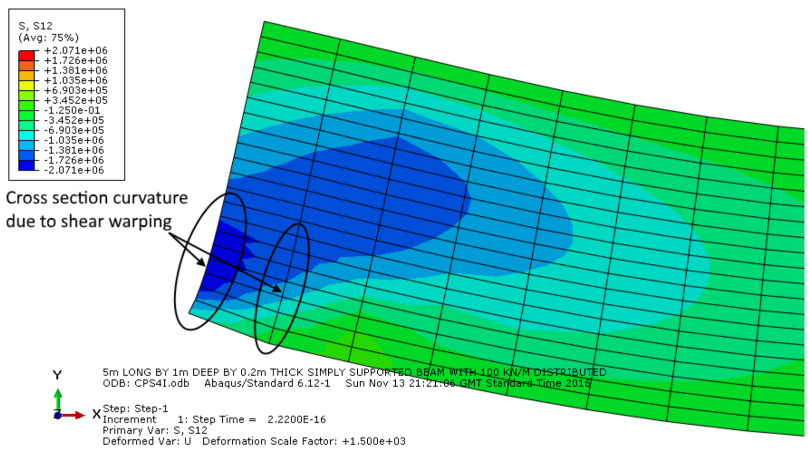



**Figure 11. Shear warping near end support in CPS4I model**

# X. Question 11

The B22 models matched very well with the Timoshenko theory values so the difference between these and the plane stress elements is similar to that described in the previous section. The plane stress elements show greater deflection due to the shear warping which is not captured by the Timoshenko formulation of the B22 elements. The B23 is significantly further off the plane stress values as it uses an Euler-Bernoulli formulation which, as described earlier, under-predicts the displacement of thick beams.

Based on these results, the CPS4I element seems the best choice from the various plane stress elements as it performs best overall in predicting bending stress, shear stress and deflection, the CPS8 element predicts the bending stress marginally better but significantly overestimates the shear stress and thus exhibits more shear warping and greater deformation. The CPS4R predicts the bending and shear stresses well in the sections plotted above, but hourglassing leads to non-physical displacements and stresses elsewhere in the model (see figures 6c and 12). Given that both the beam and plane stress elements showed almost identical stresses that agreed well with the theoretical predictions, using beam elements would be the better choice if the user is primarily interested in stress values However, if the design of the structure being studied is instead primarily constrained by displacement then using the plane stress elements is a better option.

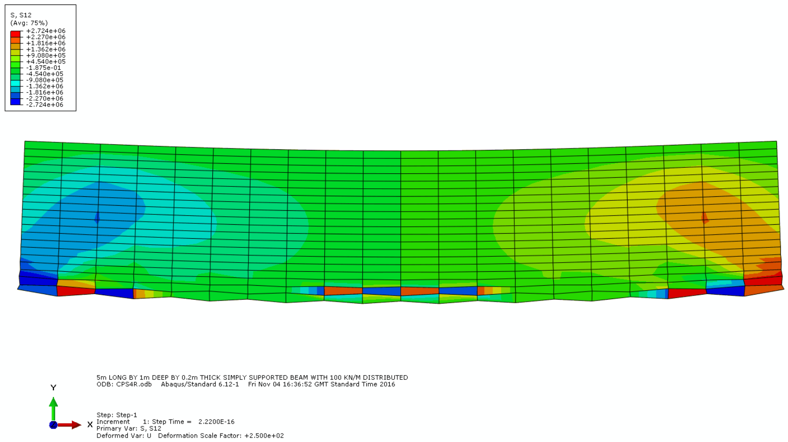
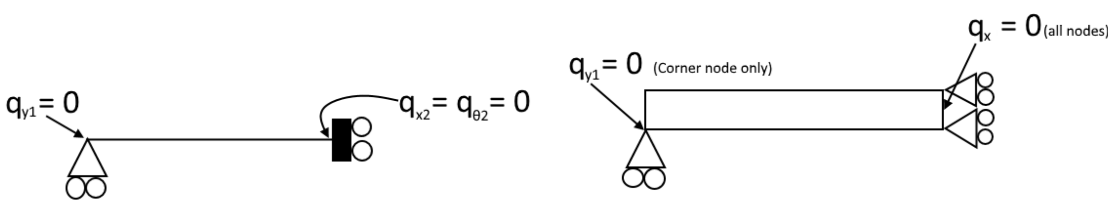



**Figure 12. Non-physical shear stress distribution in the CPS4R model**

## XI. Question 12

Because the beam geometry, loading and boundary conditions are symmetric, all the models in this project could be modelled more efficiently using symmetry conditions. This would be done by halving the length of the beam and applying a symmetry boundary condition at one of the ends. For the beam models, one end would be fixed in the y direction representing a simple support and the other end would be fixed in the axial direction ($q_x = 0$) and in the rotational direction ($q_\theta = 0$). For the plane stress element model, one corner node would be fixed in y and fixing the x displacement of the nodes on the other end would lock both the axial and rotational deformation. Schematics of the required boundary conditions are shown in figures 13a and 13b.




**a) For beam elements**

**b) For 2D elements**

**Figure 13. Symmetry boundary conditions**

## A. Input Files

### A. B22_thick_10EL.inp



```
*****************************************************************
** FEMSS4 Project                                              **
** Alasdair Gray, S1208454                                     **
** Beam model input file generated by my wonderful python script **
*****************************************************************
*HEADING
5m LONG SIMPLY SUPPORTED thick BEAM WITH -100000 kN/m DISTRIBUTED LOAD, 10
ELEMENT
B22 MESH
******************************
*NODE
** DEFINE LOCATION OF END NODES
1, 0.0, 0.0
21, 5.0, 0.0
******************************
*NGEN, NSET=NALL
** GENERATE NODES ALONG LENGTH AND DEFINE NODE SET 'NALL'
1, 21, 1
******************************
**DEFINE THE FIRST 'MASTER ELEMENT' NUMBER 1 AS A B22
*ELEMENT, TYPE=B22
1, 1, 2, 3
******************************
** GENERATE THE REST OF THE ELEMENTS BY COPYING ELEMENT 1, MAKE 10 ELEMENTS,
ELEMENT NUMBER
** INCREMENT OF 1 AND NODE NUMBER INCREMENT OF 2, 1 ROW, ASSIGN TO ELSET
'EALL'
*ELGEN, ELSET=EALL
1,10,2,1,1,1,1
******************************
** DEFINE MATERIAL CALLED 'STEEL'
*MATERIAL,NAME=STEEL
******************************
** DEFINE ELASTIC MATERIAL PROPERTIES, E = 200*10^9 AND POISSON'S RATIO 0
*ELASTIC
200E9, 0.3
******************************
** DEFINE THE BEAM SECTION, ASSIGN IT TO THE ELEMENT SET 'EALL', ASSIGN IT
WITH THE 'STEEL' MATERIAL AND DEFINE A
** 0.2 M THICK BY (depth)s M DEEP BEAM
*BEAM SECTION, ELSET = EALL, MATERIAL = STEEL, SECTION = RECT
0.2, 1.0
******************************
**ASK FOR HISTORY AND MODEL DATA TO BE INCLUDED IN RESULTS FILE
*PREPRINT, ECHO=YES, MODEL=YES, HISTORY=YES
******************************
** BEGIN LINEAR STEP
*STEP, PERTURBATION
******************************
** DEFINE STEP AS STATIC
*STATIC
```

```
    *****************************
    ** DEFINE FIXED BOUNDARY AT NODE 1 LOCKED IN X AND Y DIRECTION AND AT NODE
21 LOCKED IN Y DIRECTION
    *BOUNDARY
    1, 1, 2
    21, 2, 2
    *****************************
    ** DEFINE DISTRIBUTED LOAD OF -100000 KN/M IN Y DIRECTION ON ELSET 'EALL'
    *DLOAD
    EALL, PY, -100000
    *****************************
    ** ASK ABAQUS TO PRINT COORDINATE, STRESS AND STRAIN FOR EACH ELEMENT (AT
INTEGRATION POINTS INSIDE CELLS)
    *EL PRINT
    COORD
    S
    E
    *****************************
    ** ASK ABAQUS TO PRINT STRESS VALUES AT NODES
    *EL PRINT, POSITION=AVERAGED AT NODES
    S
    *****************************
    ** ASK ABAQUS TO PRINT FORCES AND DISPLACEMENTS AT NODES
    *NODE PRINT
    U
    RF
    *****************************
    ** END THE STEP
    *END STEP
```

**B. CPS4I.inp**

```
                                   CPS4I.inp
*HEADING
5m LONG BY 1m DEEP BY 0.2m THICK SIMPLY SUPPORTED BEAM WITH 100 KN/M DISTRIBUTED
LOAD, 20X20 CPS4I MESH, E=200 GPa, V = 0.3
******************************
*NODE
** DEFINE LOCATION OF CORNER NODES
1, 0.0, 0.0
21, 5.0, 0.0
2001, 0.0, 1.0
2021, 5.0, 1.0
******************************
*NGEN, NSET=BOTTOM
** GENERATE ELEMENTS ALONG BOTTOM AND ADD NODES TO SET "BOTTOM"
1, 21, 1
******************************
** CREATE NODES ALONG TOP AND ADD NODES TO SET "TOP"
*NGEN, NSET=TOP
2001, 2021, 1
******************************
** CREATE NODES BETWEEN "TOP" AND "BOTTOM", 20 NODES BETWEEN ENDS WITH NODE
NUMBERING STEP OF 100
*NFILL
BOTTOM, TOP, 20, 100
******************************
**DEFINE THE FIRST "MASTER ELEMENT" NUMBER 1 AS A CPS4I CONSISTING OF NODES
1,2,102 AND 101
*ELEMENT, TYPE=CPS4I
1, 1, 2, 102, 101
******************************
** GENERATE THE REST OF THE ELEMENTS BY COPYING ELEMENT 1, 20 ELEMENTS IN X, 20
ELEMENTS IN Y, NODE NUMBER INCREMENT OF 1 IN X, NODE NUMBER INCREMENT OF 100 IN
Y, ELEMENT NUMBER INCREMENT OF 100 FROM ROW TO ROW, ASSIGN ELEMENTS TO SET
"EALL"
*ELGEN, ELSET=EALL
1,20,1,1,20,100,20
******************************
** DEFINE MATERIAL CALLED "STEEL"
*MATERIAL, NAME=STEEL
******************************
** DEFINE ELASTIC MATERIAL PROPERTIES, E = 200 GPa AND POISSON'S RATIO 0.3
*ELASTIC
200E9, 0.3
******************************
** CREATE SECTION FROM MATERIAL "STEEL" AND ASSIGN IT TO ELEMENT SET "EALL",
THICKNESS IS 0.2M
*SOLID SECTION, MATERIAL=STEEL, ELSET=EALL
0.2
******************************
**ASSIGN ELEMENTS IN TOP ROW TO ELSET "TOP"
*ELSET, ELSET=TOP, GENERATE
381, 400, 1
******************************
```

```
**ASK FOR HISTORY AND MODEL DATA TO BE INCLUDED IN RESULTS FILE
*PREPRINT, ECHO=YES, MODEL=YES, HISTORY=YES
******************************
** BEGIN LINEAR STEP
*STEP, PERTURBATION
******************************
** DEFINE STEP AS STATIC
*STATIC
******************************
** DEFINE FIXED BOUNDARY AT NODE 1 LOCKED IN X AND Y DIRECTION AND AT NODE 21
LOCKED IN Y DIRECTION
*BOUNDARY
1, 1, 2
21, 2, 2
******************************
** DEFINE CONCENTRATED LOAD OF -0.5N IN Y DIRECTION ON EACH END CORNER NODE
*DLOAD
TOP, P3, 500E3
******************************
** ASK ABAQUS TO PRINT COORDINATE, STRESS AND STRAIN FOR EACH ELEMENT (AT
INTEGRATION POINTS INSIDE CELLS)
*EL PRINT
COORD
S
E
******************************
** ASK ABAQUS TO PRINT STRESS VALUES AT NODES
*EL PRINT, POSITION=AVERAGED AT NODES
S
******************************
** ASK ABAQUS TO PRINT FORCES AND DISPLACEMENTS AT NODES
*NODE PRINT
U
RF
******************************
** END THE STEP
*END STEP
```

## B. Python Scripts

## A. Loop to create input and batch files

File - C:\Users\Ali\Documents\MEGA\University\5th_Year\FEM 4\Project\beam_models\Create_Beam_Inps.py

```python
from beam_input_generator import beam_input_generator # Import input file writing function
import os # import the os library will allow the loop to create subfolders
mesh_size = [2,10] # define the range of mesh sizes
element = ['B23', 'B22', 'B21'] # define the range of element types
beam_type = ['thin', 'thick'] # define the two beam sizes
filepath = os.path.dirname(os.path.realpath(__file__)) # get the location the script is being run from
batch = open('run_beam_models.bat', 'w') # create a new batch file
for n in mesh_size: # for each mesh size
    for type in element: # for each element type
        for beam in beam_type: # for each beam size
            filename = beam_input_generator(type, n, beam) # create the input file
            # the input file writing function creates a subfolder in the run directory so the first part of the batch
            #  command needs to move the computer to this directory
            batch.write('cd ' + filepath + '\\'  + filename + '& ')
            # Tell the batch file to print a statement at the start of each simulation
            batch.write('echo Running next simulation & ')
            # Run the simulation
            batch.write('abaqus job=' + filename + ' interactive & ')
batch.write('PAUSE')
batch.close()

```

## B. Loop to create input and batch files

File - C:\Users\Ali\Documents\MEGA\University\5th_Year\FEM 4\Project\beam_models\beam_input_generator.py

```
def beam_input_generator(element_type, element_number, beam_size):
    """"
    Beam model input file generator for FEMSS4 project
    Alasdair Gray, S1208454
    November 2016
    This function takes 3 input parameters:
    element type: either 'B21', 'B22' or 'B23'
    element number: The number of elements along the beam
    beam size: either 'thin' or 'thick'
    The function then creates a subfolder in the directory it is being run from and writes an appropriate input file
    with the same name to the subfolder, the function outputs the filename as a string.
    """
    # First, define the element properties based on the chosen element type
    if element_type in ['B21', 'B23']:
        node_per_el = 2
        el_def = '1, 2'
    else:
        node_per_el = 3
        el_def = '1, 2, 3'
    # Define properties based on beam size
    if beam_size == 'thick':
        depth = 1.0
        load = -100000
    elif beam_size == 'thin':
        depth = 0.2
        load = -10000
    # Calculate the number of nodes required along the beam
    total_nodes = str(element_number*(node_per_el-1)+1)
    import inspect, os
    filename = element_type + '_' + beam_size + '_' + str(element_number) + 'EL'
    Parent = os.path.dirname(os.path.realpath(__file__)) # Get the location the function is being run in
    Child = Parent + '/' + filename # Create parameter specific subfolder name
    if not os.path.exists(Child): # Create subfolder if it doesn't already exist
        os.makedirs(Child)
    os.chdir(Child)
    output = open(filename+'.inp', 'w') # Define the inp file to write to
    template = """
    ***************************************************************
    ** FEMSS4 Project
```

```
                                                            **
    ** Alasdair Gray, S1208454                                              **
    ** Beam model input file generated by my wonderful python script **
    *********************************************************************
    *HEADING
    5m LONG SIMPLY SUPPORTED %(beam_size)s BEAM WITH %(load)s kN/m DISTRIBUTED LOAD, %(element_number)s ELEMENT
    %(element_type)s MESH
    ******************************
    *NODE
    ** DEFINE LOCATION OF END NODES
    1, 0.0, 0.0
    %(total_nodes)s, 5.0, 0.0
    ******************************
    *NGEN, NSET=NALL
    ** GENERATE NODES ALONG LENGTH AND DEFINE NODE SET 'NALL'
    1, %(total_nodes)s, 1
    ******************************
    **DEFINE THE FIRST 'MASTER ELEMENT' NUMBER 1 AS A %(element_type)s
    *ELEMENT, TYPE=%(element_type)s
    1, %(el_def)s
    ******************************
    ** GENERATE THE REST OF THE ELEMENTS BY COPYING ELEMENT 1, MAKE %(element_number)s ELEMENTS, ELEMENT NUMBER
    ** INCREMENT OF 1 AND NODE NUMBER INCREMENT OF %(node_shift)s, 1 ROW, ASSIGN TO ELSET 'EALL'
    *ELGEN, ELSET=EALL
    1,%(element_number)s,%(node_shift)s,1,1,1,1
    ******************************
    ** DEFINE MATERIAL CALLED 'STEEL'
    *MATERIAL,NAME=STEEL
    ******************************
    ** DEFINE ELASTIC MATERIAL PROPERTIES, E = 200*10^9 AND POISSON'S RATIO 0
    *ELASTIC
    200E9, 0.3
    ******************************
    ** DEFINE THE BEAM SECTION, ASSIGN IT TO THE ELEMENT SET 'EALL', ASSIGN IT WITH THE 'STEEL' MATERIAL AND DEFINE A
    ** 0.2 M THICK BY (depth)s M DEEP BEAM
    *BEAM SECTION, ELSET = EALL, MATERIAL = STEEL, SECTION = RECT
    0.2, %(depth)s
    ******************************
    **ASK FOR HISTORY AND MODEL DATA TO BE INCLUDED IN RESULTS
```

```
FILE
    *PREPRINT, ECHO=YES, MODEL=YES, HISTORY=YES
    ******************************
    ** BEGIN LINEAR STEP
    *STEP, PERTURBATION
    ******************************
    ** DEFINE STEP AS STATIC
    *STATIC
    ******************************
    ** DEFINE FIXED BOUNDARY AT NODE 1 LOCKED IN X AND Y DIRECTION AND AT NODE %(total_nodes)s LOCKED IN Y DIRECTION
    *BOUNDARY
    1, 1, 2
    %(total_nodes)s, 2, 2
    ******************************
    ** DEFINE DISTRIBUTED LOAD OF %(load)s KN/M IN Y DIRECTION ON ELSET 'EALL'
    *DLOAD
    EALL, PY, %(load)s
    ******************************
    ** ASK ABAQUS TO PRINT COORDINATE, STRESS AND STRAIN FOR EACH ELEMENT (AT INTEGRATION POINTS INSIDE CELLS)
    *EL PRINT
    COORD
    S
    E
    ******************************
    ** ASK ABAQUS TO PRINT STRESS VALUES AT NODES
    *EL PRINT, POSITION=AVERAGED AT NODES
    S
    ******************************
    ** ASK ABAQUS TO PRINT FORCES AND DISPLACEMENTS AT NODES
    *NODE PRINT
    U
    RF
    ******************************
    ** END THE STEP
    *END STEP """
    # Now define the parameter based strings which will be inserted into the template string
    context = {
    "total_nodes":total_nodes,
    "element_type":element_type,
    "el_def":el_def,
    "element_number": str(element_number),
    "node_shift" : str(node_per_el-1),
    "depth" : str(depth),
    "load" : str(load),
    "beam_size" : beam_size
```

```python
    }
    # Fill the template blanks with the parameter based strings and write to the input file
    output.write(template % context)
    output.close() # Close the input file
    return filename # Return the parameter specific filename to the caller

if __name__ == "__main__":
    beam_input_generator('B22', 2, 'thin')

```