ME 7120: Project 2

Finite Element Method Applications

Due: December 2, 2016

Team Members:

Michelle Callaham

Sagar Sangle

Mohammed Al Rifaie

# Project Introduction

This project allowed us to compare a brick finite element model using two different software programs. Using the WFEM code in MATLAB, we wrote a brick element code that would ultimately allow us to assemble the stiffness and mass matrix for the elements, plot the element, allow us to apply boundary conditions, and loads to find the displacement when a force is applied. There were also multiple checkpoints for us to do along the way to ensure that our brick element was working properly. We compared the results using ANSYS and the beam element code (from project 1). In ANAYS, we constructed the model with the same geometry, material properties, and number of elements. In the end we were able to compare the deflection results among the WFEM code, ANSYS simulation, and beam code. To prove this test for this project, we evaluated our codes on two different examples: a pyramid and a tapered cylinder element.

# Brick Element Code and Tests

We created a brick element code in MATLAB that is compatible with the WFEM code. The code we wrote was set up to include the correct number of nodal points, B matrix, Jacobian matrix, elasticity matrix, and shape functions, all of which are essential to building the brick element in WFEM. We also assembled the local and global stiffness and mass matrices, applied shear locking, and applied gauyn reduction. The shear locking produced a B matrix the size of 33x33. We applied gauyn reduction, we reduced the size of the matrix from 33x33 and 24x24. This removed all the zeros from the matrtix.

To prove our brick element was working, there were several checks that we performed. First, we ensured that the eigenvalues were correct for the brick element, six of our eigenvalues had the value of 0 and eighteen eigenvalues were positive. Figure 1 below shows the output of the MATLAB file for the eigenvalues of our brick element.

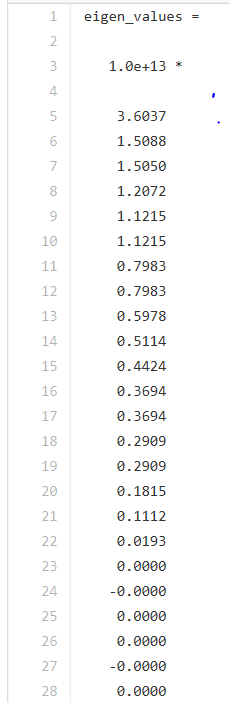


Figure : Eigenvalue Test Reults

Additionally, we checked the tension, bending, shear, and torsion in each direction of the brick under different loading conditions. We applied tension, bending, shear, and torsion on our brick element by applying different forces and loading conditions. We repeated this process for all directions of the brick: x, y, and z. The images produced were as expected and showed the fundamental characteristics of these properties. For the simple brick element we created for these tests, A=1, L=1, and F=40. The result we got from MATLAB for the Tension test is 1.966E-10. In the closed form equation, we also get the value 1.966E-10. Below in Figures 2 through 4, are images of our brick element under these different conditions of bending, shear, tension, and torsion.

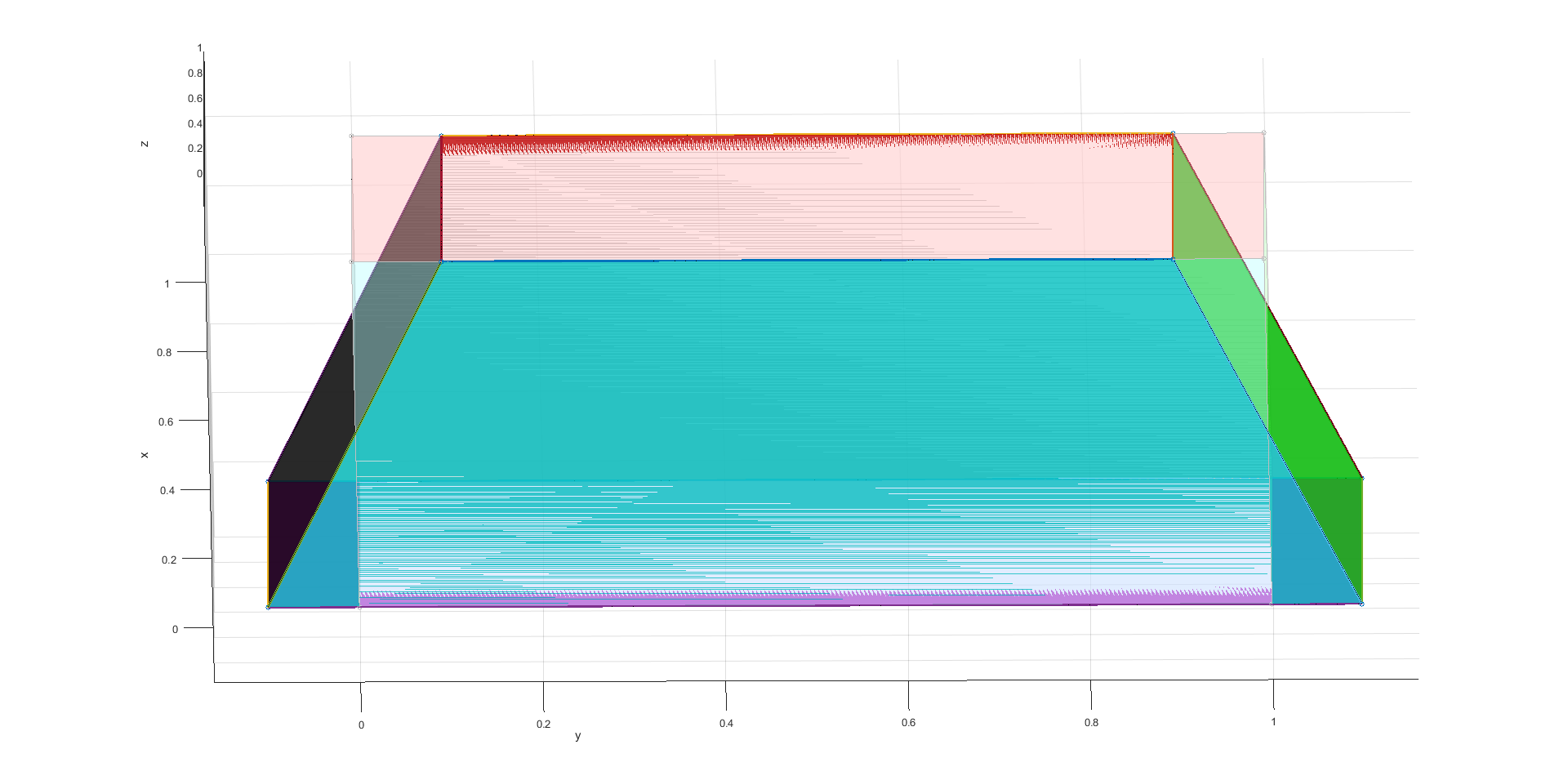


Figure : Bending Test

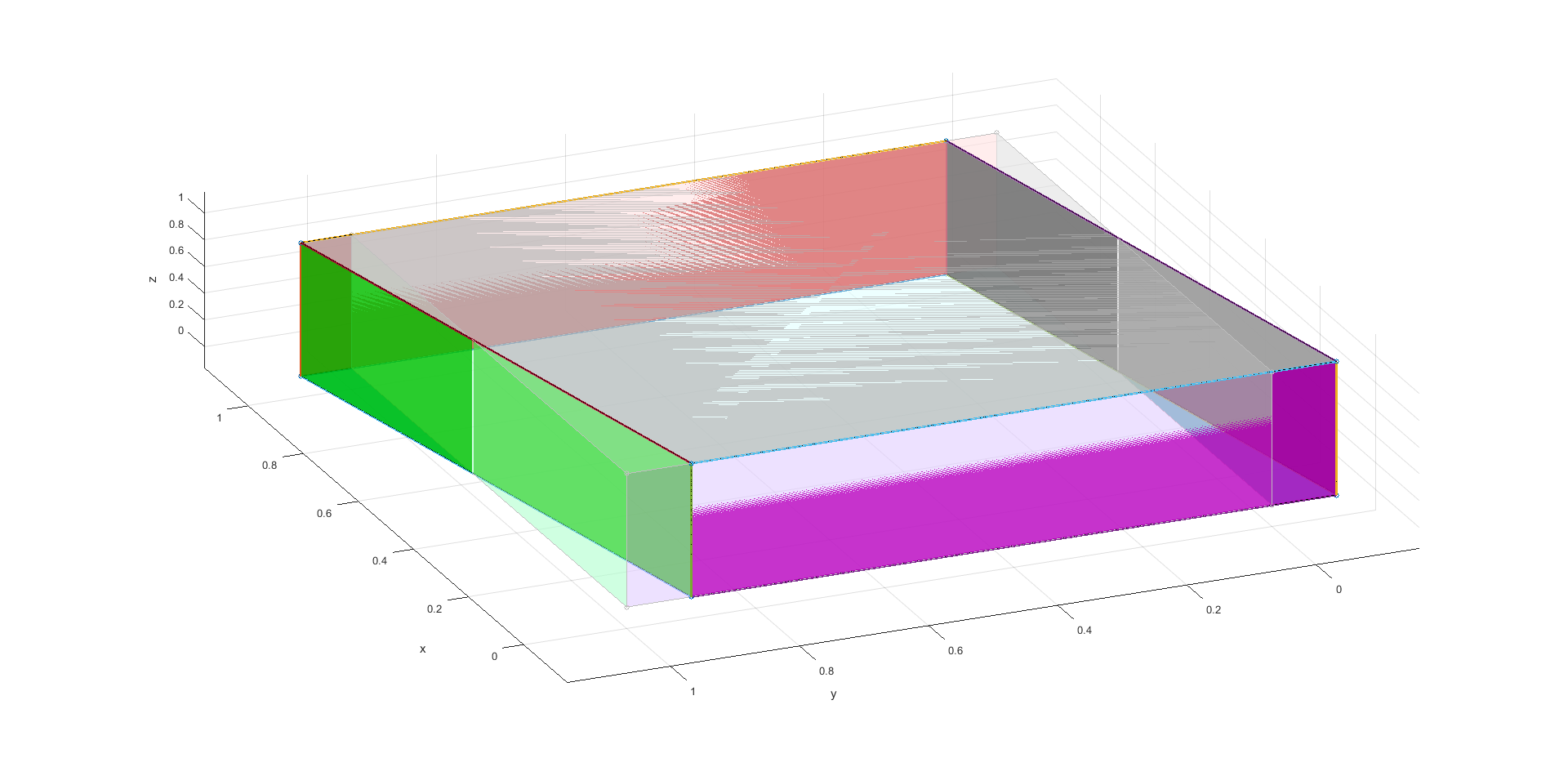


Figure : Shear Test

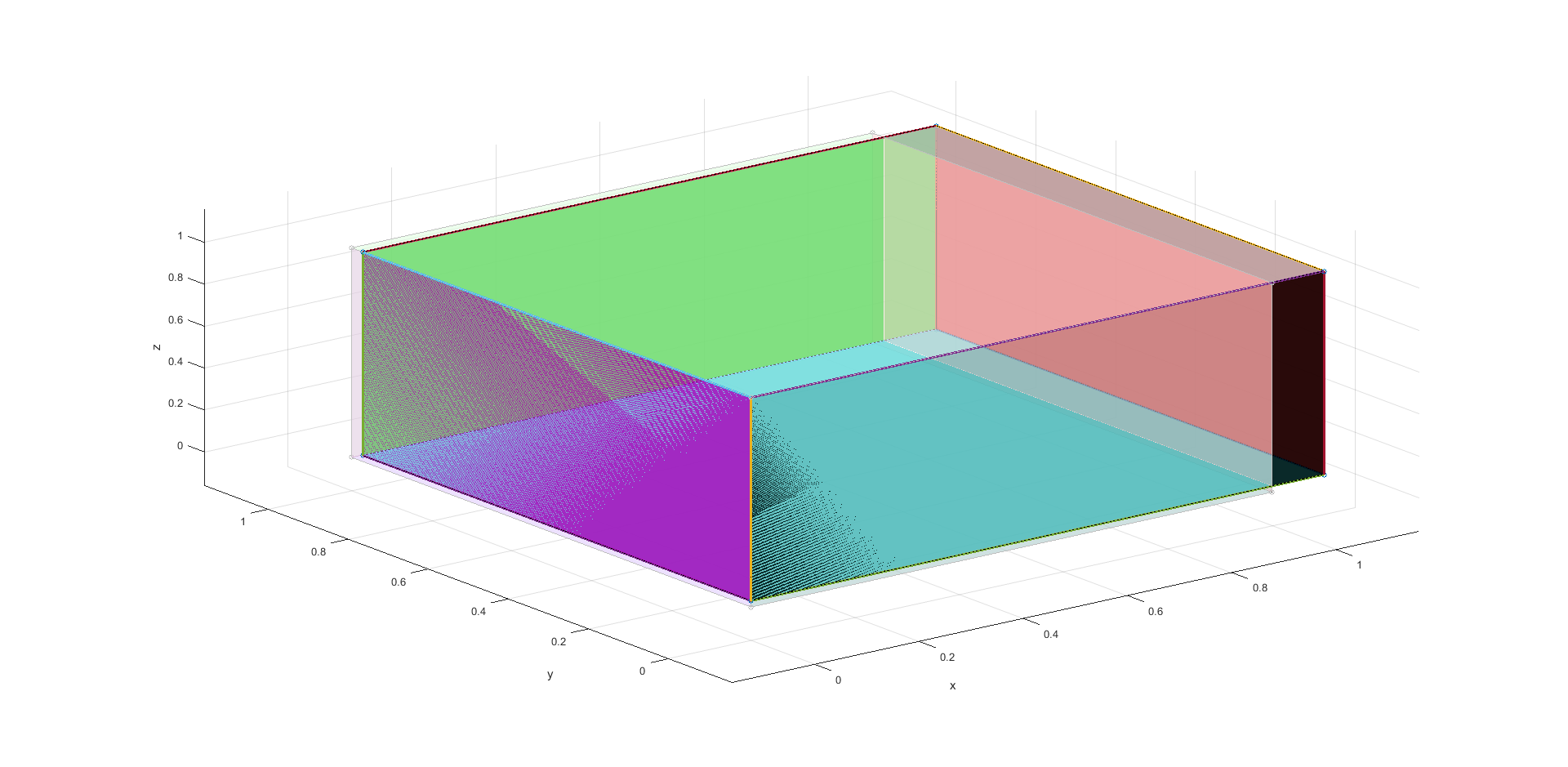


Figure : Tension Test

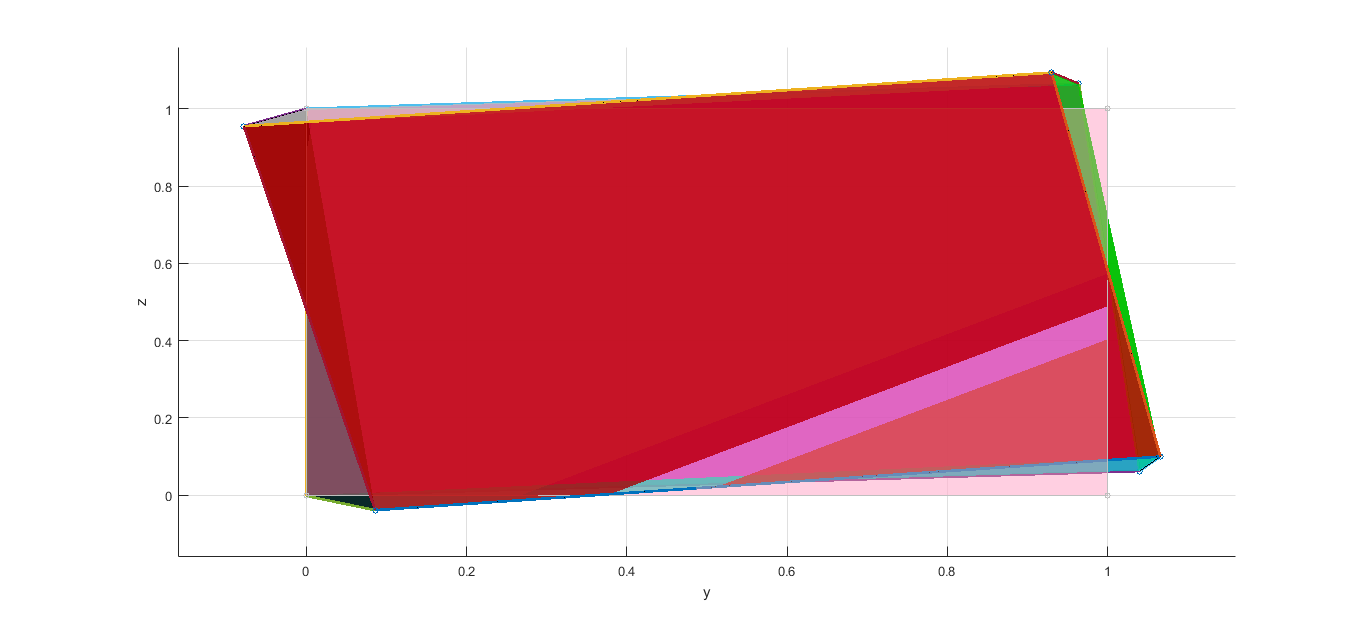


Figure : Torsion Test

Lastly, we did the patch test on a simple brick element. We created a very simple brick element that was 2x2x2 and had center points on each surface. By adding an extra node and applying evenly distributed forces, we can make sure our element still maintains its shape. Below in Figures 6 through 8, our images produced by MATLAB are shown. Figure 6 is a good illustration of the isometric view of the patch test performed on the brick element and seeing all the nodal points. Node 8 is fixed and node 4 is fixed in y and z direction. The other nodal points shown are fixed only in the y direction. These are the boundary conditions for the patch test. Figure 7 shows the patch test and the forces applied on the brick element. Figure 8 shows a side view of the element showing how the brick maintained its shape.

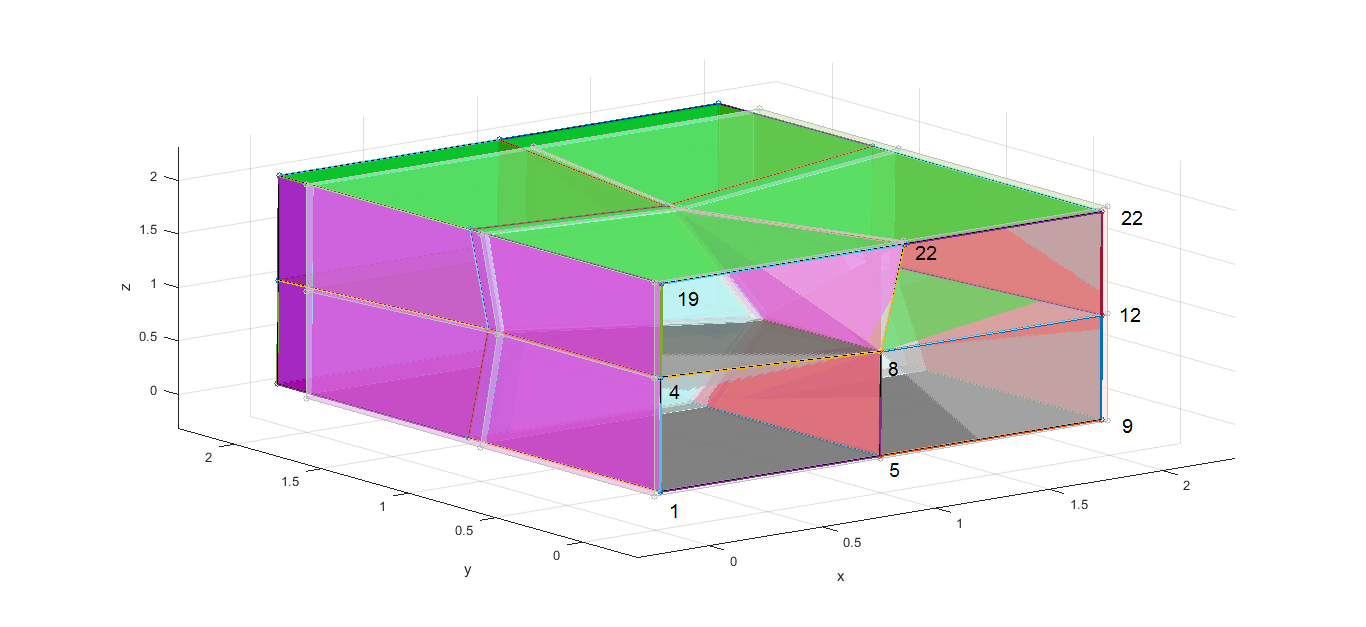


Figure 6: Patch Test – Boundary Conditions Applied to Brick Element

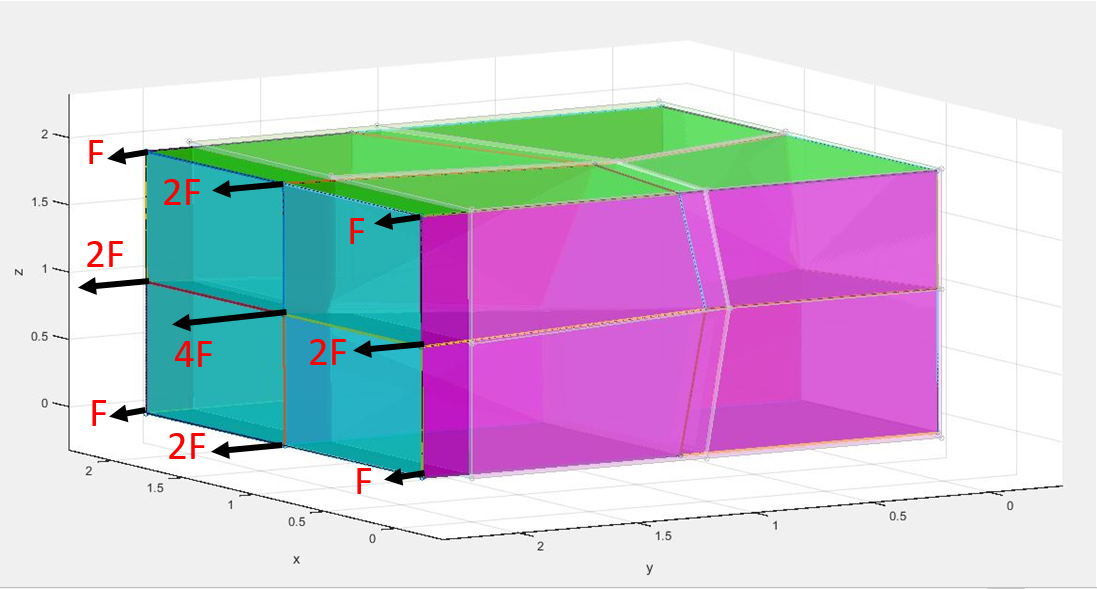


Figure : Patch Test - Forces Applied to Brick Element

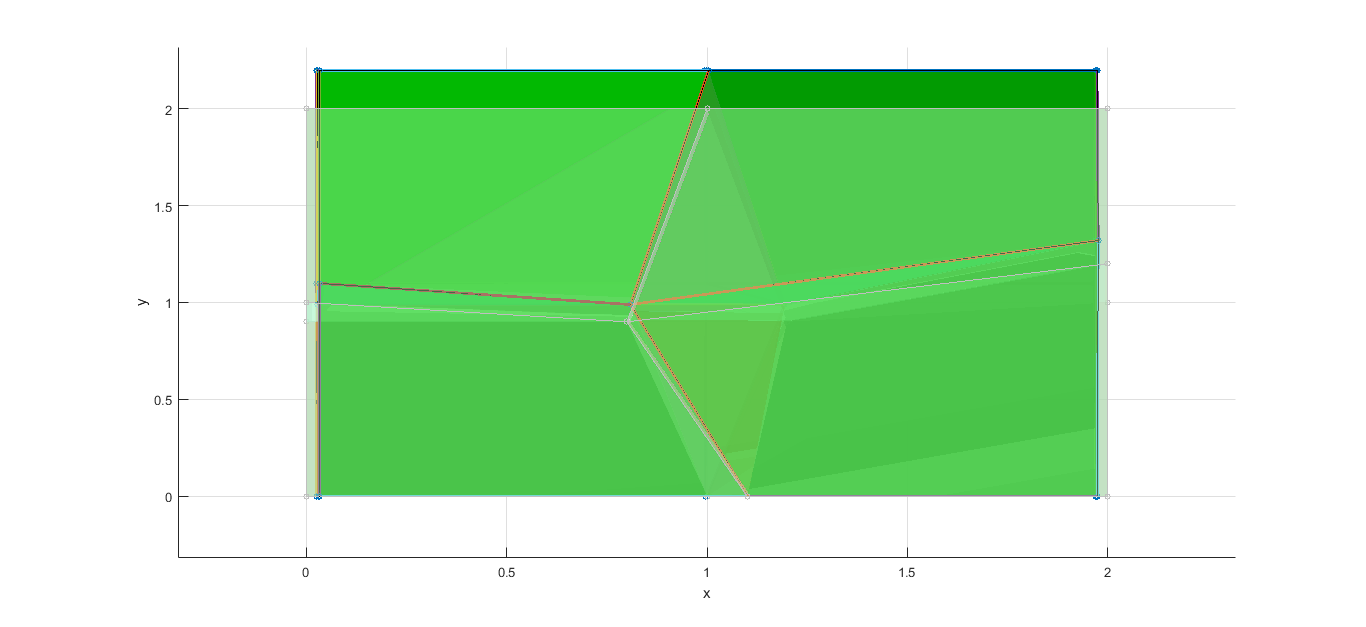


Figure 8: Side View of Patch Test on Brick Element

All of the tests we performed on our brick element produced accurate results and ensured that our brick element was working properly and we could proceed with testing the pyramid and tapered cylinder submodules.

# Finite Element Submodules

We were tasked to test our code on two different submodules. The first was a pyramid with a height of 190nm, bottom side length of 74nm, and top side length of 10nm. The second submodule was a tapered cylinder element with a height of 190nm, bottom diameter of 74nm, and top diameter of 10nm. We split the element into multiple elements. We also we also based the material properties off the steel material predefined in WFEM.

*Material Properties:*

The objective was to find the tip displacement of a vertical post to a 10mN transverse tip load. First, we needed to find the tip displacement using our MATLAB code. We also needed to compare these results to our beam model (from project 1) and then using ANSYS. We also needed to find the error percentage among the results produced from the different methods.

## Pyramid Element Example

### Pyramid Element using MATLAB

We first built the pyramid using our MATLAB code and ensured that the pyramid looked like the desired geometry before any loading conditions were defined. Figure 9 below is our pyramid element without any loading conditions.

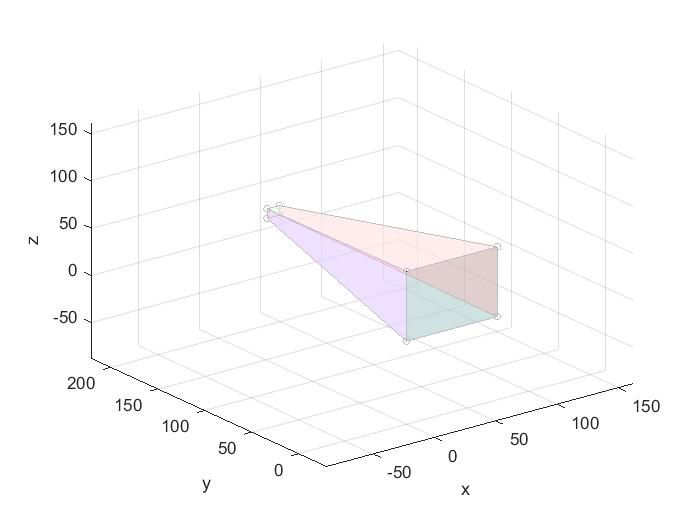


Figure 9: Pyramid Element

Since our pyramid element appeared to be working correctly in our MATLAB program, we applied the desired load of 10nM to the tip of the element. Using the code, we determined the deflection of the tip was -3.72222E-10m. The pyramid element after deflection is shown below in Figure 10.

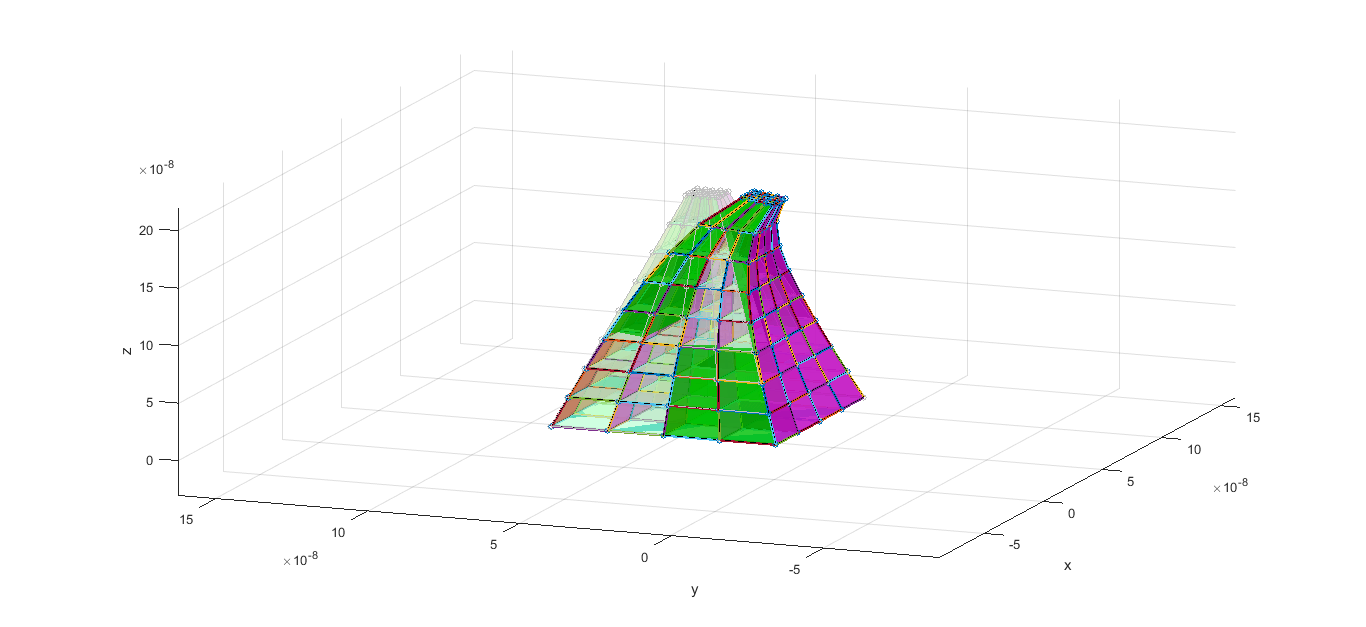


Figure 10: Pyramid Element after Load is Applie

### Pyramid Element using ANSYS

Next, we used ANSYS to compare our results for the tip deflection of the pyramid when the load was applied. Using the ANSYS method, the tip deflection had a result of -0.347E-09m. Figure 11 below shows the results of the deflection of the pyramid element using the ANSYS software. Also it should be noted that we did a convergence test in ANSYS for the element to find the most accurate number of elements to use.

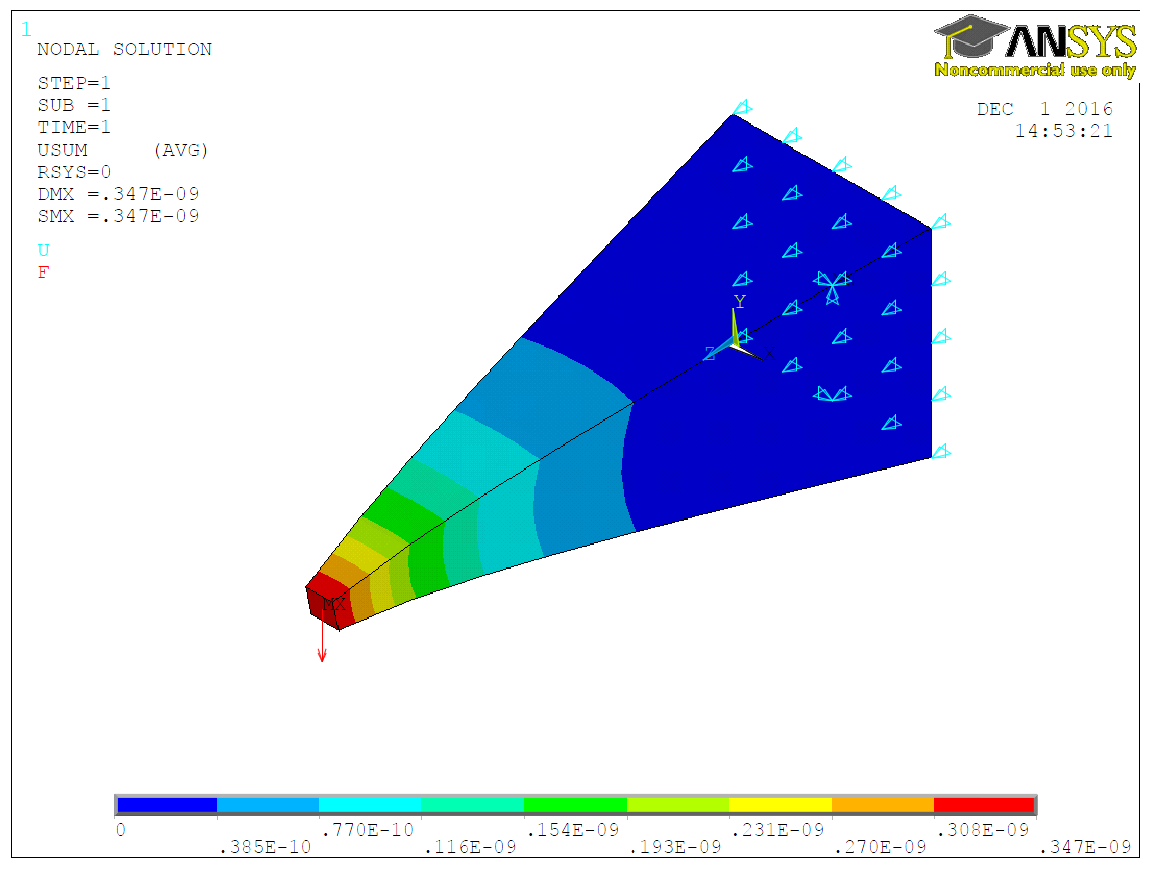


Figure 11: Pyramid Element Deformation in ANSYS

### Pyramid Element using Beam Model

We also ran the pyramid element using the beam model from project 1. We got a tip deflection value of -3.0448E-10m. The Figure 12 shows the deflection of the pyramid element using the beam code.

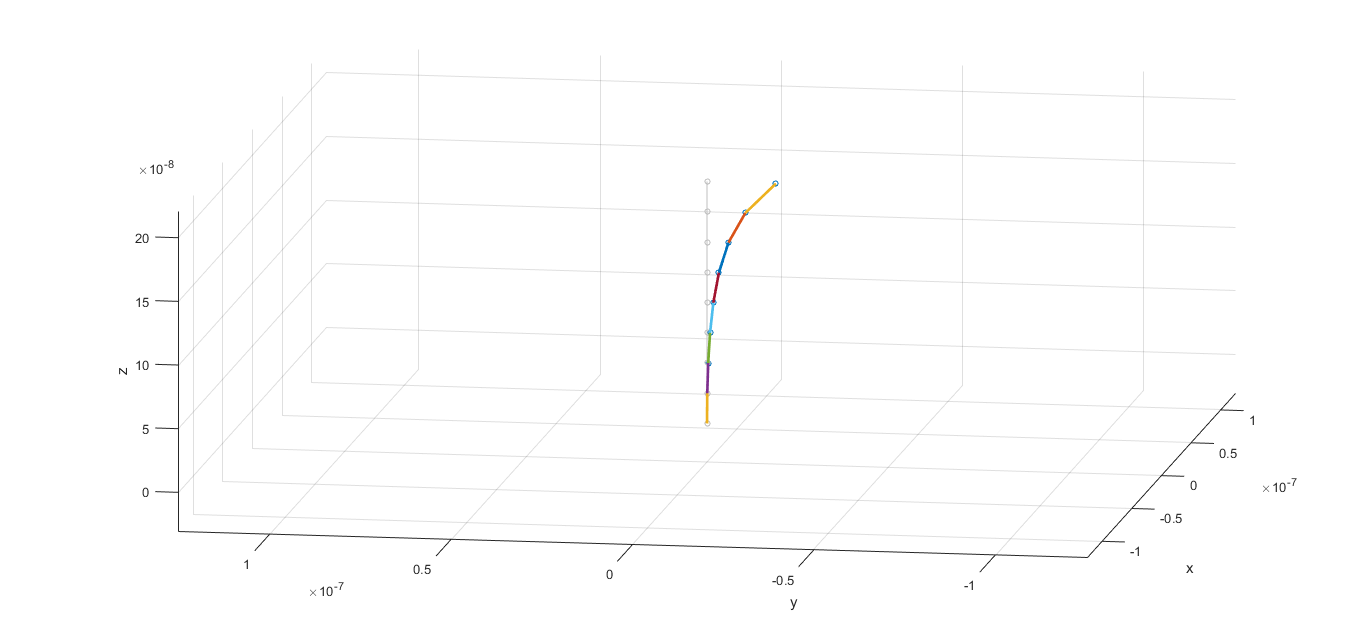


Figure : Pyramid Element Deflection

### Comparison of Results

We are now able to compare the values for the tip deflection from the brick element in WFEM, ANSYS, and the beam element in WFEM. Table 1 shows the values for the tip deflection using the brick element in MATLAB and the ANSYS outcomes, as well as the error.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Brick Element in MATLAB | ANSYS Method | Error (%) |
| Tip Deflection | -3.72222E-10m | -0.347E-09m | 6.77% |

Table : Tip Deflection of Pyramid Element Comparison between Brick Element and ANSYS Solutions

We also used the beam code from project 1 to determine the tip deflection of the pyramid element. Table 2 shows the comparison between the Brick code in WFEM and the Beam Code in WFEM.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Brick Element in MATLAB | Beam Element in WFEM | Error (%) |
| Tip Deflection | -3.72222E-10m | -3.0448E-10m | 18.2% |

Table 2: Tip Deflection of Pyramid Element Comparison between Brick Element and Beam Element

The beam element is not written to consider all the degrees of freedom, such a shear, like how our brick element does. We would not use the beam element for our brick element (in reality) but it is a good comparison of the tip deflection for this project.

Overall, there were very small errors between the different methods used to find the deflection for this submodule. Each method has its own way of solving the finite element problem, so that is where the error could lie. We feel that our brick element code created with WFEM gave the most accurate answer.

## Tapered Cylinder Element Example

### Tapered Cylinder Element using MATLAB

Like we did with the pyramid element, we first built the tapered cylinder element using our MATLAB code. Since the brick code is not setup for a circular object, we created a figure that would be “good enough” for this finite element analysis. The tapered cylinder element was made out of 12 nodes for each circular face. There was a 30° difference between the nodal points, so we thought that this would be sufficient for this problem. The figure displayed of the tapered cylinder element is shown in Figure 13 and displayed the results we were expecting.

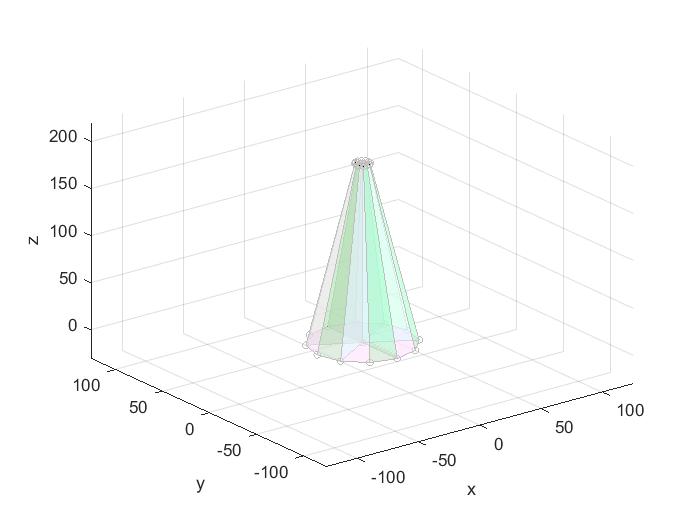


Figure : Tapered Cylinder Before Deformation

Since our tapered cylinder element appeared to be working correctly in our MATLAB program, we applied the desired load of 10nM to the tip of the element. Using the code, we determined the deflection of the tip was -6.5242E-10m. The tapered cylinder element after deflection is shown below in Figure 14.

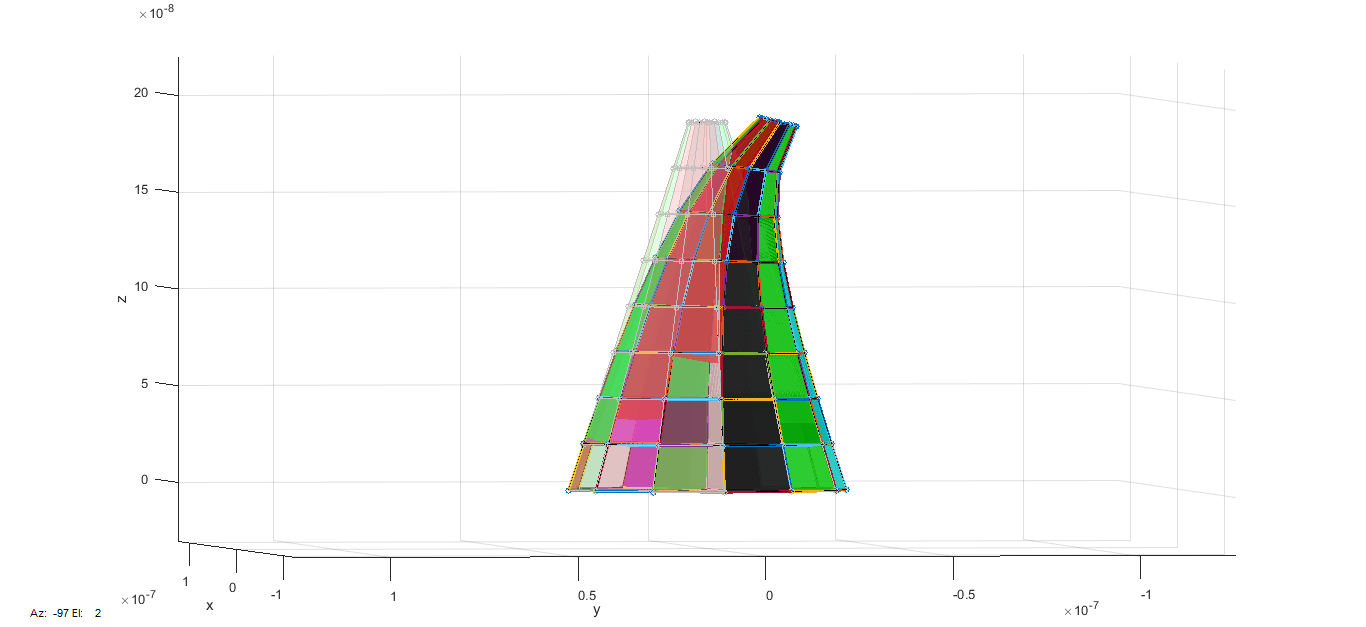


Figure : Tapered Cylinder After Deflection

### Tapered Cylinder Element using ANSYS

Next, we used ANSYS to compare our results for the tip deflection of the tapered cylinder as the load was applied. The tip deflection using ANSYS had a result of -.584E-09m. A convergence study was also performed in ANSYS for this element. Figure 15 below shows the image displaying the tapered cylinder’s tip deflection.

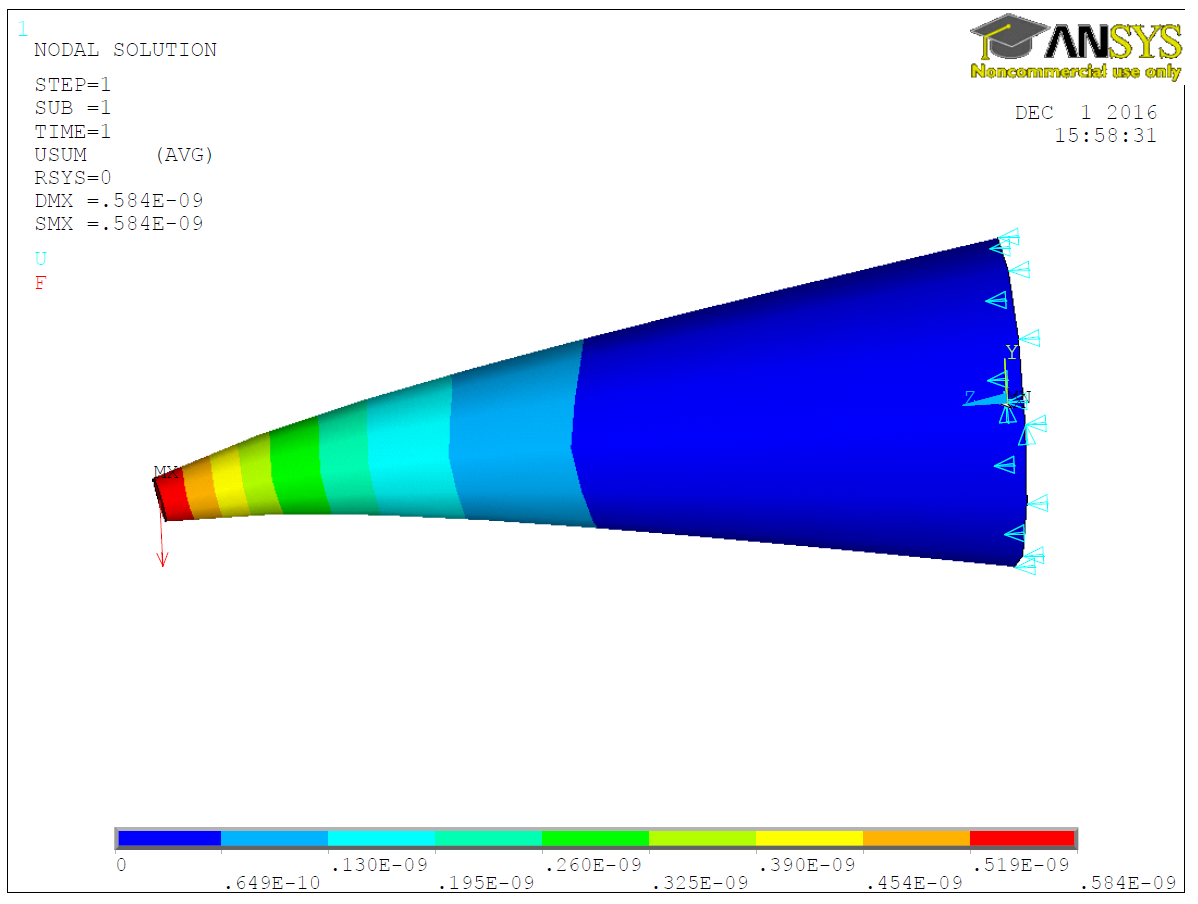


Figure : Tapered Cylinder Element in ANSYS

### Tapered Cylinder Element using Beam Model

We also ran the tapered cylinder element using the beam model from project 1. We got a tip deflection value of -5.169-10m. The Figure 16 shows the deflection of the tapered cylinder element using the beam code.

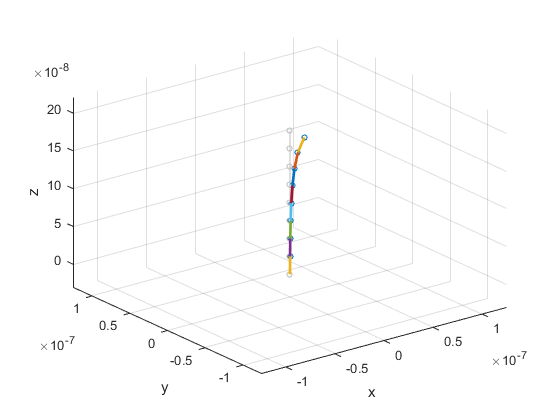


Figure : Tapered Cylinder Element Deflection

### Comparison of Results

We are now able to compare the values for the tip deflection from the tapered cylinder element in WFEM, ANSYS, and the beam element in WFEM. Table 3 shows the values for the tip deflection using the brick element in MATLAB and the ANSYS outcomes, as well as the error percentage.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Brick Element in MATLAB | ANSYS Method | Error (%) |
| Tip Deflection | -6.4242E-10m | -0.584E-09m | 9.09% |

Table : Tapered Cylinder Deflection Comparison between Brick Element and ANSYS

We can also compare the results from the tapered cylinder element code and the beam element codes. Table 4 below shows the values for the tip deflection ad well as the error percentage.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Brick Element in MATLAB | Beam MATLAB Code | Error (%) |
| Tip Deflection | -6.4242E-10m | -5.169E-10m | 19.53% |

Table : Tapered Cylinder Element Deflection Comparison between Brick Element and Beam Element in WFEM

For this element overall, the values were relatively close to one another. The Brick Element code in MATLAB through the WFEM program and ANSYS had very close numbers and either could be acceptable. The beam element does not allow for all the degrees of freedom of the brick element, so it produced a higher error percentage.

# Conclusion

This project allowed us to use three different methods of solving a brick element as a finite element model. We created a brick element code compatible with the WFEM software in MATLAB. We then compared our results using ANSYS and the beam code from project1 in MATLAB. All methods used produced accurate results, however both the Brick MATLAB code and ANSYS gave the best results. The Beam Code just does not allow for enough degrees of freedom. There was a very small difference among the tip deflection results from ANSYS and the Brick code. Either of these two methods could be acceptable in industry. This project allowed us to see that comparing results in different software methods is important.