Assessing Linear Elastic Behavior Through Repeated Bending Tests with Digital Image Correlation (DIC) and Analytical Comparisons

By: Maria Alvarado, Gabriel Kret, and Jonas Margono

Last Revised: 5/14/2025

ME-360: Engineering Experimentation

The Cooper Union for the Advancement of Science and Art

Table of Contents

Abstract.......... 3

1. Introduction 4

2. Experimental.......... 5

2.1 Equipment 6

2.1.1 Instrumentation.......... 6

2.1.2 Apparatus 6

2.2 Procedure 7

3. Results 11

3.1 Mechanics of Material Solution 15

3.2 Finite Element Model.......... 23

3.3 Hollow Box Beam 25

3.4 Small Solid Beam.......... 27

3.5 Large Solid Beam.......... 30

4. Discussion.......... 34

5. Conclusions.......... 37

6. Acknowledgments.......... 38

7. References 39

Appendix I: Problem Formulation Diagram.......... 40

Appendix II: Bill of Materials.......... 41

Appendix III: Sample Data.......... 42

Appendix IV: Python Code and Matlab 43

Mechanics of Materials Simulation Model 43

Python Code for Strain Distributions 61

MATLAB Code to Generate Heatmaps for VIC Data 65

Appendix V: SolidWorks Simulations 72

Hollow Box Beam Simulations.......... 72

Small Solid Beam Simulations 75

Large Solid Beam Simulations 80

Polyurethane Solid Beam Simulations 83

Appendix VI: M1 Style Procedure and Datasheet.......... 85

Abstract

This study investigates the linear elastic behavior of aluminum beams subjected to a three-point bending test, employing Digital Image Correlation (DIC), finite element analysis (FEA), and analytical methods derived from classical Mechanics of Materials theory. The primary objective was to create a reliable, repeatable experimental framework enabling precise full-field strain measurement through DIC, offering near-pixel level insights into localized deformation behaviors often overlooked in traditional theoretical models. Experimental results were directly compared with theoretical predictions and FEA simulations conducted in SolidWorks, highlighting both the accuracy and limitations of each method.

Significant findings include strong agreement among the DIC measurements, analytical solutions, and FEA simulations, particularly at higher loads, validating the linear elasticity assumptions fundamental to mechanics of materials. However, discrepancies identified at lower loads underscore the sensitivity of DIC to minimal strain levels, showing the importance of rigorously validating appropriate test conditions in an experimental setup. It is notable that when using the DIC, beams that deflect less, whether that be due to a lower load or higher moment of inertia, have nosier and less reliable results. The goal of the experiment is to develop a clear, streamlined process such that future students will be able to replicate the experiment and achieve high quality results while learning from the methods outlined in this report. The integrated approach presented in the report not only provides a comprehensive validation framework for assessing linear elastic behavior but also demonstrates the critical role of combining theory, simulation, and experimentation in understanding real-world material response. The work emphasizes the nuance involved in correlating theoretical models with experimental data and highlights how hands-on methodologies can reveal deeper insights into structural behavior under load.

1. Introduction

In engineering education, bridging the gap between theoretical analysis and real-world material behavior is essential for developing a deep and practical understanding of mechanical systems. While traditional approaches in Mechanics of Materials offer valuable insight into concepts like stress, strain, and elastic behavior, these methods often rely on simplifying assumptions that do not capture the full complexity of real-world applications, especially when it comes to material imperfections, surface wear, or repeated loading cycles.

This project focuses on the repeatable testing of a beam under three-point bending within the linear elastic deformation region, deliberately avoiding plastic deformation to ensure consistent results over multiple trials. Identifying and staying within this region is essential, as it is a required assumption of Mechanics of Materials theory. The goal is to develop a reliable, hands-on experimental setup that enables students to explore core concepts in solid mechanics such as elastic deformation, linearity of stress-strain relationships, and deflection behavior, while reinforcing the practice of validating theoretical predictions through experimentation.

To this end, the experiment integrates Digital Image Correlation (DIC) as a non-contact, full-field strain measurement technique. DIC enables precise tracking of material surface deformation through speckle-pattern imaging during loading, yielding high-resolution strain maps that reflect localized mechanical behavior. These experimental results are then compared with predictions from both SolidWorks finite element models and analytical solutions derived using Python scripts based on Mechanics of Materials theory. As part of this process, the uncertainty in DIC measurements will also be assessed to understand the precision and reliability of the data. In addition to DIC, the experiment utilizes an Instron testing machine to apply the compressive load. The DIC captures detailed, localized strain data that may reflect physical realities like minor imperfections or wear from repeated use.

Beyond validation, this setup is designed as an educational tool. Students will not only collect and analyze their own experimental data using DIC but will also build their own SolidWorks models and apply the provided Python code to interpret mechanical behavior. In doing so, students will develop critical skills in experimental mechanics, data analysis, and model verification, skills that are essential for future engineers tasked with interpreting physical data and designing reliable, real-world systems.

2. Experimental

The experimental setup consisted of an Instron Universal Testing Machine equipped with a three-point bending fixture, along with a Correlated Solutions VIC-EDU Digital Image Correlation (DIC) system, which included a stereo camera pair and VIC-Snap/VIC-3D software for full-field strain measurement. Beam specimens made of 6061 Aluminum were prepared with a matte white paint background and a random black speckle pattern for DIC tracking. A calibration target plate was used for camera calibration, and digital calipers were used to measure the beam dimensions before testing. Computer systems with Python and SolidWorks were used for data analysis and finite element modeling. A schematic of the experimental setup is shown in Figure 1, and a photograph of the actual setup is shown in Figure 2.

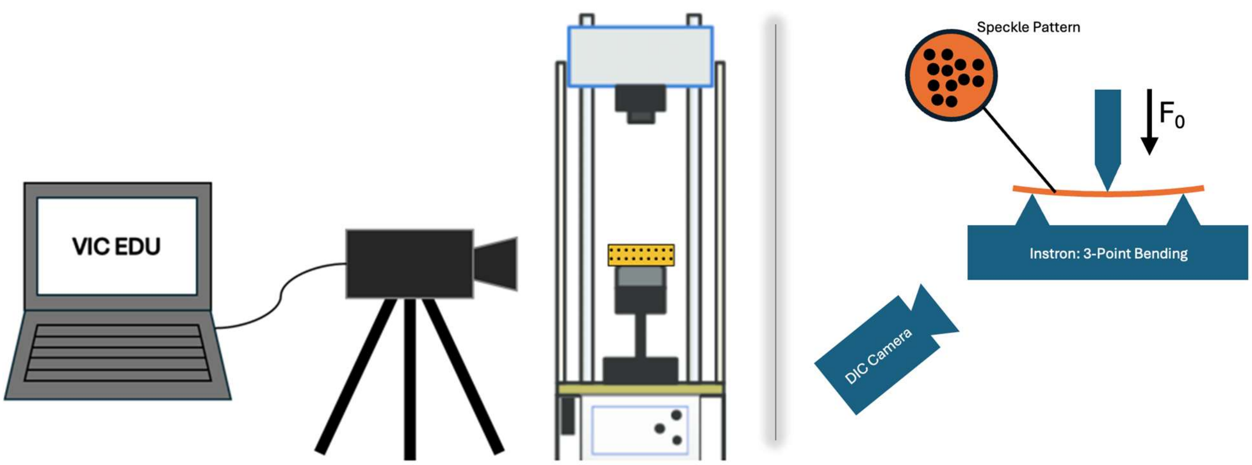


Figure 1: Schematic of the three-point bending experimental setup, showing the Instron testing machine, three-point bending fixture, beam specimen, and VIC-EDU DIC camera system positioned for full-field strain capture.

2.1 Equipment

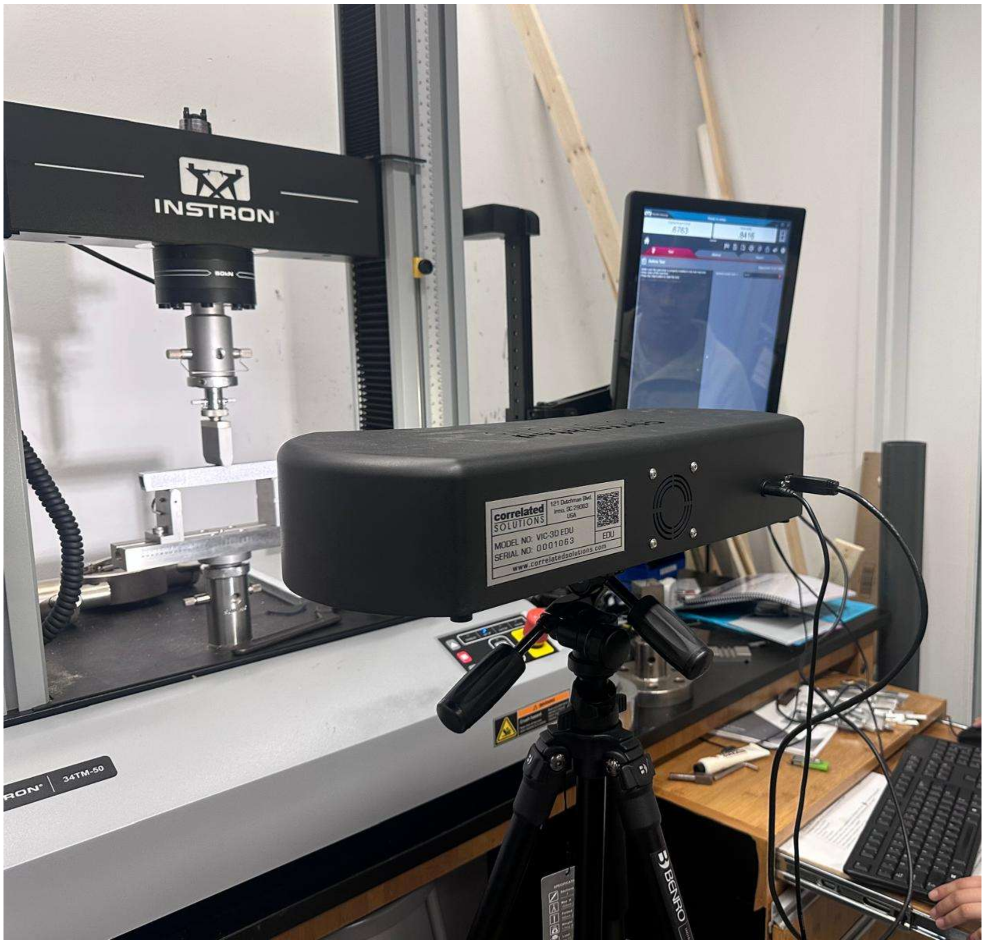


Figure 2: Photograph of the actual experimental setup, including the Instron machine, mounted beam specimen, DIC cameras, and computer system used for image acquisition and data processing.

2.1.1 Instrumentation

- **Calipers –** Used to measure beam dimensions before testing.
- **Correlated Solutions VIC-EDU Digital Image Correlation System –** Included stereo cameras and VIC-Snap/VIC-3D software for full-field strain measurement.
- **Computer with Python and SolidWorks–** For comparing experimental data with theoretical predictions.

2.1.2 Apparatus

- **Beam Samples –** Machined rectangular cross-section beams used as test specimens. The samples should be a minimum of 12cm in length and must not exceed 30cm. The recommendation is to use Aluminum 6061 because it is inexpensive and readily available, though other linear elastic materials can be used.

- **Instron Universal Testing Machine (Model 34TM-50)** – Applies controlled compressive loads at the center of the span.
- **Computer with Instron Software, VIC-Snap 9, and VIC-3D Software**– For performing the experiment and capturing and processing the images.
- **Three-Point Bending Fixture –** Adjustable support fixture installed onto the Instron with two lower supports a fixed distance apart and a central loading nose.
- **Calibration Target Plate –** Used for camera calibration prior to data collection.
- **Matte White Spray Paint –** Used to create a background that will contrast with the speckle pattern and avoid reflecting light into the lens of the camera.
- **Black Speckle Patterning Tool and Ink Pad –** Used to create a high-contrast random surface pattern on beam specimens.

2.2 Procedure

1. Cut rectangular beam samples from aluminum stock using a bandsaw or milling machine, ensuring consistent dimensions across all samples.
2. Lightly sand the surface of each sample to remove oxidation and promote paint adhesion.
3. Clean sample with rubbing alcohol or soapy water to remove and grease or oil.
4. Apply a thin, even coat of matte, white spray paint to the surface where strain will be analyzed. Let dry for at least 24 hours to avoid smudging during testing.
    - Note: It is recommended to paint at least 2 of the 4 sides of the beam in case the speckle pattern is not correctly applied on the first attempt.
5. Once dry, apply a random speckle pattern using the roller stamp and ink pad. The speckle density should be fine and consistent to allow for accurate DIC tracking. See Figure 4.
    - Do not touch the speckle pattern as it may smudge.
    - If the pattern is too sparse or smudged, rotate to another clean, painted side and attempt to reapply the speckle pattern.
6. Mount the beam horizontally in a three-point bending fixture on the Instron testing machine, ensuring the span of the supports are aligned with the desired test length and that the beam is centered on the test stand. Make sure the loading head will contact the top face of the beam evenly to avoid unintended torsional effects.
7. Set up the Tripod for the camera and attach the camera to the tripod.
    - The height of the tripods should be adjusted such that the lenses of the cameras should be aligned with the height of the sample and should not be looked up or down at the sample.
8. Connect the Correlated Solutions VIC-EDU Camera system with the included power and USB cables.

- Note: The license for the software is stored on the DIC device, the software won't be accessible if the camera is not plugged in via USB.

9. Open the Vic-Snap 9 software and create a new project folder to store captured images. Set this folder as the working directory.
    - VIC-Snap is used to take images, VIC-3D (as explained later) will be used to process the images and extract data.
10. Using the camera view in VIC-Snap, position the camera such that both cameras can see the entire area that is to be analyzed. Further ensure that both supports can be seen across the 2 views (1 support should be seen close to the edge of the frame in each camera).
    - Note: As of May 2025, the floor is marked with 3 pieces of blue tape which represent the recommended initial starting position of the camera. This is to be used as a helpful starting reference point and the camera and tripod should be moved from there to meet the above specifications.
11. The next step is to calibrate using the calibration plate, hold the calibration plate at the same distance as the sample, facing the camera. Avoid covering calibration dots with your fingers. Use the Calibration Images mode in VIC-Snap 9 to collect the calibration images.
12. Adjust the camera gain (the blue slider) in VIC-snap so that no red areas appear in the view.
13. Capture at least 15–20 calibration images by selecting the calibration mode and pressing the space bar to capture each image, rotating and translating the plate between shots to ensure full coverage and 3D calibration accuracy.
    - Increasing the number of calibration images taken will lower the projection error of the camera.
    - Futher details regarding calibration can be found in the Correlated Solutions VIC-EDU Manual.
14. In Vic-Snap, switch to Speckle Images mode.
15. Perform a hand-calculation to determine the max load that could be applied before the beam reaches the plastic region.
    - Note: The python script includes these calculations and will serve as a check. This value, divided by a factor of safety, will be used as the maximum limit for the force applied by the Instron.
16. Decide on a reasonable load increment value such that there will be 20 evenly spaced load increments, this will depend on the maximum load calculated.
    - For example, if you have a max load of 4000 N, the load should be in 200 N increments starting from 0 N.
17. With no load applied, capture a reference image of the undeformed sample.
18. Under the supervision of an instructor or qualified laboratory technician, use the Instron software to apply force incrementally at the center of the beam.

- The maximum load applied should be approximately 50% of the calculated load. This is a reasonable factor of safety that will ensure that the beam does not undergo plastic deformation.
- Do not exceed the material’s yield strength, **the goal is to remain within the linear elastic region**.

19. After each static load increment, pause the Instron and capture a corresponding image.
    - Avoid taking images during active motion to ensure frame alignment.
    - Make sure to take note of the load the Instron applies. It will be similar, but not exactly the same as the load prescribed.
20. Continue the loading sequence until sufficient data points across the elastic range are captured. Stop well before plastic deformation to preserve the sample for repeated testing.
21. Release the load on the sample and take a final image again under a zero-load condition.
22. Open Vic-3D and load the previously captured calibration images. Select Calibrate stereo system from the Calibration menu and verify that the calibration score is low (Ideally less than 0.1). Recalibrate if necessary.
23. Load the full set of speckle images taken from VIC-Snap into VIC-3D.
24. Use the rectangular Area of Interest (AOI) tool to select the region you wish to analyze.
    - Note: This AOI should cover the portion of the beam beginning at one support and end at the other. Also, it should cover both the top and bottom edge for the best results.

25. Use the Suggest Subset Size tool to choose an appropriate subset and step size.
    - The software generally does a good job at selecting an appropriate subset, so one can usually use the suggested values.

26. Run the DIC analysis by pressing the green run bottom and inspect the resulting strain fields.

27. Save all project files and export relevant data for reporting.
28. If desired, samples can be unloaded and reused in future trials, provided they remain undamaged and within the elastic regime.
29. Exporting the data: The data can be exported by selecting the Export item from the data menu. Export the data as a comma-separated variable.

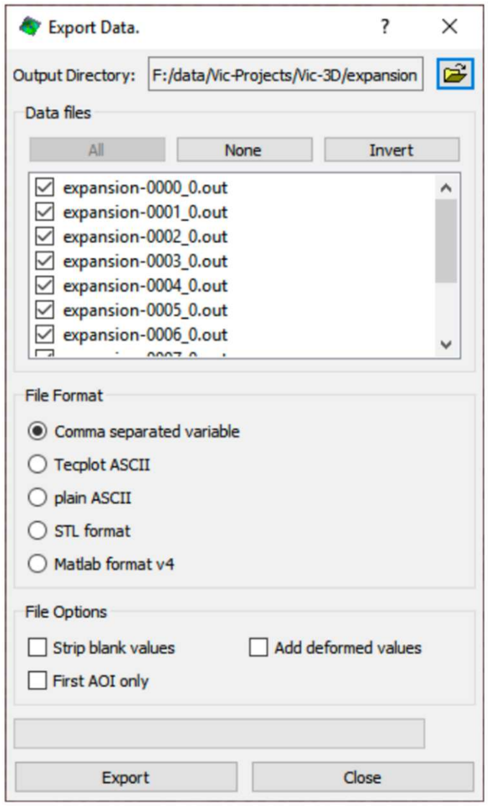


Figure 3: Export Instructions for Entire Strain Field

30. Use the Python scripts (Appendix IV) to analyze the DIC-generated strain fields and strain distributions across the beam.
31. Compare these experimental results with:
    - Theoretical stress and strain distributions calculated using a provided Python script based on beam theory. The code is modular and properties can be modified to fit the beam being tested.
    - Finite Element Analysis results from a SolidWorks Simulation of the same test configuration. (i.e. create a SolidWorks model of the 3-point bending setup and simulate the experiment at the loads tested). General guidelines for developing the FE model for a beam in 3 point bending can be found in Appendix V.
32. Evaluate DIC accuracy and uncertainty by analyzing noise in unloaded frames and comparing results to theory. This assessment helps quantify measurement precision and interpret how accurate both the DIC results and the Mechanics of Materials approximations are with respect to one another.

3. Results

The results of the three-point bending experiment are presented in this section. The analytical Mechanics of Materials (MechMat) predictions, experimental Digital Image Correlation (DIC) measurements, and SolidWorks finite element simulations were compared to evaluate the beam's behavior within the linear elastic regime. The purpose of this study was not only to validate the linear elastic behavior under a single loading condition, but also to demonstrate the modularity of the experimental setup and MechMat analysis for different beam geometries, applied loads, and evaluation positions (x-coordinates) along the beam.

Beam "Name"	Outer Width and Height (in)	Inner Width and Height (in)
Small Solid Beam	0.625	N/A
Hollow Box Beam	0.75	0.50
Large Solid Beam	1.00	N/A

Table 1: Beam Dimension for the 3 Beams Tested. All Beams are Square (H=W). All Beams have a Length of 16 cm (6.29 inches) with Supports 12 cm Positioned Apart in the Three-Point Bending Rig.

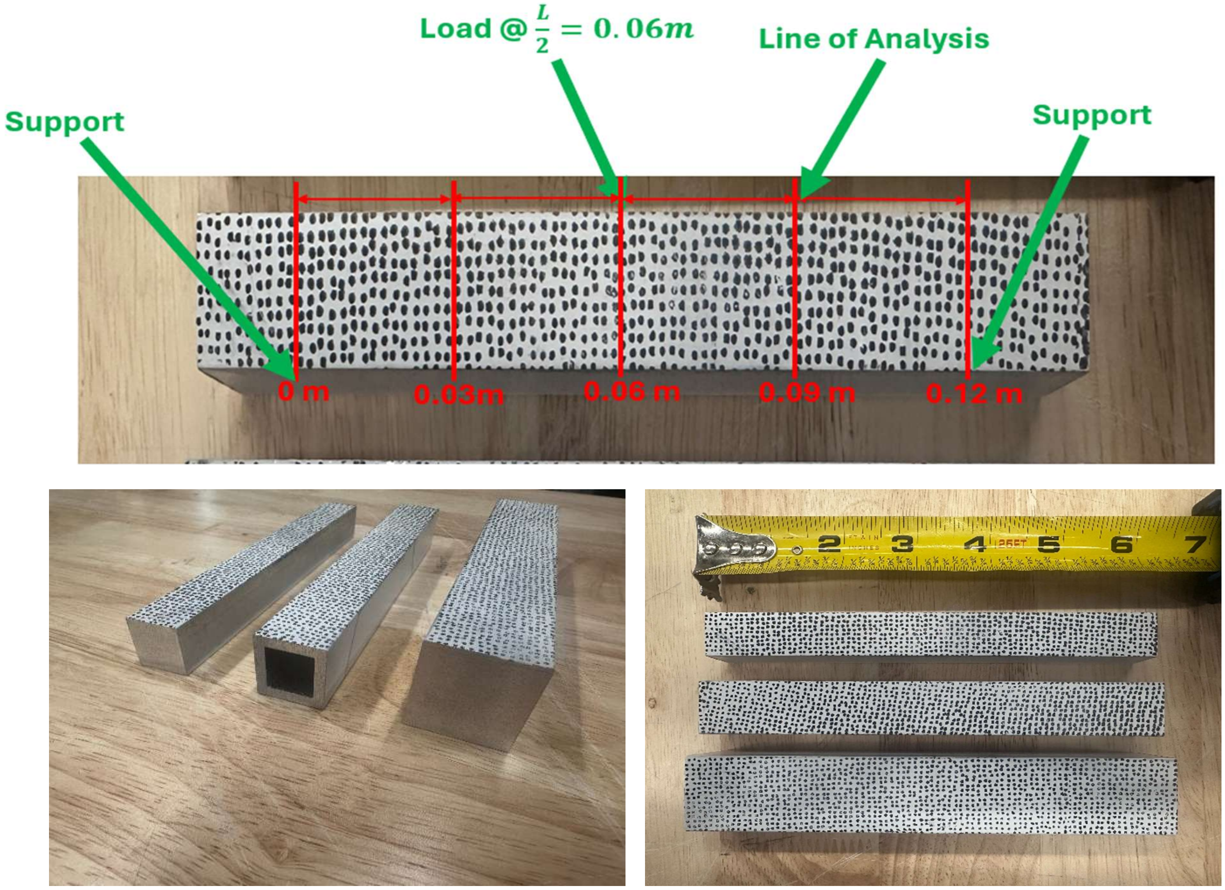


Figure 4: Speckle Pattern Stamped on Each Beam listed in Table 1. The top photo shows the 'Line of Analysis Method 'to Select X_coord in the MechMat and Matlab codes. Set X_coord = to the value on the x-axis for the vertical line 'y' you wish to analyze the strain curves at. IMPORTANT: Ffor the Matlab code, X_coord = 0 is the center of the beam while for the Mechmat Code, X_coord = 0 is at the leftmost support. This must be accounted for if using both scripts.

The data was post-processed in python and MATLAB. As previously mentioned, three aluminum beam samples, each 16 cm in length, were tested to evaluate strain under applied loads of 1000 N and 4000 N. The normal strain along the x and y directions, and the shear strain along the x-y plane were plotted (Figures 5-7).

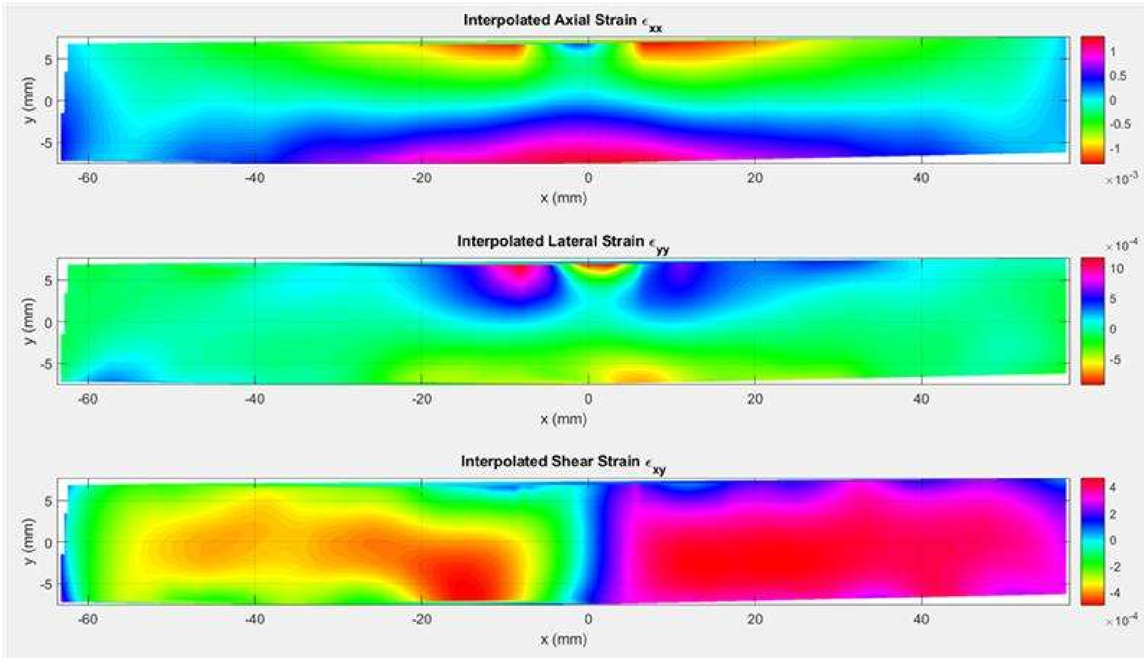


Figure 5: Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Hollow Box Beam in 3-Point Bending under a Load of 4000N processed from DIC experimental measurements.

Figure 5 shows the strain fields plotted from the DIC data collected for the hollow box beam in 3 point bending under a load of 4000 N. Peak tensile strain occurs at the bottom of the surface mid-span aligning with expected bending deformation. Peak compressive strain is highest at the top surface. The strain values for lateral strain and shear strain are also significantly lower than axial strain.

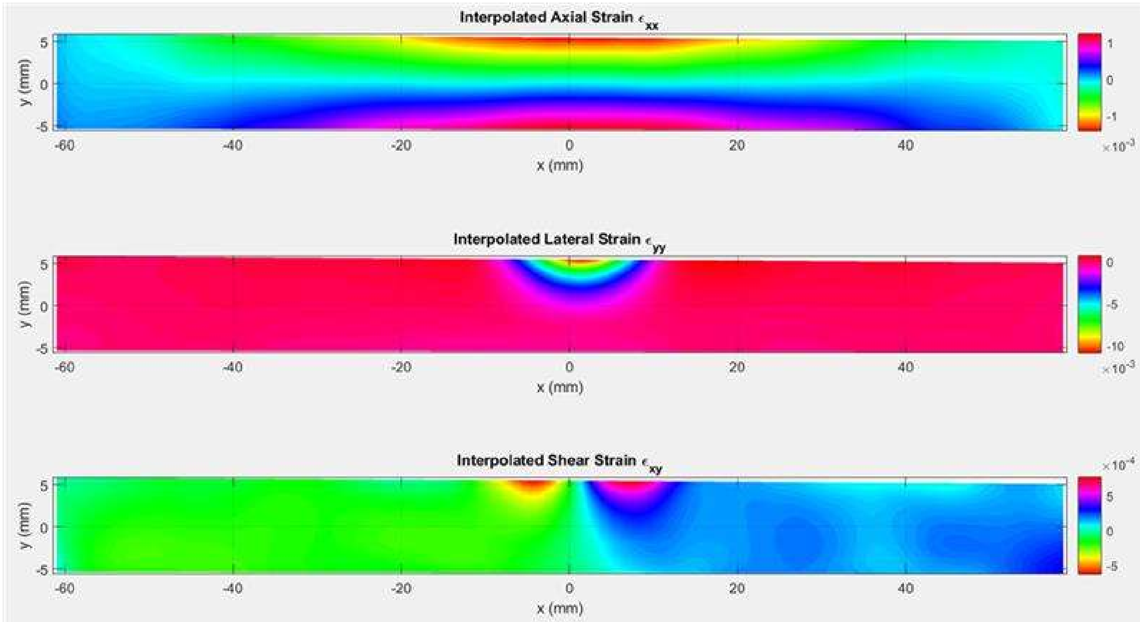


Figure 6: Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Small Solid Beam in 3-Point Bending under a Load of 4000N processed from DIC experimental measurements.

Figure 6 shows the strain fields plotted from the DIC data collected for the small solid beam in 3 point bending under a load of 4000 N. Compared to the hollow box beam (Figure 5) and large solid beam (Figure 7) the small solid beam showed the highest resolution, with peak tensile strain at the bottom of the beam and peak compressive strain at the top of the beam. Lateral strain shows the best resolution among the three strain fields.

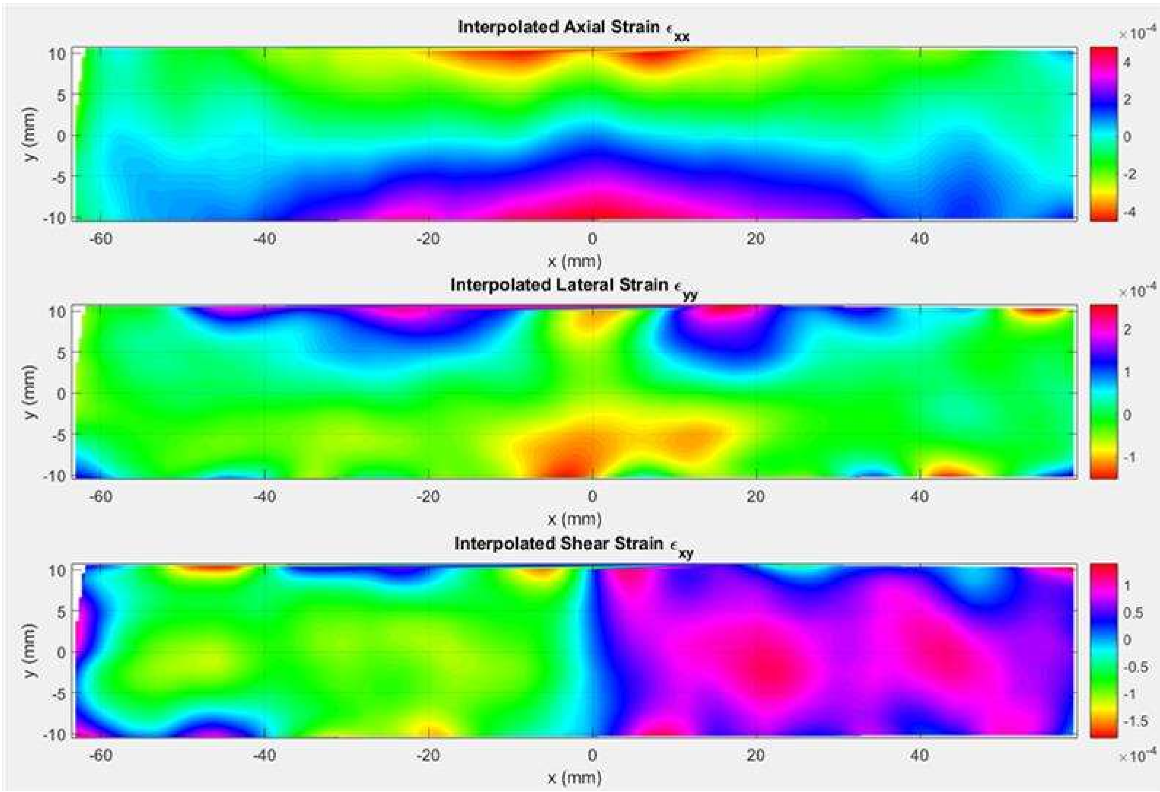


Figure 7: Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Large Solid Beam in 3-Point Bending under a Load of 4000N processed from DIC experimental measurements.

Figure 7 shows the strain fields plotted from the DIC data collected for the large solid beam in 3 point bending under a load of 4000 N. Compared to the hollow box beam and small solid beam, the large solid beam showed the worst resolution for both load cases of 1000N and 4000N and the most noise in the data. However, beam bending theory remains consistent with the peak tensile and compressive strain at the bottom and top of the beam respectively.

3.1 Mechanics of Material Solution

Analytical calculations were conducted using a custom Python script (Appendix IV), based on classical Mechanics of Materials theory for simply supported beams under a central load.

The bending moment, shear force, and vertical deflection are calculated across the beam using established formulas for simply supported beams with a central load:

- **Bending Moment**: $M(x) = \begin{cases} \frac{Fx}{2} & for\ 0 \leq x < \frac{L}{2} \\ \frac{F(L-x)}{2} & for\ \frac{L}{2} \leq x \leq L \end{cases}$
  - Linear variation with maximum at midspan.

- **Shear Force**: $V(x) = \begin{cases} \frac{F}{2} \ \ for\ 0 \le x < \frac{L}{2} \\ -\frac{F}{2} for\ \frac{L}{2} \le x \le L \end{cases}$
    - Constant magnitude, switching sign at the centerline.

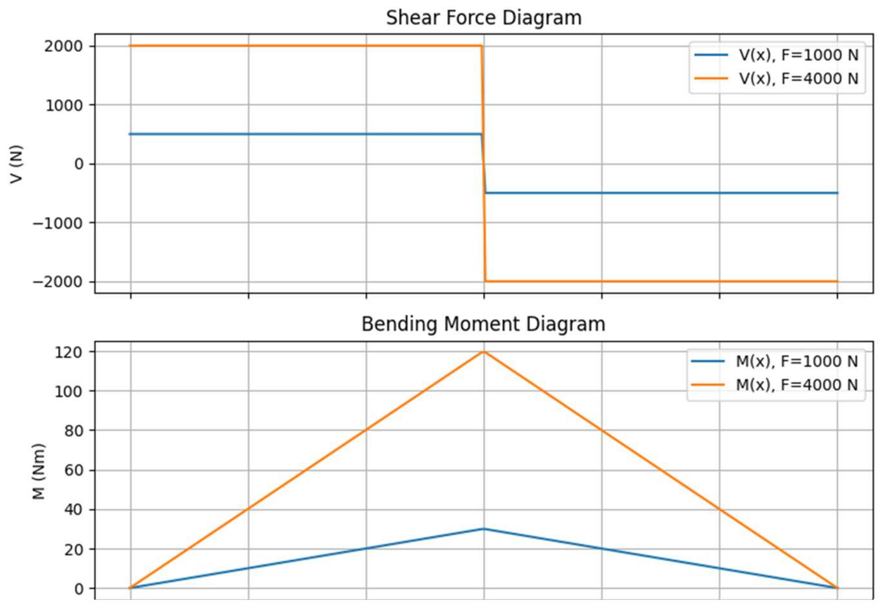


Figure 8: Shear Force and Bending Moment Diagrams Along the Length of the Beam. This Curve is Independent of the Material

- **Vertical Deflection:** $\delta(x) = \begin{cases} \frac{Fx(3L^2-4x^2)}{48EI} & for\ 0 \le x < \frac{L}{2} \\ \frac{Fx(3(L-x)^2-4x^2)}{48EI} & for\ \frac{L}{2} \le x \le L \end{cases}$
    - Symmetric about midspan with maximum deflection at the center.

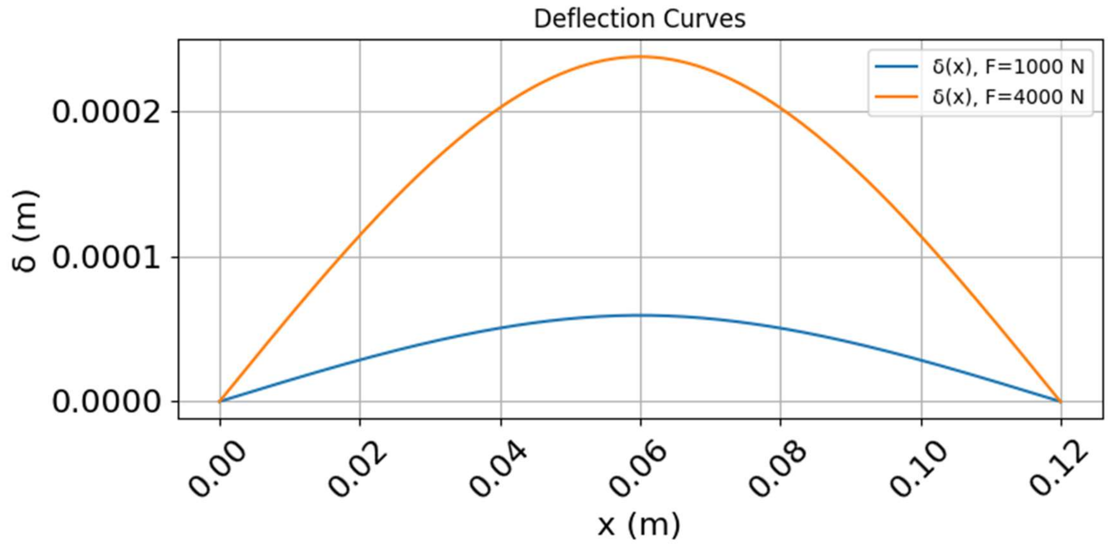


Figure 9: Deflection Along the Length of the Beam for the Hollow Box Beam

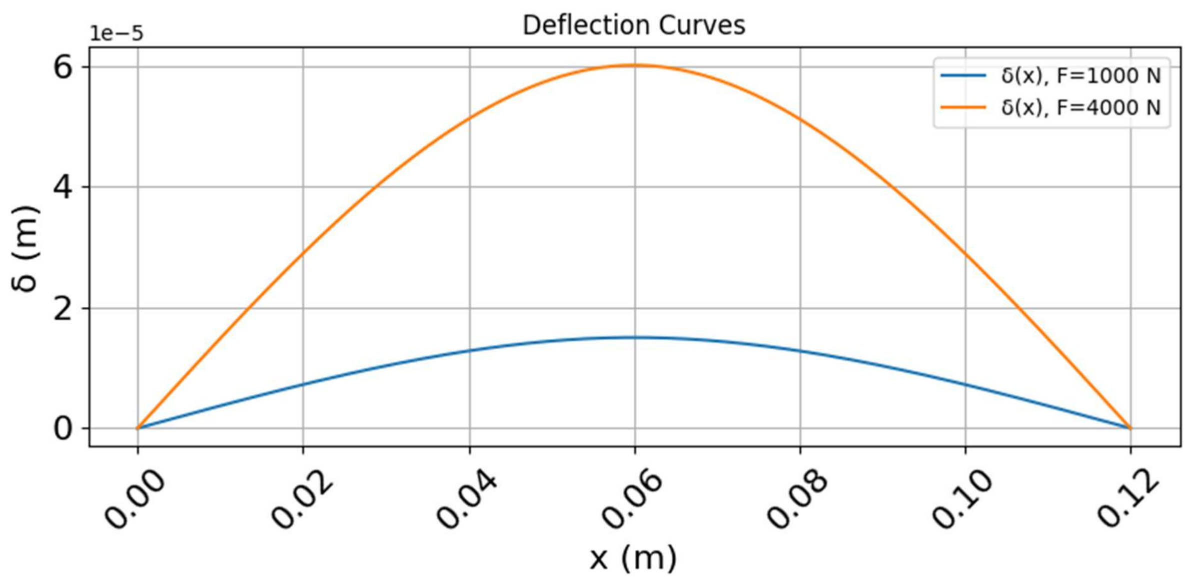


Figure 10: Deflection Along the Length of the Beam for the Large Beam

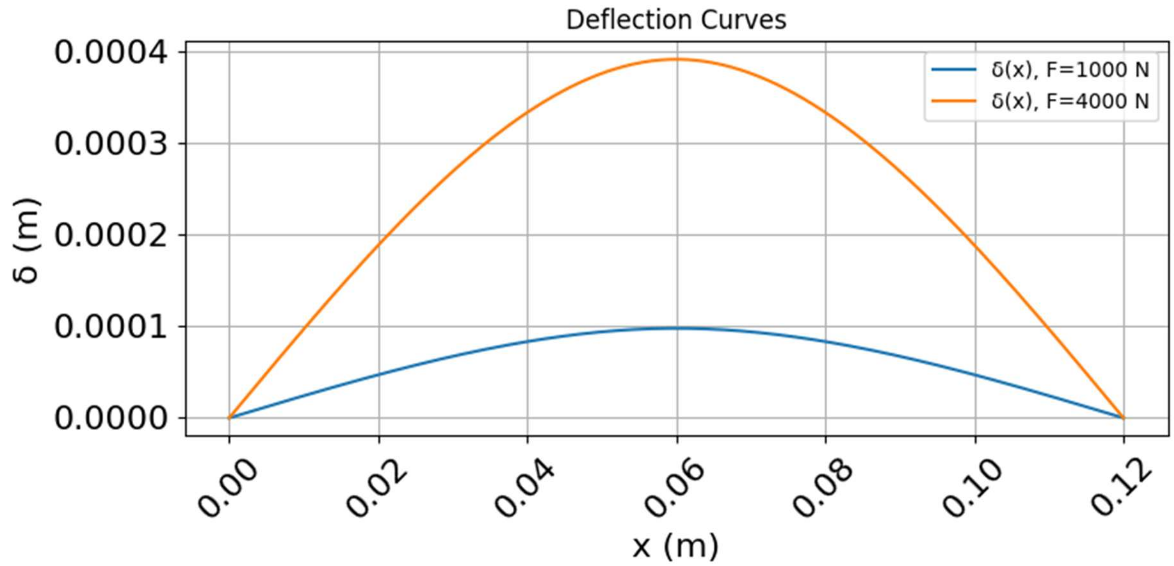


Figure 11: Deflection Along the Length of the Beam for the Small Beam

Plots of the bending moment and shear force diagrams are shown in Figure 4, and the vertical deflection curves are presented in Figure 9, Figure 10, and Figure 11 for the hollow box beam, small solid beam, and large solid beam, respectively.

At a selected position x_{coord} along the beam (either at the support, quarter-span, midspan or any other location of interest), detailed cross-sectional distributions are computed:

- **Axial Strain:** $\boldsymbol{\epsilon_{xx} = -\frac{yM}{EI}}$
  - Varies linearly across the beam height.
  - Maximum tension at the bottom fiber and maximum compression at the top fiber.

- **Lateral Strain:** $\boldsymbol{\epsilon_{yy} = -\nu\epsilon_{xx}}$
  - Induced via Poisson's effect; compression perpendicular to the applied load.
- **Shear Strain**: $\boldsymbol{\gamma = \frac{3}{2}\left(\frac{V}{wh}\right)\left(1 - \left(\frac{2y}{h}\right)^2\right)}$ **à** $\boldsymbol{\epsilon_{xy} = \frac{1}{2}\gamma = \frac{\frac{3}{2}\left(\frac{V}{wh}\right)\left(1-\left(\frac{2y}{h}\right)^2\right)}{2}}$
  - Parabolic distribution, peaking at the neutral axis and reducing to zero at the outer surfaces.

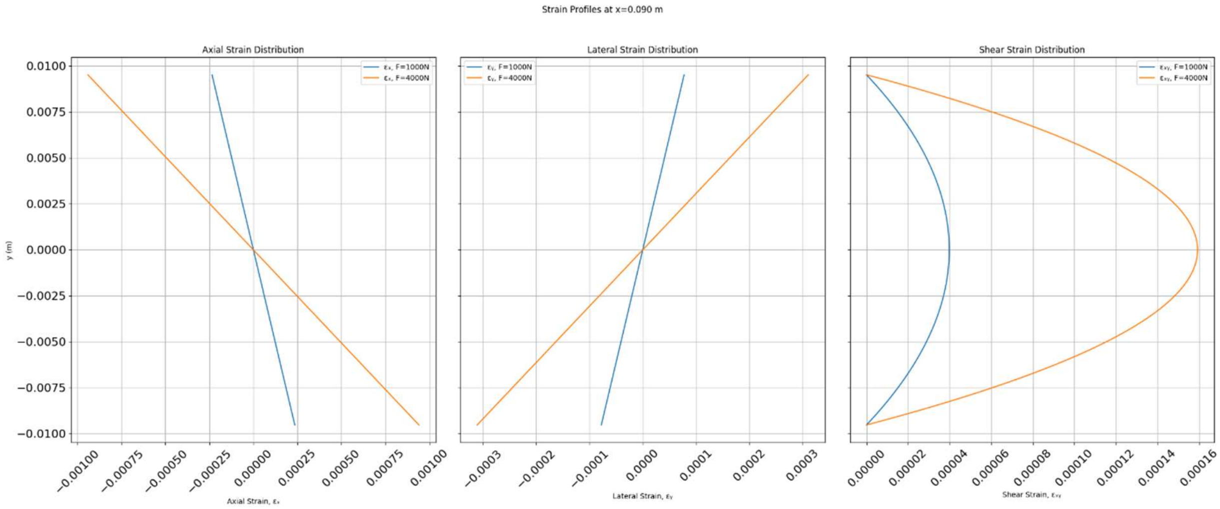


Figure 12: Strain vs. y at x= 0.09 m for the Hollow Box Beam

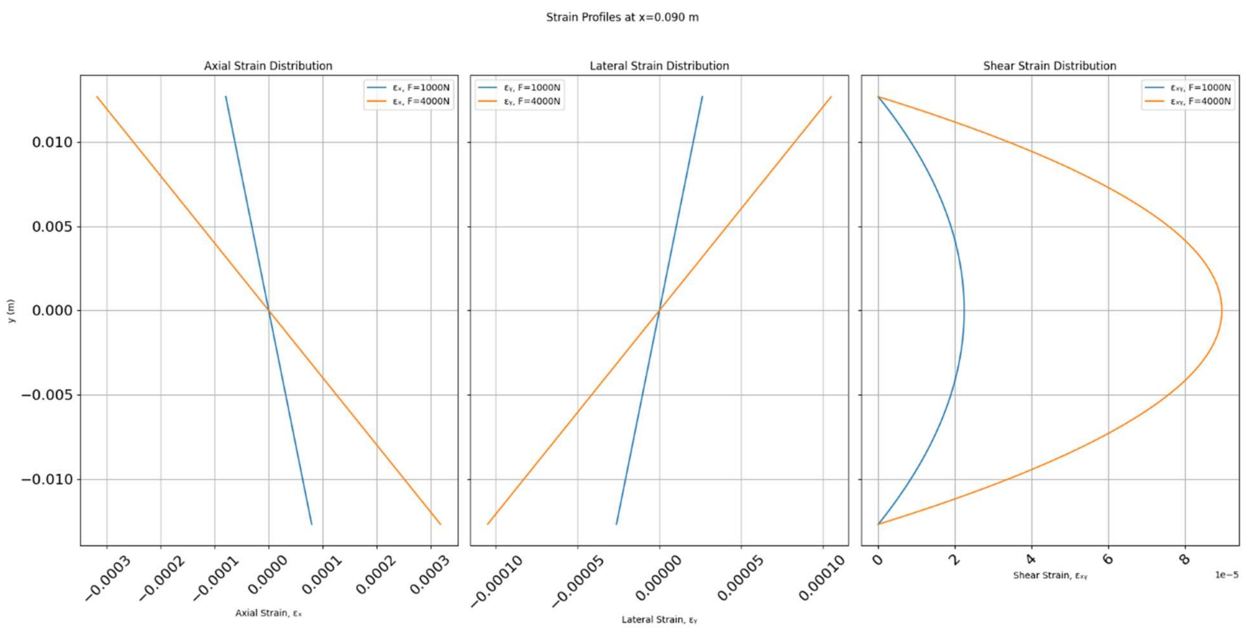


Figure 13: Strain vs. y at x= 0.09 m for the Large Beam

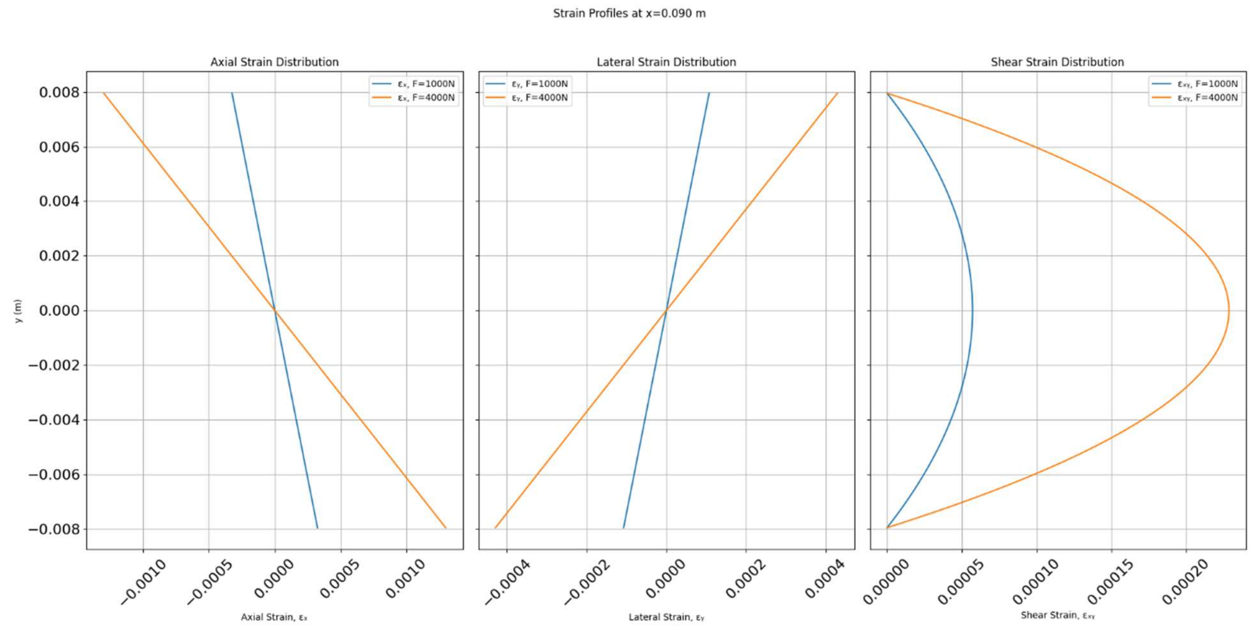


Figure 14: Strain vs. y at x= 0.09 m for the Small Beam

The corresponding stress fields can be computed by multiplying the Axial and Lateral strains by the Young's Modulus (E) and the Shear Strain by the Shear Modulus (G), such that:

- **Normal Stress:** $\sigma_{xx} = E\epsilon_{xx}$ due to bending.
- **Lateral Stress:** $\sigma_{yy} = E\epsilon_{yy}$ from Poisson's effect.
- **Shear Stress:** $\tau = G\gamma$ due to transverse shear forces.

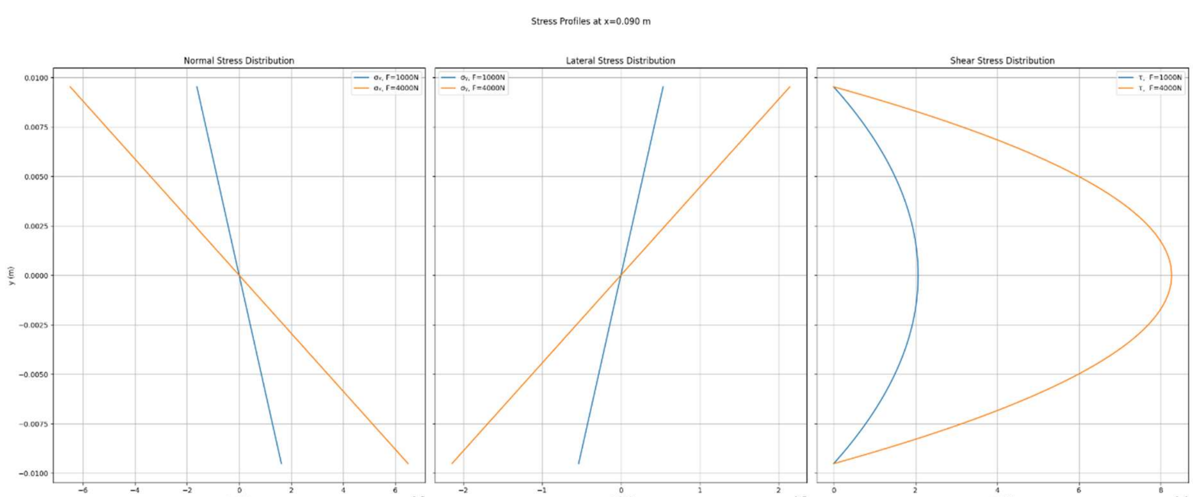


Figure 15: Stress vs. y at a x = 0.09 meters for the Hollow Box Beam.

Plots of these cross-sectional strain distributions are seen in Figures 12, 13, and 14 for each of the beams. Additionally, for simplicity, a single sample stress distribution is shown in Figure 15 for the Hollow Box Beam is shown as an example. While analogous curves exist for the other sample beams, they are not relevant for further analysis within this report and are therefore excluded.

The script automatically evaluates whether the applied load causes stresses that exceed the material's yield strength:

- The critical load P_{Yield} is calculated based on the material properties and beam geometry.
  - $M_{yield} = \frac{\sigma_{yield} \times I}{c}$ where $c = \frac{h}{2}$, the length from the neutral axis to the outer layers
  - $P_{yield} = \frac{4 \times M_{yield}}{L}$
- The maximum induced bending stress is compared to the yield strength.
  - $\sigma_{Max} = \frac{M(x) \times c}{I}$
- The beam is confirmed to be operating within the elastic regime if the bending stress is less than the yield strength. However, when testing, we use a factor of safety of 0.5 to ensure that local plastic deformation at the load point and supports is avoided.

To visualize the complete strain behavior along the entire beam span:

- 2D heatmaps of axial strain (ϵ_{xx}), lateral strain (ϵ_{yy}), and shear strain (ϵ_{xy}) are generated.
- These plots provide a full spatial picture of the mechanical behavior under load, highlighting maximum strains at midspan and diminishing strains near supports.

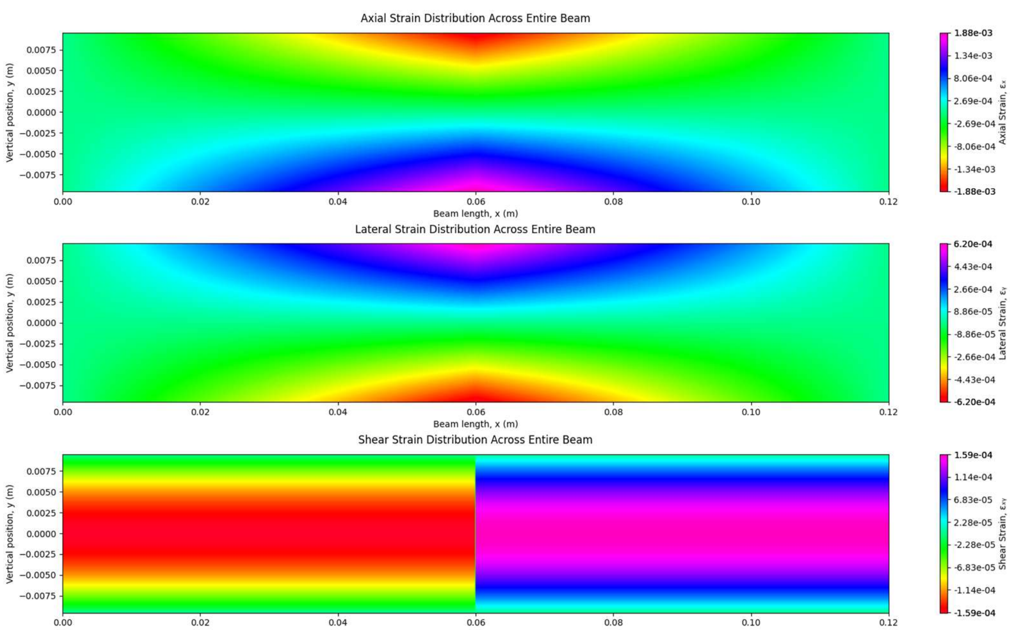


Figure 16: Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Hollow Box Beam in 3-Point Bending under a Load of 4000N.

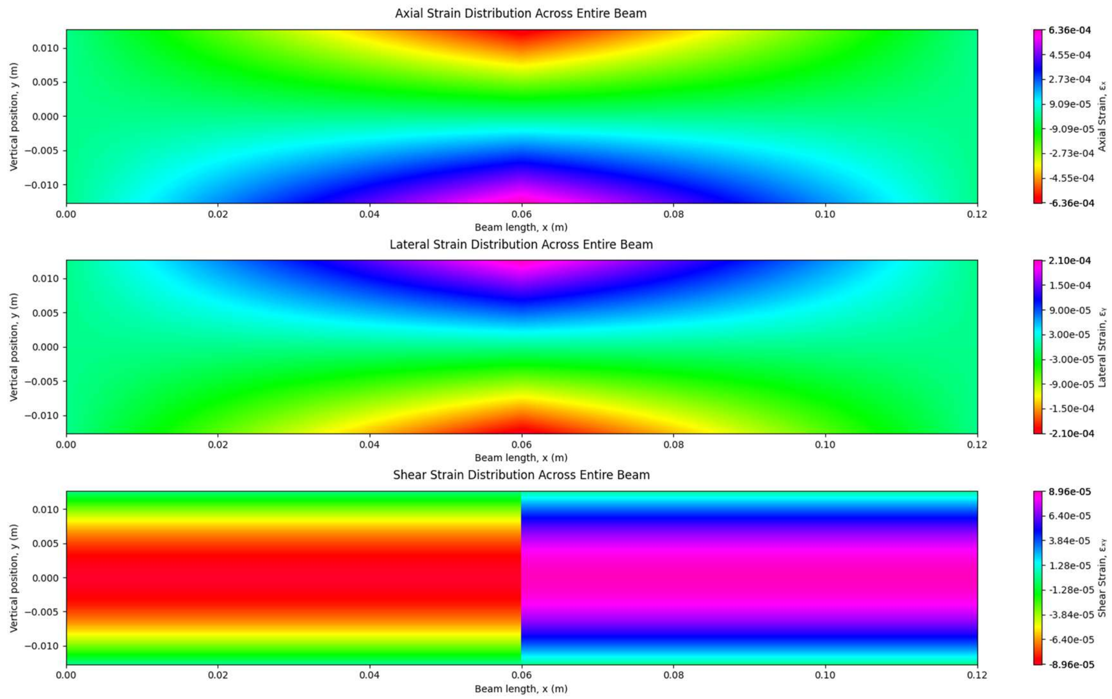


Figure 17: Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Large Beam in 3-Point Bending under a Load of 4000N.

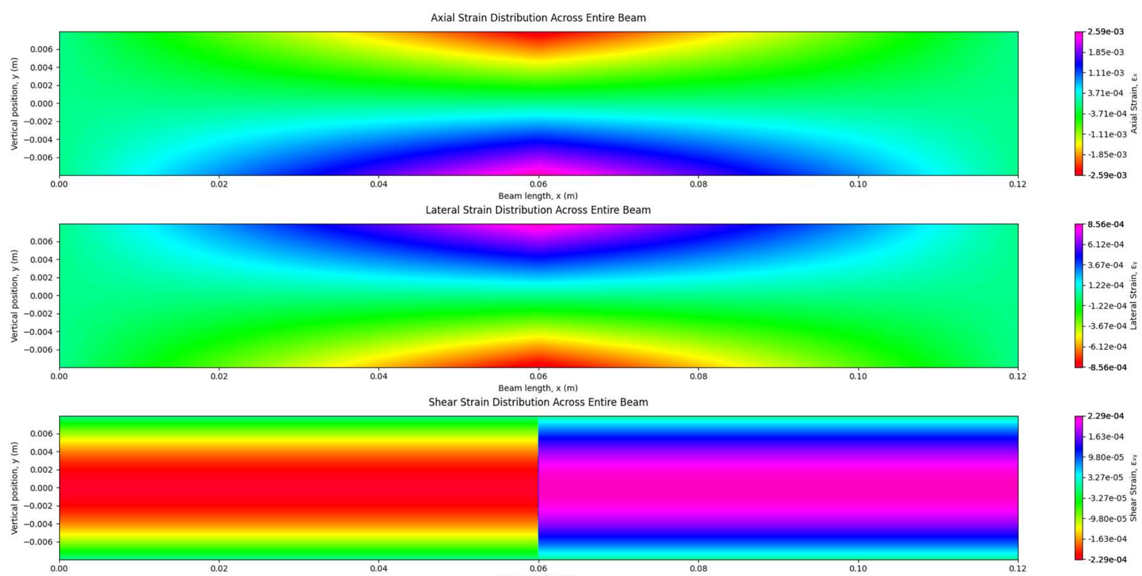


Figure 18: Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Small Beam in 3-Point Bending under a Load of 4000N.

The full-beam strain heatmaps are provided in Figures 16Figure 17, 17, and 18.

3.2 Finite Element Model

The Finite Element Model was developed using SolidWorks to simulate the strain response of the aluminum samples under 1000 N and 4000 N loading conditions. A finite element model for polyurethane was initially developed, however, due to the complexity of its material properties and the lack of predefined material data in SolidWorks, unlike aluminum, accurate modeling proved challenging. Simulations for the polyurethane sample can be found in Appendix V.

The analysis utilized a static loading scenario, applying a line load in the negative y-direction at half of the sample's length, to evaluate strain distribution. Boundary conditions were set to mimic the experimental constraints, ensuring an accurate comparison with the Mechanics of Materials solution and the DIC measurements. The simulation results presented in this section are shown using color bar limits set to the experimental values processed using MATLAB. This is done to compare the strain distribution patterns of each sample under loads of 1000 N and 4000 N. For reference, simulations with a true scale and deformed scale to better visualize the deformation of the samples are also provided in Appendix V. Figures 19 through 21 demonstrate axial (ϵ_{xx}), lateral (ϵ_{yy}), and shear (ϵ_{xy}) strain distributions for the hollow box, small solid, and large solid beams in 3-point bending under a load of 4000N each.

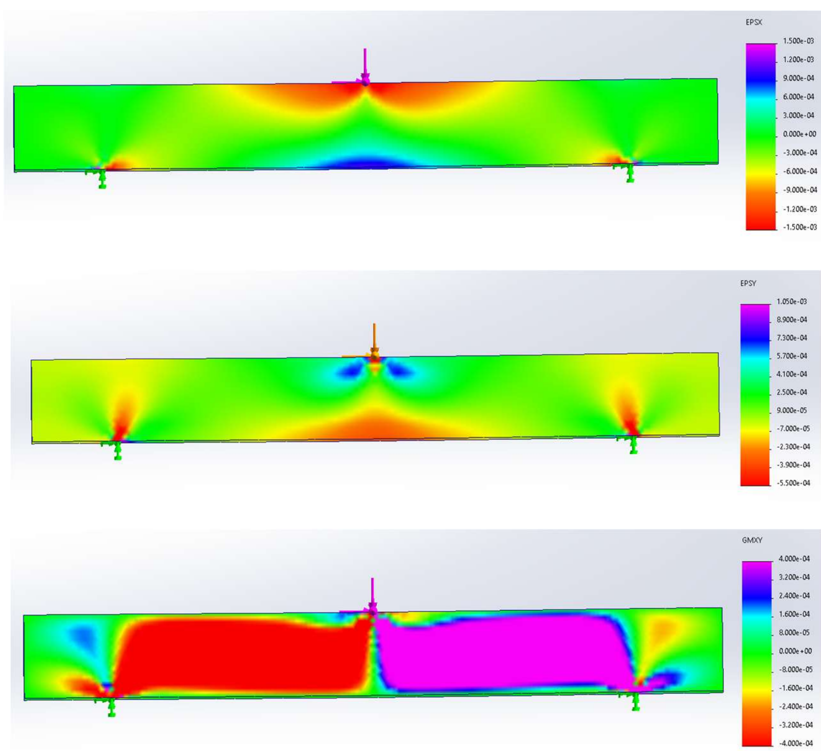


Figure 19: FE model of Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Hollow Box Beam in 3-Point Bending under a Load of 4000N.

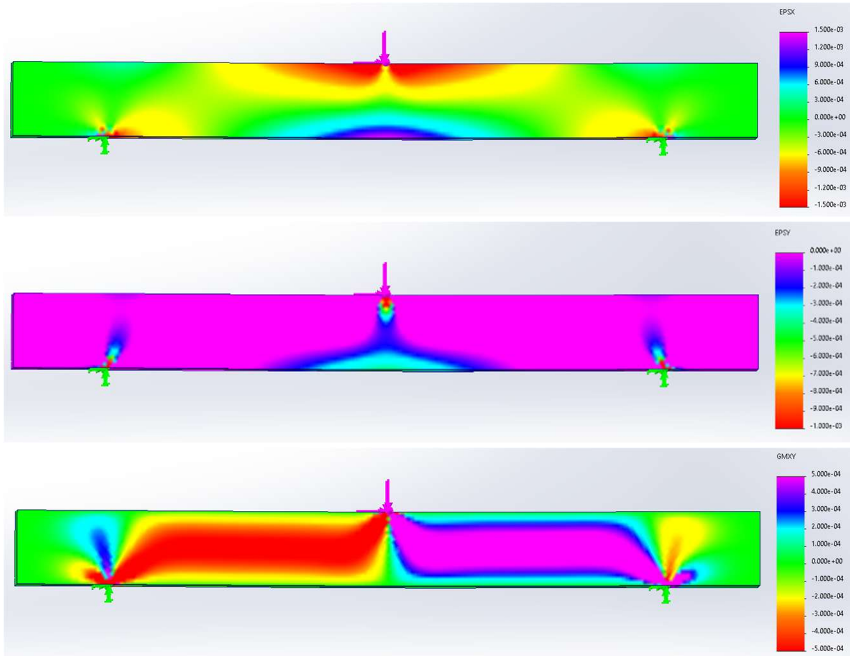


Figure 20: FE model of Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of Small Solid Beam in 3-Point Bending under a Load of 4000N

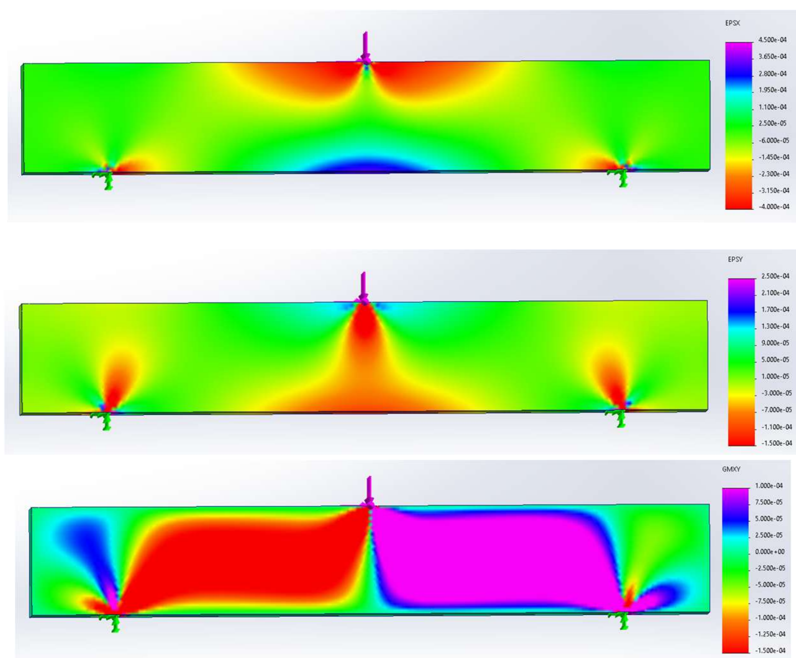


Figure 21: FE model of Axial (ϵ_{xx}), Lateral (ϵ_{yy}), and Shear (ϵ_{xy}) Strain Distributions Across the Length of the Large Solid Box Beam in 3-Point Bending under a Load of 4000N

The FEA results for the aluminum beams under 3-point bending with a 4000 N applied load (Figures 19 – 21) depict key similarities among the three models:

- Axial Strain (ε_{xx}): Peak tensile strain appears at the bottom surface near mid-span, while compressive strain is concentrated at the top surface.
- Lateral Strain (ε_{yy}): Minimal lateral deformation, with strain localized around the loading point.
- Shear Strain (ε_{xy}): Highest shear strain occurs near the supports and load application, showing expected stress concentrations.

Overall, the strain patterns confirm bending dominated deformation, with shear effects primarily near constraints. Moreover, the beam remains within the elastic range in the FE model.

3.3 Hollow Box Beam

This section compares strain patterns obtained from Digital Image Correlation (DIC) measurements with those predicted by a finite element model in SolidWorks and for a hollow box beam subjected to 1000 N and 4000 N loads. Note that the FE model captures strain fields across the entire beam length while the DIC and MechMat strain fields focus on the material within the supports.

Strain Distribution

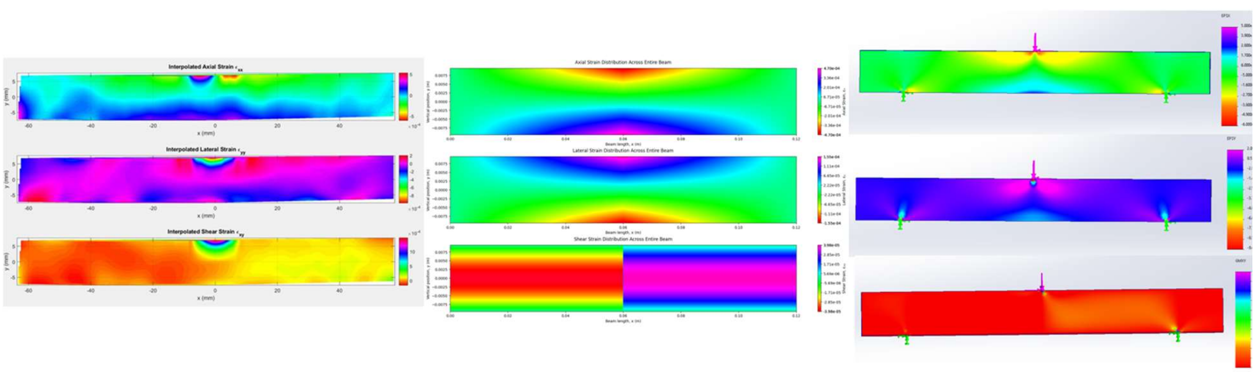


Figure 22: Axial (top row), Lateral (middle row) and Shear (bottom row) Strain Distribution for Hollow Box Beam under 1000 N Load generated from experimental values (left column), MechMat theory (middle column) and a FE model (right column).

Figure 22 depicts a comparative summary of axial, lateral, and shear strain distributions for the hollow box beam generated from experimental data collected, MechMat theory, and the FE model. Across all three plots for axial strain, peak tensile and compressive strains appear at the bottom and top of the beam respectively. In addition, lateral and shear strain remain minimal compared to axial strain. There is less resolution in the axial and lateral strain fields, with shear strain showing the best resolution.

4000 N Applied Load

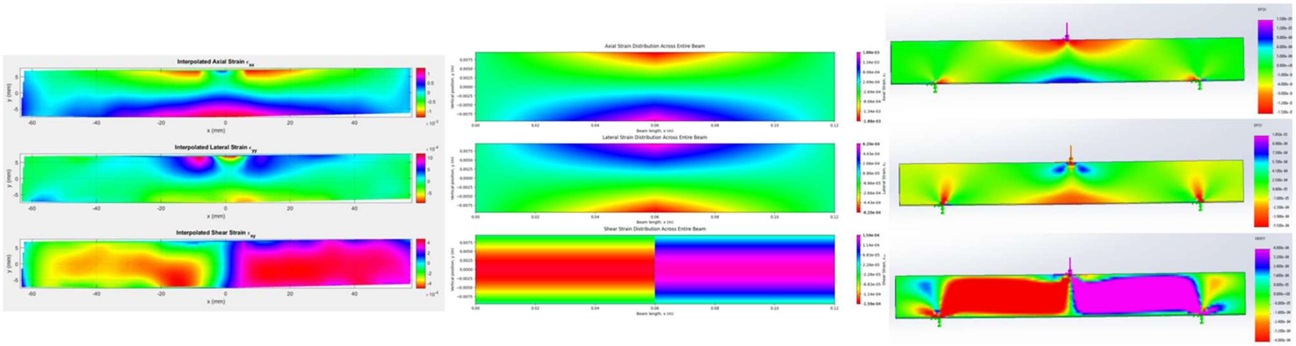


Figure 23: Axial (top row), Lateral (middle row) and Shear (bottom row) Strain Distribution for Hollow Box Beam under 4000 N Load generated from experimental values (left column), MechMat theory (middle column) and a FE model (right column).

Figure 23 compares strain distributions for the hollow box beam generated from experimental data collected, MechMat theory, and the FE model subjected to a 4000 N applied load. At 4000 N, the axial strain field resolution becomes finer although noise is still observably present.

Strain Curves at Specified X-Coordinate

This section shows the axial and shear strains at x = 3 m for 1000 N and 4000 N loads (Figure 24).

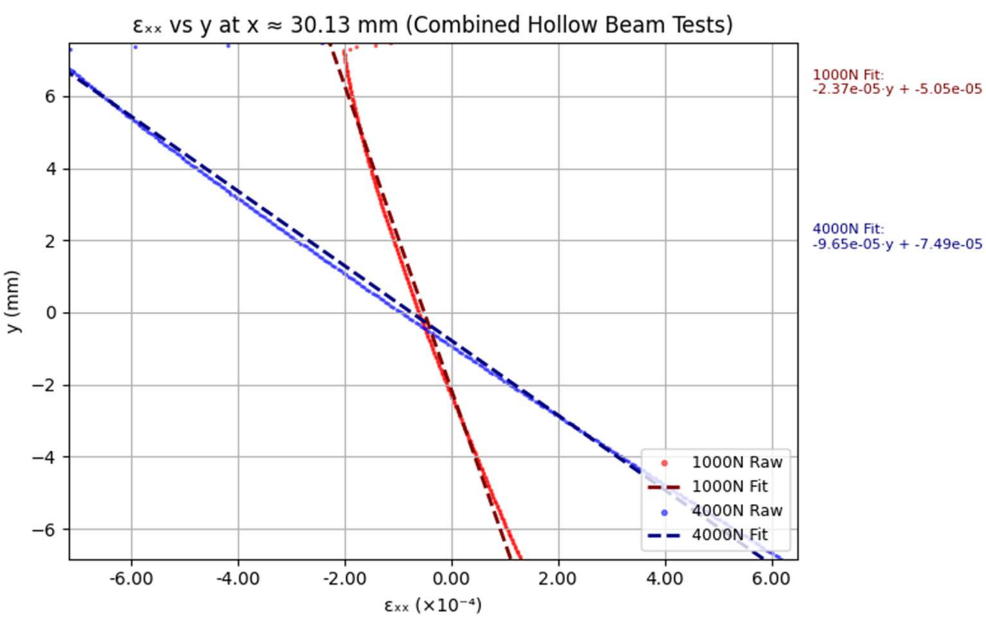


Figure 24: Axial strain vs y plotted at y = 30.13 meters from the load application.

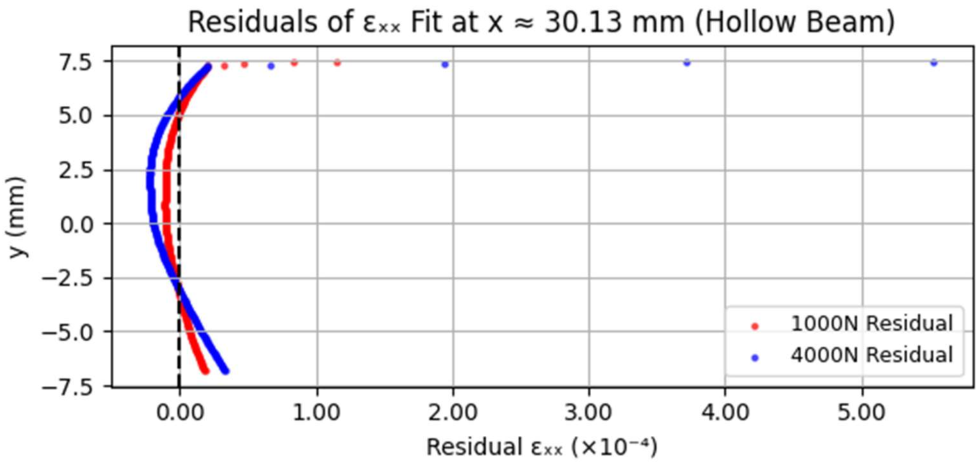


Figure 25: Residual scatter plot of axial strain vs y plotted at y = 30.03 mm from the load application.

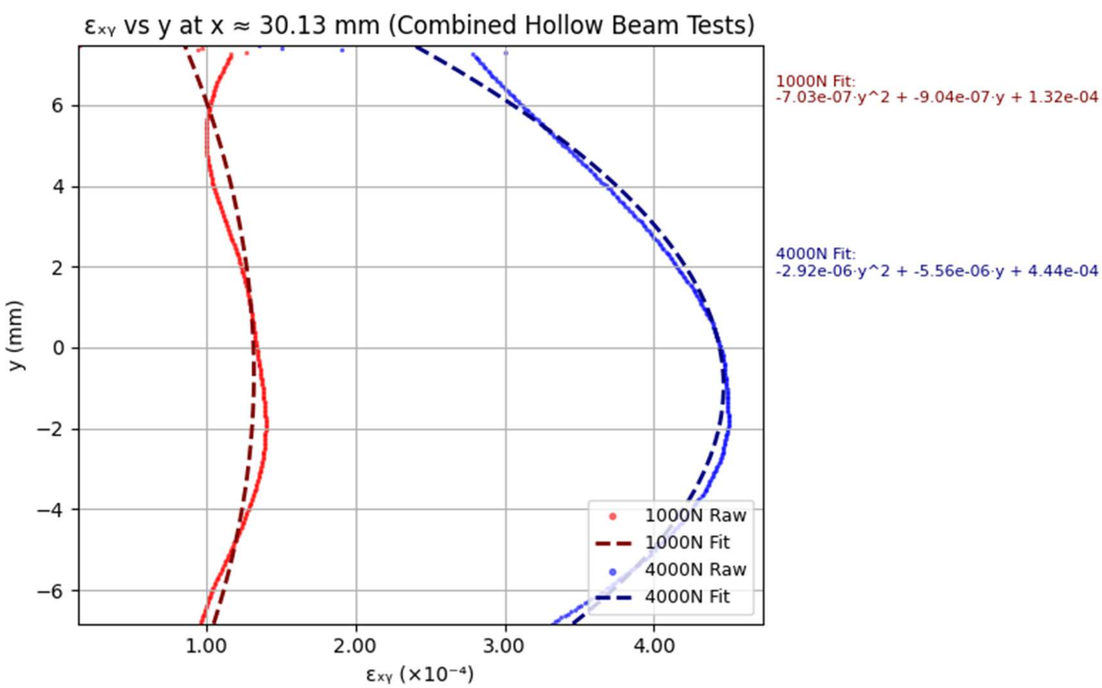


Figure 26: Shear strain vs y at x = 30.13 mm from the load of application.

3.4 Small Solid Beam

This section compares strain patterns obtained experimental measurements with those predicted by a finite element model in SolidWorks for the small solid beam subjected to 1000 N and 4000 N loads.

Strain Distribution

1000 N Applied Load

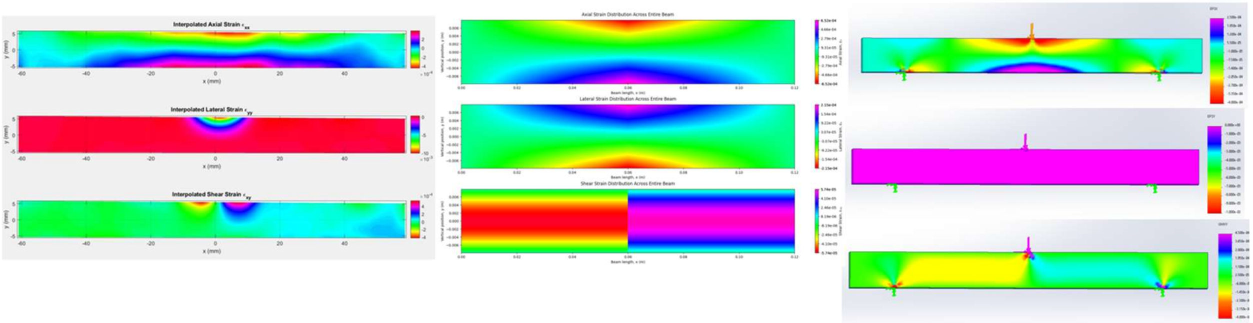


Figure 27: Axial (top row), Lateral (middle row) and Shear (bottom row) Strain Distribution for the Small Solid Beam under 1000 N Load generated from experimental values (left column), MechMat theory (middle column) and a FE model (right column).

Figure 27 depicts strain distributions for the small solid beam generated from experimental data collected, MechMat theory, and the FE model subjected to a 1000 N applied load. Axial strain follows expected trends, with tensile strain concentrated along the bottom mid-span and compressive strain near the top surface. The FE model shows a smooth strain gradient across the full beam length, while DIC data captures localized deviations within the supported region, highlighting minor asymmetries due to real-world imperfections as well as noise present in the data. Lateral strain across all three plots remains minimal, with slight expansion near the load application point. Shear strain for the DIC, MechMat, and FE model plots displays peak values at the point of loading. The FE model also shows peak strain values near the supports, where reaction forces induce material rotation.

4000 N Applied Load

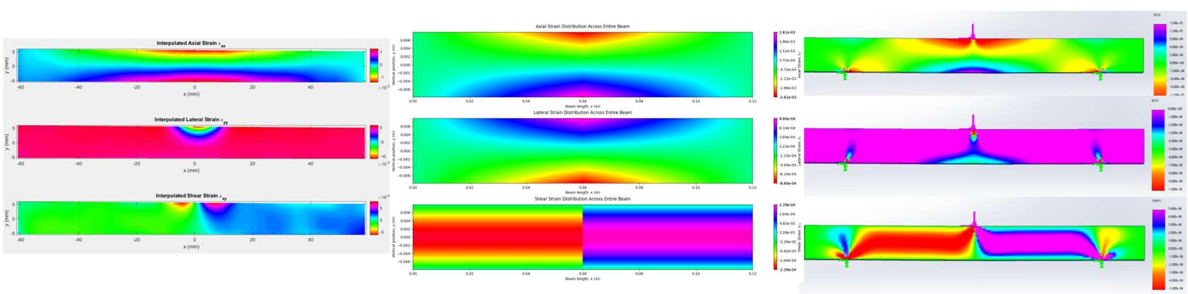


Figure 28: Axial (top row), Lateral (middle row) and Shear (bottom row) Strain Distribution for Small Beam under 4000 N Load generated from experimental values (left column), MechMat theory (middle column) and a FE model (right column).

Figure 28 depicts strain distributions for the small solid beam generated from experimental data collected, MechMat theory, and the FE model subjected to a 4000 N applied load. At a higher load, there are higher magnitudes of strain values which result in less noise in the DIC data resulting in finer resolution across all three strain fields for the small solid beam.

Strain Curves at Specified X-Coordinate

This section shows the axial and shear strains at x = 30mm for 1000 N and 4000 N loads (Figure 29).

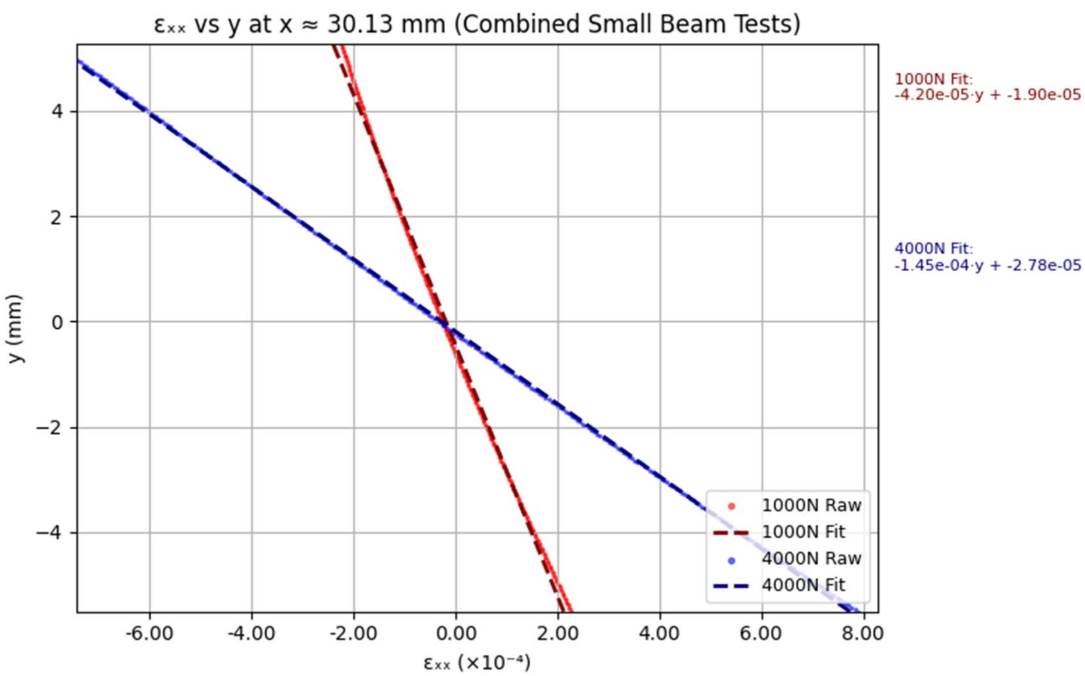


Figure 29: Axial strain vs y plotted at x = 30.13 mm from the load application.

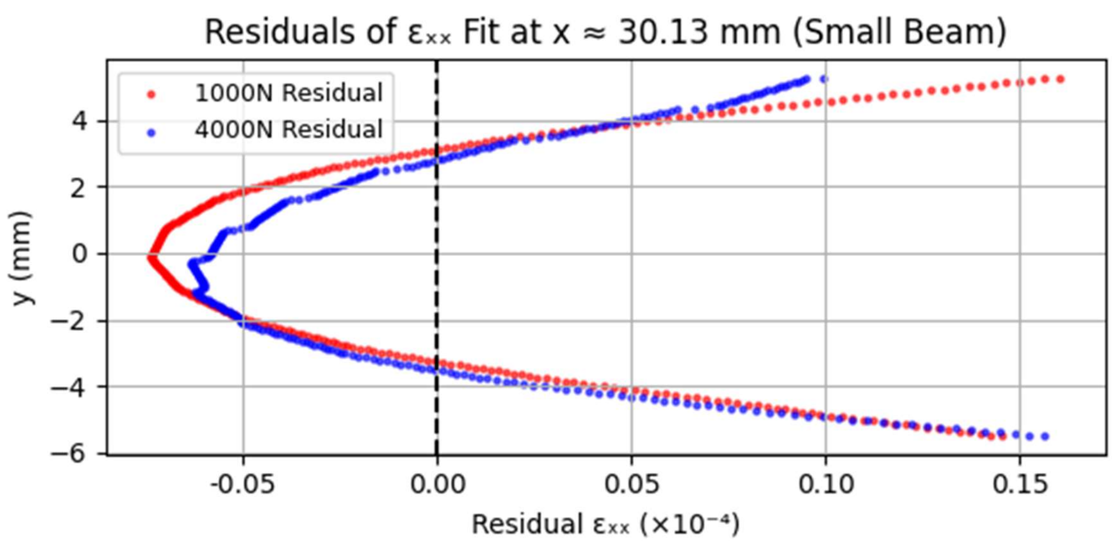


Figure 30: Axial strain vs y plotted at x = 30.03 mm from the load application.

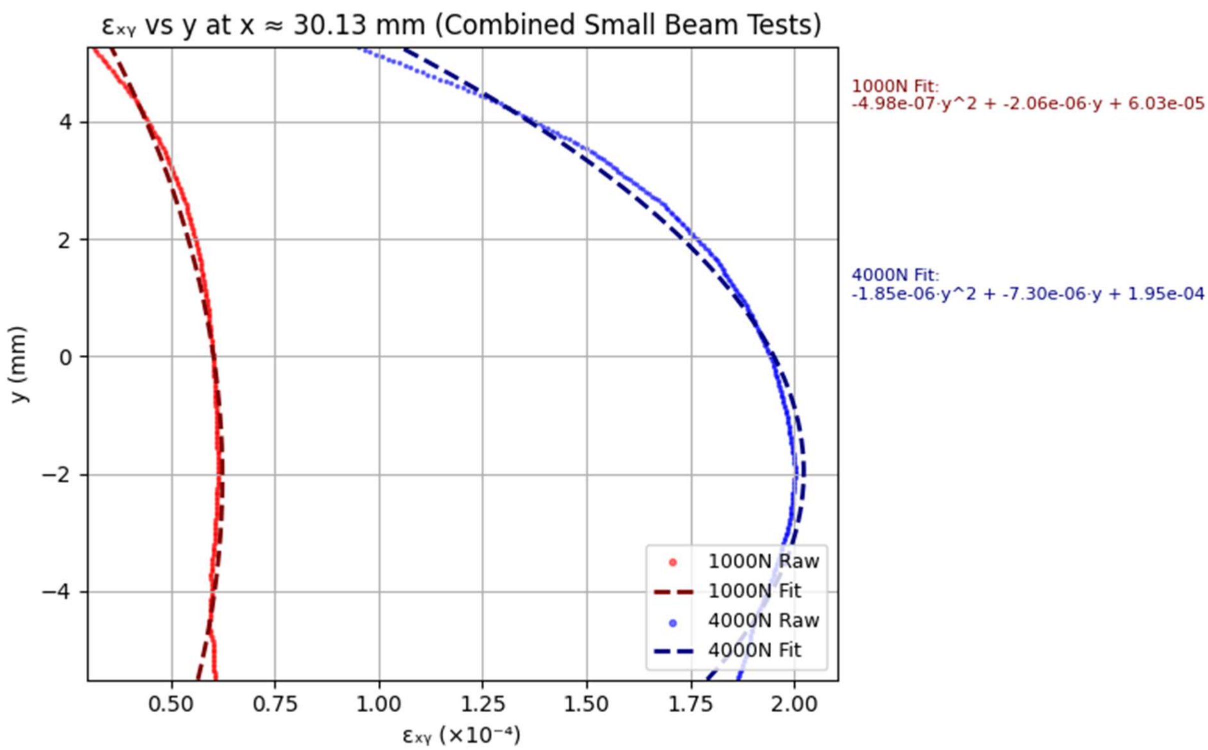


Figure 31: Shear strain vs y plotted at x = 30.13 mm from the load application.

3.5 Large Solid Beam

This section compares strain patterns obtained experimental measurements with those predicted by a finite element model in SolidWorks for the large solid beam subjected to 1000 N and 4000 N loads.

Strain Distribution

1000 N Applied Load

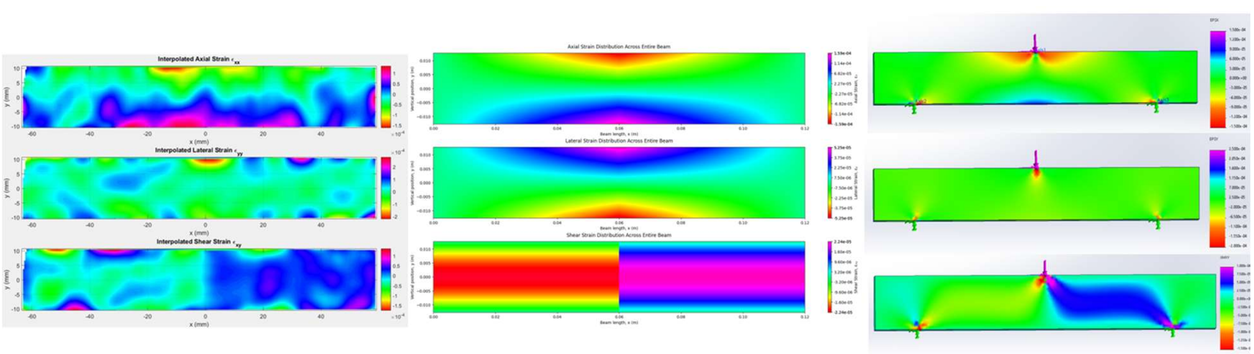


Figure 32: Axial (top row), Lateral (middle row) and Shear (bottom row) Strain Distribution for the Large Solid Beam under 1000 N Load generated from experimental values (left column), MechMat theory (middle column) and a FE model (right column).

Figure 32 depicts strain distributions for the large solid beam generated from experimental data collected, MechMat theory, and the FE model subjected to a 1000 N applied load. The large solid beam at 1000 N shows lower magnitudes of strain values, resulting in the worst resolution across all trials and samples. Moreover, the strain patterns for the DIC data are slightly asymmetrical whereas the MechMat theory and FE model predict smoother strain distributions.

4000 N Applied Load

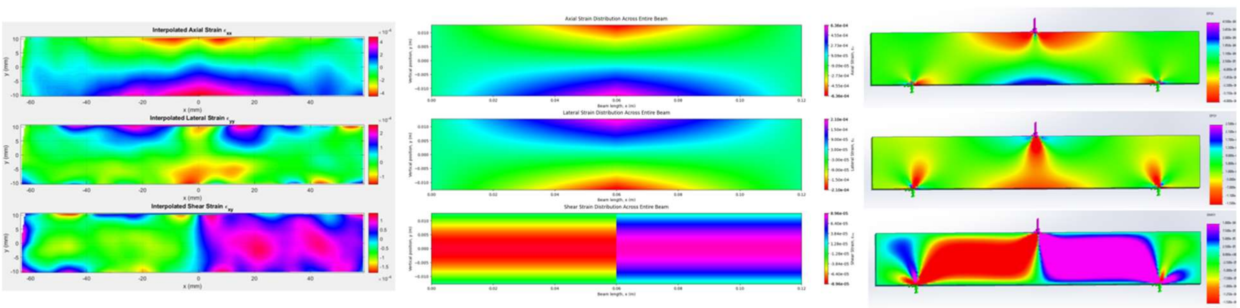


Figure 33: Axial (top row), Lateral (middle row) and Shear (bottom row) Strain Distribution for the Large Solid Beam under 4000 N Load generated from experimental values (left column), MechMat theory (middle column) and the FE model (right column).

Figure 33 compares axial, lateral, and shear strain distributions for the large solid beam generated from experimental data collected, MechMat theory, and the FE model subjected to a 4000 N applied load. At a higher load, the large solid beam strain fields show finer resolution, although noise in the data is still evident.

Axial strain consistently showed peak tensile and compressive regions at the bottom and top surfaces, respectively, across all models. Lateral and shear strains remained minimal in comparison. Noise was notably present in axial and lateral strain fields, particularly at lower load levels, affecting strain resolution.

For the small and large solid beams, strain distributions followed expected trends, with the FE model presenting smoother gradients, while DIC data captured real-world imperfections and slight asymmetries within the supported region. At 1000N, lower strain magnitudes resulted in poorer resolution, particularly for the large beam. However, at 4000N, strain values increased, reducing noise and refining strain field resolution across all beam types. The shear strain distributions consistently peaked at loading points and supports, confirming theoretical predictions.

Overall, the MechMat plots assisted in providing predictions for the experimental strain plots, although lacking localized effects. Moreover, the FE model served as a validation tool for the experimental results. Higher loads improved resolution in the DIC data, while FE simulations provided broader strain visualization across full beam lengths. The analysis confirms strong agreement between theoretical, numerical, and experimental methods, validating strain behavior predictions under applied loading conditions.

Strain Curves at Specified X-Coordinate

This section shows the axial and shear strains at x = 3m for 1000N and 4000N loads (Figure 34).

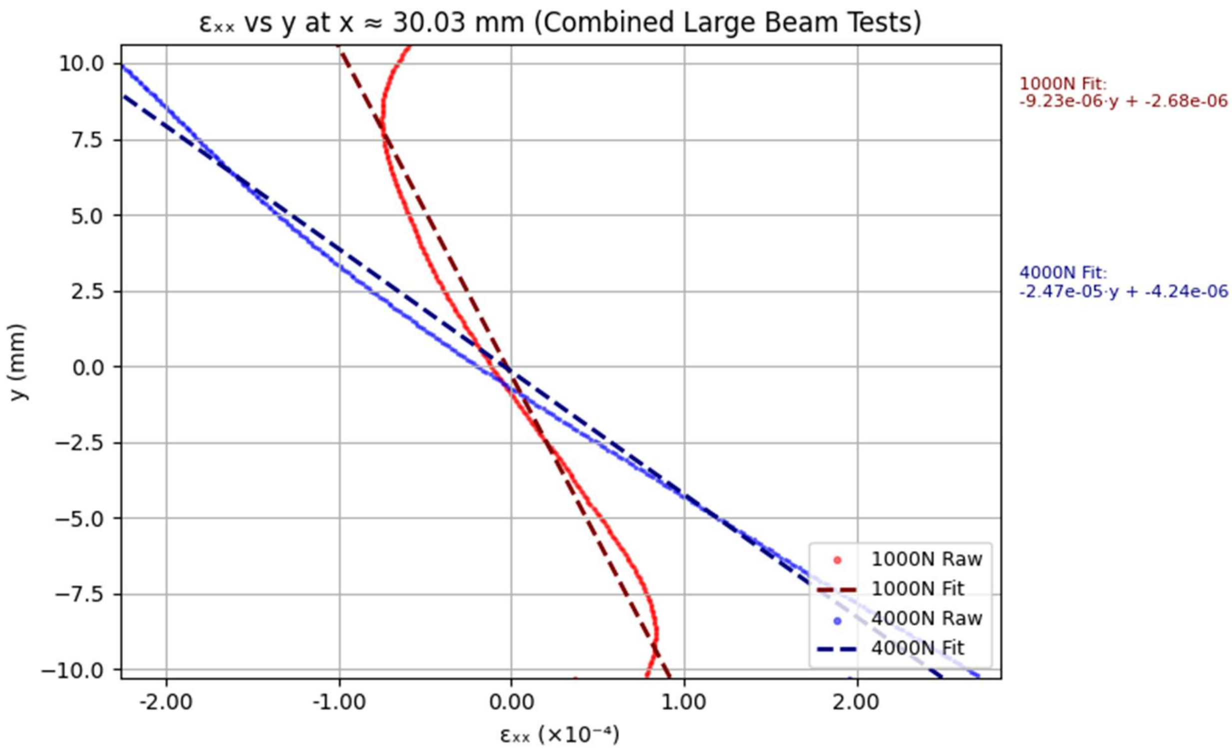


Figure 34: Axial strain vs y plotted at y = 30.03 mm from the load application.

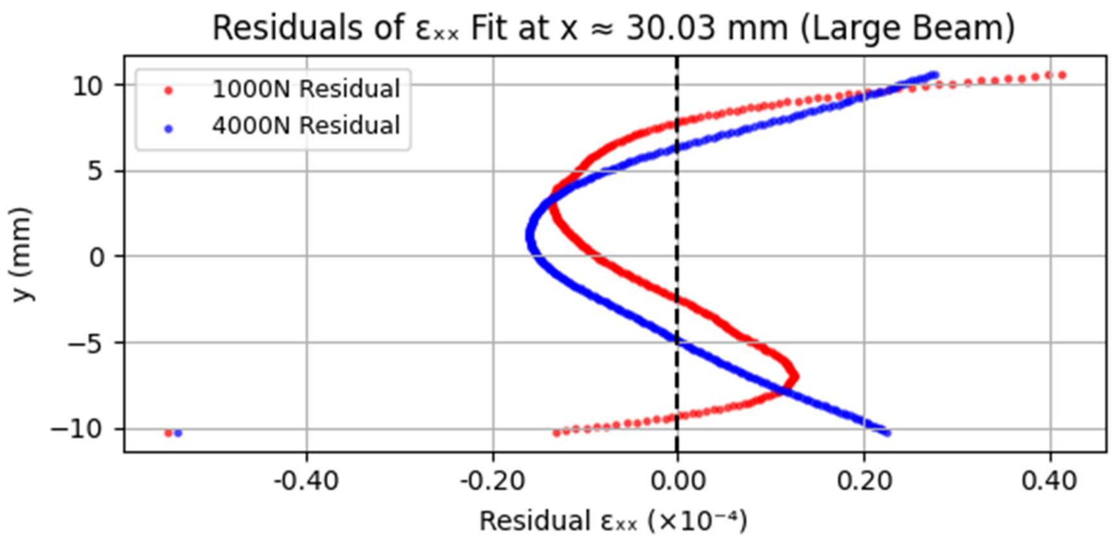


Figure 35: Residual of axial strain vs y and linear model fit.

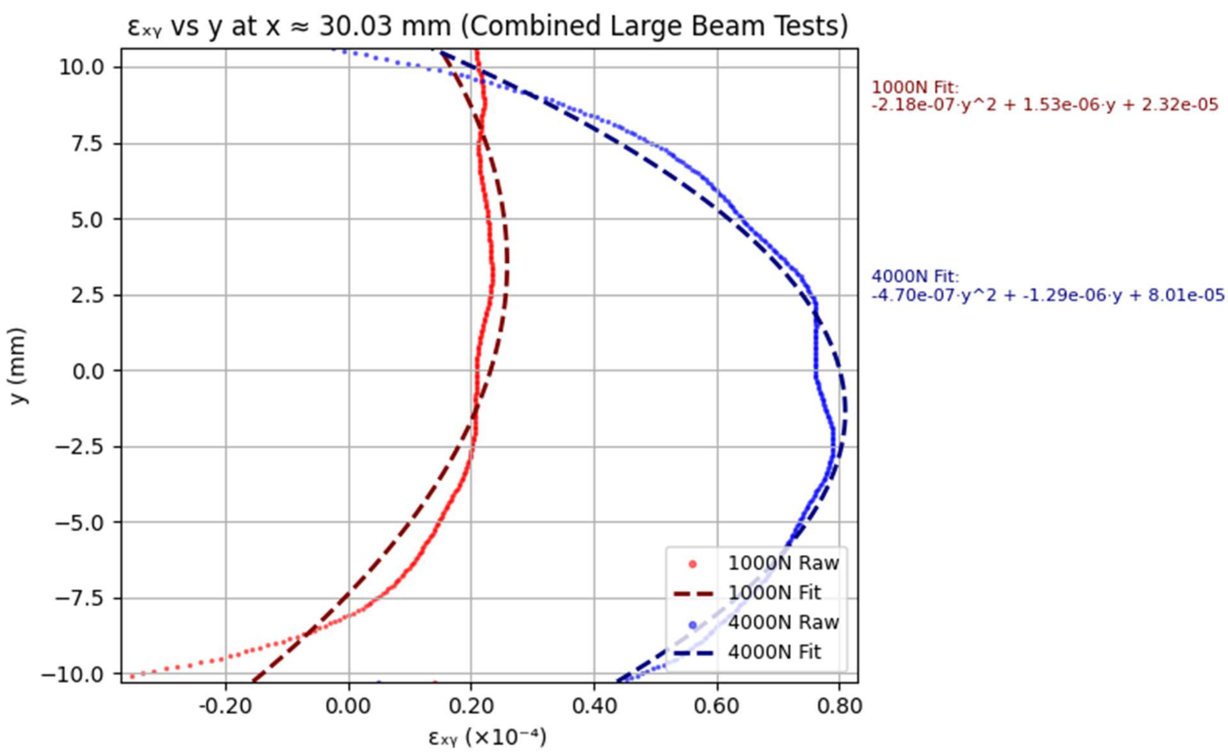


Figure 36: Shear strain vs y plotted at y = 30.03 mm from the load application.

4. Discussion

The discrepancies between experimental strain measurements and SolidWorks FEA predictions suggest that various modeling assumptions influence the results. While theoretical mechanics provides a foundation for strain estimation, real-world behavior introduces complexities that challenge simplified simulations.

One significant source of error arises from the material variability of aluminum samples. Unlike the standardized material properties assigned in SolidWorks, real aluminum specimens may exhibit manufacturing inconsistencies which impact strain response.

Another critical limitation involves the loading conditions in the 3-point bending experiment compared to the FEA model. In SolidWorks, loads were applied as simplified line forces, whereas in the physical setup, the loading rig and supports have finite contact areas with a curved radius, creating localized stress distributions that differ from theoretical expectations. The distributed nature of force application leads to strain variations that the FE model does not fully capture. Furthermore, support constraints introduce additional uncertainties, as the physical mounts may introduce contact deformation resulting in an experimental strain pattern that deviates from the rigidly constrained FE model.

Mesh resolution in the FEA simulation also plays a significant role in strain prediction accuracy. If the mesh is too coarse, strain concentrations may not be properly resolved, particularly near regions of high stress gradients. A finer mesh with localized refinement in critical deformation zones could yield results that better approximate experimental values. Finally, experimental uncertainties in Digital Image Correlation (DIC) measurements may further contribute to discrepancies. The accuracy of DIC depends on camera resolution, speckle pattern quality, and environmental factors such as lighting. Small distortions or misalignment in strain mapping could lead to measurement inaccuracies, affecting direct comparisons with FE results.

The quality of the strain contours depends on the magnitudes of strain and thus depend on the beam shape. The strain contours were compared to the mechanics of materials solution. Overall, the contour plots for the small beam at 4000N contours matched the theory the best, while the large beam at 1000N was the worst. This is evident by comparing the theoretical contours and the DIC contours in Figure 28 and 32. In Figure 28 the contour has smooth transitions high and low strain, while in Figure 32 the contour is spotty and does not present with the smooth transition seen in the theory. The noise seen in the DIC system for the larger beam under a lower compressive force can be attributed to the geometry and loads which all affect strain; the small beam has the smallest moment of inertia while the large beam has the highest moment of inertia. Therefore, at the lowest load the large beam has the lowest deflection values, while the small beam has the highest deflection at a higher load. The DIC has trouble when trying to track smaller displacements thus adding noise to the system.

These findings were validated by the strain vs y plots, taken halfway between the support and point of loading (x = 30mm), in Figures 29 and 34 which show the linear elastic relationship that was seen in the theory. In Figure 29 the scatter plot and the line of best fit, in blue, for the 4000N case is nearly linear, which was also validated by a residual plot in Figure 30. The residual plot is suggestive of higher order relationship for strain vs y; however, the residuals are small. The larger residuals towards the edges of the beam indicate localized strain effects, meaning that the beam stretched more on the edges than in the middle, which physically aligns with intuition, because the edges are furthest from the center of the beam and are therefore less insulated and more susceptible to localized effects. Figure 34 shows the strain vs y curve for the large beam at a 4000N and 1000N load, the 1000N data the linear fit does not represent the data as well as the small beam at 4000N; The residuals are about three times larger at the edges, and deviate much more in the center of the beam as evident in Figure 35. The slopes of the lines of best fit were also compared between the small beam at 4000N and the large beam at 1000N; the slope of the large beam was $m = -9.23x10^{-6}$, while the magnitude for the small beam was $m = -1.45x10^{-4}$. The magnitude of the small beam slope is 15 times larger than the large beam slope indicating that the small beam had a larger variation in strains. The smallest strain at x = 30mm for the large beam was $\epsilon_{xx\_large} \approx 1 * 10^{-4}$, while the largest strain value for the small beam was $\epsilon_{xx\_small} \approx 8 * 10^{-4}$ further validating that the small beam had larger strains and therefore increased the resolution that the DIC camera and thus the contours could capture. The hollow beam results were between the large and small beam, as the moment of inertia value was between the large and small beam. Although the hollow beam had good data that represented the linear fit well the magnitudes of the DIC data and theory had the largest discrepancy, the theoretical strain was 6 times larger than the DIC strain, $\epsilon_{xx\_small_{raw}} \approx 8 * 10^{-4}$ and $\epsilon_{xx\_smal\ theor} \approx 1.6 * 10^{-4}$. Overall, the small beam at the largest load matched the theory the best because it had the largest strains, while the large beam did not strain enough, and the theory did not account for localized effects of the hollow beam.

The shear strain resembled theory as well, with larger shear strains along the neutral axis for the cross sections at 30mm from the point of loading. As previously mentioned, beams with larger strains such as the small beam and hollow beam outperformed beams with smaller strains. It is interesting to note that the hollow beam had the largest shear strain, the maximum magnitude of shear strain for the large beam was $\epsilon_{xy\_hollo} \approx 4 * 10^{-4}$ as seen in Figure 26. This was more than twice the shear strain of the small beam, which is most likely because there is less material resisting shearing. The shear strain in the hollow beam might also be prone to more local effects because there is less material resisting shear. The large beam performed the worst as it had the smallest shear values, $\epsilon_{xy\_large} \approx 0.8 * 10^{-4}$. Even though the small beam had shear strains values that were half of the hollow the plot in Figure 31, they still resembled beam bending theory. The largest magnitudes of the shear strains for the small and large beam are almost identical to the theory with strains around $\epsilon_{xy\_large} \approx 8 * 10^{-5}$ and $\epsilon_{xy\_small} \approx 2 * 10^{-4}$ at the 4000N load. The hollow beam strain was half of the theoretical, $\epsilon_{xy_{hollow\_raw}} \approx 4 *$

10^{-4} and $\epsilon_{xy_{hollo\ \_theory}} \approx 1.6 * 10^{-4}$. Again, since the theoretical solution only inputs a moment of inertia to calculate the shear strain, it does not account for local effects due to thin walls. The theory effectively calculates shear strain for a solid cross-sectional beam with a lower moment of inertia. Overall, the small beam produced the best results when compared to the theory.

5. Conclusions

This experiment demonstrated a clear, reproducible approach to studying linear elastic behavior through a combination of Digital Image Correlation (DIC), analytical Mechanics of Materials solutions, and finite element modeling. The results confirmed that strain patterns captured via DIC closely matched theoretical predictions when strain magnitudes were sufficiently large. Beams with lower moments of inertia and higher applied loads, such as the small solid beam at 4000 N, produced strain fields with higher resolution and stronger agreement with theory. However, it is important to note that the hollow beam does not follow the mechanics of materials theory as well as the hollow beams due to local strain effects. Conversely, low-strain cases, particularly those involving higher moments of inertia beams at lower loads, introduced noise and reduced fidelity in DIC measurements.

The findings emphasize the importance of selecting appropriate beam geometries and load conditions when using DIC to assess elastic deformation. They also highlight the limitations of idealized FE models when simulating contact-based experimental setups, particularly when support and load contact conditions introduce unmodeled strain concentrations. Ultimately, this integrated methodology provides a robust foundation for students to compare real-world data with classical models, enabling deeper insight into structural behavior and promoting best practices in experimental mechanics.

In terms of future work, there are refinements that can be made specifically with the Finite Element Analysis. Moreover, as opposed to applying line loads, surface loads concentrated on areas representing the bending rig and the supports can be applied to improve the generated strain fields. In addition, a positioning rig for the VIC-EDU camera system can be machined to ensure the camera and beam sample are directly aligned. This experimental method can also be utilized to determine the Young's Modulus of a material with unknown modulus by finding the breaking point of a specific sample.

Overall, this experiment provides a rigorous and reproducible approach to studying linear elastic behavior, reinforcing the value of Digital Image Correlation (DIC) in validating theoretical predictions. The results underscore the importance of selecting appropriate beam geometries and loading conditions to achieve meaningful strain measurements while highlighting the limitations of idealized Finite Element and Mechanics of Materials models in replicating contact-based experimental setups. Future refinements, including improved boundary condition modeling and enhanced camera alignment strategies, will further strengthen the reliability of this methodology. By integrating analytical, experimental, and computational techniques, this study not only deepens our understanding of material behavior but also serves as a valuable framework for advancing best practices in experimental mechanics.

6. Acknowledgments

The authors would like to express their gratitude to Dr. David Wootton and Dr. Kamau Wright for directing the ME-360: Engineering Experimentation course, and for their roles throughout this project. Dr. Wootton served as the client, providing clear expectations and framing the engineering goals, while Dr. Wright acted as our mentor, offering essential guidance in experimental design, data analysis, documentation, interpretation of results, and organization. Their combined leadership ensured that we each derived benefit from the course.

We also wish to thank Mike Giglia for his technical support. Mike was instrumental in helping us run the experiment by assisting with the Instron setup, ensuring correct use of the DIC system, and supervising critical steps during data collection. His hands-on assistance helped maintain consistency across test runs and contributed significantly to the reliability of our results.

Additional thanks are extended to Max Summer for providing some of the aluminum test samples used in our three-point bending experiments, and to Sinisa Janjusevic for his help in preparing the beam specimens. His support in cutting, cleaning, and ensuring consistent dimensions was critical to the repeatability and accuracy of our testing.

Finally, we acknowledge The Cooper Union Mechanical Engineering Department for the use of laboratory facilities and equipment, including the Instron testing machine and DIC imaging systems. Their support enabled us to conduct this study with high-quality tools and develop a comprehensive experimental framework.

7. References

[1] Correlated Solutions, Inc., VIC-EDU System Overview, 2022. [Online]. Available: VIC-EDU_2022.pdf.

[2] Correlated Solutions, Inc., VIC-EDU User Manual, Version 6, 2024. [Online]. Available: VIC-EDU_manual_2024_digital.pdf.

[3] A. J. Wheeler and A. R. Ganji, Introduction to Engineering Experimentation, 3rd ed. Upper Saddle River, NJ: Pearson Higher Education, 2010.

[4] McMaster-Carr, “Abrasion-Resistant Polyurethane Rubber Square Bar, 6" Long, 1-1/2" Thick,” [Online]. Available: https://www.mcmaster.com/product/87235K72.

[5] McMaster-Carr, “Multipurpose 6061 Aluminum Bar, 1" Thick x 1" Wide, 24" Long,” [Online]. Available: https://www.mcmaster.com/8975K12.

[6] McMaster-Carr, “Multipurpose 6061 Aluminum Bar, 5/8" Thick x 5/8" Wide, 24" Long,” [Online]. Available: https://www.mcmaster.com/9008K11-9008K114/.

[7] P. Berkovitz, M. Hoque, and A. Saissi, *Tracking Strain in Stress Concentration using Digital Image Correlation: Experimental System Design Report for ME360*, Cooper Union, May 6, 2024.

Appendix I: Problem Formulation Diagram

Updated: 5/1/2025

Ad/Slogan: Spec it! Bend it! Track it! Re-use it!

PROJECT TITLE: Digital Image Correlation (DIC)

Need Statement:
Determine a reusable material that can undergo bending or torsion without experiencing plastic deformation for a class demonstration or laboratory assignment. Utilize DIC to track the localized strain across the material.

CLIENT: Dr. David Wootton

PROBLEM FORMULATION

Broad Goal:
Determine a material that has a large linear-elastic deformation region and track the strain under bending or torsion with the DIC.

Design Objectives/Functional Requirements:

1. Learn DIC
2. Spec material and geometry
3. Determine speckle pattern and application method
4. Torsion vs. 3-pt Bending
5. Demonstrate linearity *if possible*
6. Compare to SolidWorks
7. Test aluminum

TEAM MEMBERS:
Gabriel Kret
Maria Alvarado
Jonas Margono

(Mentor: Dr. Kamau Wright)

LOGO:

Design Constraints:

1. Size of material
2. Noise in DIC/zoom/resolution
3. Budget ~\$100 (~\$50/yr for future)
4. Difficult to get speckles on aluminum

Appendix II: Bill of Materials

Digital Image Correlation
Bill of Materials

Team Members: Maria Alvarado, Gabriel Kret, Jonas Margono
School: The Cooper Union
Team #: A2
Project: Digital Image Correlation (DIC)
Last Updated: 5/1/2025
Slogan: pec it! Bend it! Track it! Re-use

Item	Description	Material	Source	Link	Quantity	Measuremen	Unit Price	Image	Total Price
Major System Names Here	*Describe the Part*	*What is it made from*	*Where did you buy it (Home Depot, AndyMark, Supply House, Etc.)*	*Provide a purchase link (if applicable)*	*How Many? (NOT by foot)*	*Piece, Inch, Etc.*	*Cost Per Unit ($)*		
Polyurathane Beam	Abrasion-Resistant Polyurethane Rubber Square Bar Amber, 6" Long, 1-1/2" Thick, 60A Durometer, Part No: 8792K27	Polyurethane	McMaster-Carr	Abrasion-Resistant Polyurethane Rubber Square Bar, Amber, 6" Long, 1-1/2" Thick | McMaster-Carr	1	6" Long, 1-1/2" Thick	$ 37.93		$37.93
Primer	Rust-Oleum Painter's Touch 2X 12 oz. Flat White Primer General Purpose Spray Paint	N/A	Home Depot	https://www.homedepot.com/p/Rust-Oleum-Painter-s-Touch-2X-12-oz-Flat-White-Primer-General-Purpose-Spray-Paint-334019/307244964	1	12 Oz	$ 5.98		$5.98
Fine Point Sharpie	SHARPIE 37161PP Permanent Markers, Ultra Fine Point, Black, 2 Count	N/A	Amazon	Amazon.com : SHARPIE 37161PP Permanent Markers, Ultra Fine Point, Black, 2 Count : Office Products	1	N/A	$ 1.97		$1.97
Aluminum Beam 1" sq (Alternate)	Multipurpose 6061 Aluminum Bar 1" Thick x 1" Wide, Part No: 9008K14	6061	McMaster-Carr	Multipurpose 6061 Aluminum Bar, 1" Thick x 1" Wide | McMaster-Carr	1		$ 22.46		$22.46
Aluminum Beam 5/8" sq (Alternate)	Multipurpose 6061 Aluminum Bar 5/8" Thick x 5/8" Wide	6061	McMaster-Carr	https://www.mcmaster.com/9008K11-9008K114/	1	5/8" Thick x 5/8" Wide, 24" Long	$ 11.59		$11.59
Hollow Aluminum Beam 3/4" Sq	Multipurpose 6061 Aluminum Rectangular Tube, 1/16" Wall Thickness, 3/4" High x 3/4" Wide Outside	6061	McMaster-Carr	https://www.mcmaster.com/6546K52-6546K144/	1	3/4" x 3/4" Outside, 1/16" Wall Thickness	$ 23.21		$23.21
Instron Universal Testing Machine (Model 34TM-50)	The 3400 Series dual column table models provide up to 50 kN capacity available in standard and extra height options - used for testing different beam samples.		Instron		1				
Correleted Solution VIC-EDU Digital Image Correlation System	The VIC Educational System (VIC-EDU) is a low-cost solution developed for academic institutions to assist in teaching the digital image correlation technique to students - used for measuring strain distribution across different beam samples.		Correlated Solutions		1				
							Subtotals:		$103.14

Appendix III: Sample Data

A small section of data extracted from the VIC-EDU software is shown below for the small solid beam under 1000N load. Note, the full size of the data was not included due to the size of the file.

"X"	"Y"	"Z"	"U"	"V"	"W"	"exx"	"eyy"	"exy"	"e1"	"e2"	"gamma"	"sigma"	"x"	"y"	"u"	"v"	"q"	"r"	"q_ref"	"r_ref"
12.9336	23.2437	528.295	0	0	0	0	0	0	0	0	0	-1	202	1085	0	0	0	0	0	0
-61.1741	5.87684	-0.01687	0.052262	-0.05134	-0.04072	6.45E-05	0.000203	-1.29E-06	0.000203	6.45E-05	-1.56145	0.00121	209	1085	0.479615	0.368585	-87.9403	-19.9033	-88.3006	-20.2951
-60.1744	5.87241	-0.02225	0.052463	-0.05952	-0.04075	1.26E-05	0.000175	-2.71E-06	0.000175	1.26E-05	-1.55406	0.001238	216	1085	0.482129	0.432119	-87.157	-19.763	-87.5171	-20.2162
-59.1715	5.86674	-0.01682	0.052235	-0.0679	-0.04179	-1.96E-05	0.000149	-6.55E-06	0.000149	-1.98E-05	-1.53208	0.00127	223	1085	0.484332	0.495836	-86.3492	-19.5958	-86.7046	-20.1127
-58.1708	5.86205	-0.01477	0.051756	-0.07621	-0.04232	-3.95E-05	0.000129	-1.28E-05	0.00013	-4.04E-05	-1.49565	0.001307	230	1085	0.483383	0.558396	-85.5693	-19.445	-85.9186	-20.0258
-57.1728	5.85857	-0.01723	0.051572	-0.08438	-0.04139	-5.18E-05	0.000111	-1.99E-05	0.000113	-5.42E-05	-1.45092	0.001328	237	1085	0.480629	0.619596	-84.8162	-19.3152	-85.1632	-19.96
-56.1763	5.85569	-0.02171	0.051413	-0.0926	-0.04013	-6.17E-05	9.24E-05	-2.83E-05	9.74E-05	-6.68E-05	-1.39473	0.00129	244	1085	0.477234	0.681068	-84.0837	-19.1949	-84.429	-19.9044
-55.1761	5.84984	-0.01321	0.051483	-0.10093	-0.03836	-7.27E-05	7.54E-05	-3.85E-05	8.48E-05	-8.21E-05	-1.3312	0.001242	251	1085	0.474168	0.742142	-83.3117	-19.0207	-83.6577	-19.7971
-54.1822	5.84851	-0.01952	0.051477	-0.10919	-0.03711	-8.77E-05	6.37E-05	-4.99E-05	7.86E-05	-0.0001	-1.27919	0.001172	258	1085	0.471935	0.803375	-82.6112	-18.9241	-82.9566	-19.766
-53.1837	5.84167	-0.01058	0.051213	-0.11739	-0.03676	-0.00011	5.57E-05	-6.19E-05	7.59E-05	-0.00013	-1.25496	0.001175	265	1085	0.470215	0.864544	-81.8683	-18.738	-82.2104	-19.6442
-52.1871	5.833	-0.0041	0.050914	-0.12536	-0.03635	-0.00014	5.23E-05	-7.37E-05	7.78E-05	-0.00016	-1.23783	0.001158	272	1085	0.468077	0.924033	-81.1491	-18.5278	-81.4876	-19.4964
-51.1937	5.82347	-0.00377	0.05062	-0.13332	-0.03661	-0.00015	5.31E-05	-8.59E-05	8.43E-05	-0.00018	-1.22251	0.001149	279	1085	0.467659	0.983738	-80.4682	-18.3092	-80.8023	-19.3395
-50.2005	5.81517	-0.00162	0.050452	-0.14151	-0.03679	-0.00017	5.94E-05	-9.71E-05	9.46E-05	-0.00021	-1.22299	0.001142	286	1085	0.46797	1.04475	-79.7918	-18.1058	-80.1227	-19.2
-49.2092	5.80842	-0.00225	0.050277	-0.14981	-0.0364	-0.0002	6.95E-05	-0.00011	0.000106	-0.00024	-1.24007	0.001116	293	1085	0.466785	1.10763	-79.1382	-17.9274	-79.4665	-19.0855
-48.2226	5.80555	-0.01342	0.050033	-0.15804	-0.0359	-0.00024	8.33E-05	-0.00011	0.000117	-0.00028	-1.27381	0.001129	300	1085	0.464813	1.1711	-78.5382	-17.8146	-78.8634	-19.0354
-47.2344	5.79976	-0.01761	0.049696	-0.16607	-0.03505	-0.00029	0.000103	-0.00011	0.000131	-0.00032	-1.31475	0.001149	307	1085	0.461313	1.23162	-77.925	-17.6543	-78.2469	-18.9376
-46.2449	5.79173	-0.01576	0.049056	-0.17396	-0.03485	-0.00034	0.000125	-0.00011	0.000148	-0.00036	-1.35239	0.001206	314	1085	0.457249	1.29007	-77.3048	-17.4576	-77.6203	-18.803
-45.2569	5.78487	-0.01539	0.048519	-0.18165	-0.03435	-0.00038	0.000147	-0.0001	0.000167	-0.0004	-1.38282	0.001201	321	1085	0.453177	1.34884	-76.7017	-17.2824	-77.012	-18.6868
-44.2673	5.77797	-0.00758	0.048136	-0.18935	-0.03397	-0.00041	0.000166	-9.95E-05	0.000182	-0.00043	-1.40571	0.0012	328	1085	0.450491	1.40749	-76.0815	-17.1024	-76.3876	-18.5658
-43.2776	5.77163	0.0028	0.047895	-0.19703	-0.03319	-0.00045	0.000182	-9.60E-05	0.000196	-0.00046	-1.42274	0.001175	335	1085	0.447812	1.46564	-75.4623	-16.9289	-75.7659	-18.452
-42.2916	5.76767	0.005312	0.047642	-0.20473	-0.03235	-0.00048	0.000199	-9.49E-05	0.000212	-0.00049	-1.43398	0.001154	342	1085	0.44491	1.52351	-74.886	-16.7953	-75.1871	-18.3787
-41.3072	5.76108	0.005803	0.047014	-0.21232	-0.03256	-0.0005	0.000219	-9.61E-05	0.000231	-0.00051	-1.44048	0.00113	349	1085	0.44193	1.58094	-74.3353	-16.6255	-74.6297	-18.2671
-40.3246	5.75424	0.003932	0.046158	-0.21966	-0.0336	-0.00052	0.000238	-9.84E-05	0.000251	-0.00054	-1.44428	0.001141	356	1085	0.439286	1.63848	-73.8094	-16.4575	-74.0944	-18.1533
-39.3448	5.74844	-0.00392	0.045246	-0.22688	-0.03333	-0.00054	0.000256	-0.0001	0.000269	-0.00055	-1.44503	0.00116	363	1085	0.432997	1.69507	-73.318	-16.3092	-73.5947	-18.0589
-38.3644	5.74264	-0.00722	0.044403	-0.23416	-0.03336	-0.00056	0.000272	-0.00011	0.000285	-0.00057	-1.44416	0.001158	370	1085	0.427941	1.75083	-72.8214	-16.1567	-73.09	-17.9618
-37.3847	5.73589	-0.00993	0.043843	-0.24149	-0.03342	-0.00058	0.000287	-0.00011	0.000301	-0.00059	-1.44142	0.001142	377	1085	0.424979	1.80598	-72.3332	-15.9886	-72.5959	-17.8504
-36.4048	5.72815	-0.0096	0.043404	-0.24881	-0.03336	-0.0006	0.000301	-0.00012	0.000318	-0.00062	-1.4368	0.001125	384	1085	0.42256	1.86078	-71.8451	-15.8035	-72.1029	-17.7223
-35.4256	5.71879	-0.00825	0.042996	-0.25602	-0.03383	-0.00063	0.000314	-0.00013	0.000332	-0.00064	-1.43299	0.001175	391	1085	0.421624	1.91512	-71.3665	-15.595	-71.6191	-17.5694
-34.4469	5.71156	-0.00615	0.042604	-0.26312	-0.03409	-0.00065	0.000324	-0.00014	0.000343	-0.00067	-1.43192	0.00117	398	1085	0.420277	1.96818	-70.8966	-15.4187	-71.1442	-17.4483
-33.4705	5.70569	-0.0081	0.042092	-0.27013	-0.03336	-0.00068	0.000331	-0.00014	0.000351	-0.0007	-1.43323	0.001135	405	1085	0.415693	2.02087	-70.4541	-15.2657	-70.6972	-17.3501
-32.4922	5.6997	-0.0017	0.041242	-0.27696	-0.03371	-0.00071	0.000336	-0.00014	0.000355	-0.00073	-1.43655	0.001063	412	1085	0.411222	2.07358	-69.9957	-15.1097	-70.2305	-17.2456
-31.5146	5.69418	0.00536	0.040217	-0.2836	-0.03383	-0.00073	0.000339	-0.00014	0.000358	-0.00075	-1.4421	0.001044	419	1085	0.404907	2.12522	-69.5482	-14.962	-69.7735	-17.1478
-30.5397	5.68779	0.006943	0.03906	-0.28995	-0.03356	-0.00076	0.000341	-0.00014	0.000357	-0.00077	-1.44916	0.001019	426	1085	0.39668	2.17547	-69.135	-14.8075	-69.3505	-17.0402
-29.5656	5.68233	0.008861	0.038009	-0.29635	-0.0327	-0.00078	0.00034	-0.00013	0.000355	-0.0008	-1.45861	0.001064	433	1085	0.38785	2.22584	-68.7311	-14.6658	-68.9382	-16.9466
-28.5929	5.67773	0.00885	0.037213	-0.30275	-0.03151	-0.0008	0.000338	-0.00012	0.00035	-0.00082	-1.47142	0.001122	440	1085	0.380061	2.2766	-68.3442	-14.5388	-68.5456	-16.8674
-27.6215	5.67241	0.007957	0.036532	-0.30889	-0.03053	-0.00082	0.000338	-9.77E-05	0.000347	-0.00083	-1.48757	0.001195	447	1085	0.373586	2.32476	-67.9727	-14.4029	-68.1689	-16.7778
-26.6516	5.66523	0.004943	0.035711	-0.31469	-0.03036	-0.00084	0.000338	-7.90E-05	0.000343	-0.00085	-1.50422	0.001235	454	1085	0.367921	2.36987	-67.6243	-14.2433	-67.8133	-16.6617
-25.6816	5.65753	0.004837	0.034969	-0.32036	-0.0312	-0.00086	0.000339	-6.38E-05	0.000342	-0.00086	-1.5177	0.001215	461	1085	0.365119	2.41339	-67.2781	-14.0747	-67.4589	-16.5359
-24.7132	5.64953	0.002662	0.034224	-0.326	-0.03161	-0.00087	0.000341	-4.95E-05	0.000343	-0.00087	-1.53008	0.001166	468	1085	0.361277	2.45658	-66.9515	-13.9031	-67.1248	-16.4071
-23.7442	5.6433	0.004948	0.033603	-0.3318	-0.03131	-0.00089	0.000346	-3.45E-05	0.000347	-0.00089	-1.54287	0.001167	475	1085	0.356709	2.50079	-66.6176	-13.7539	-66.7853	-16.3025
-22.7763	5.63739	0.006417	0.032754	-0.33751	-0.03115	-0.00091	0.000353	-1.98E-05	0.000354	-0.00091	-1.55512	0.001191	482	1085	0.350802	2.54448	-66.3007	-13.6111	-66.461	-16.2034
-21.8085	5.63187	0.01045	0.031809	-0.34274	-0.03106	-0.00094	0.000363	-8.00E-06	0.000363	-0.00094	-1.56466	0.001226	489	1085	0.344304	2.58443	-65.9869	-13.4765	-66.1389	-16.1089
-20.8426	5.62592	0.011439	0.03064	-0.34758	-0.03124	-0.00098	0.000376	-2.92E-06	0.000376	-0.00098	-1.56864	0.001246	496	1085	0.3367	2.62214	-65.6991	-13.341	-65.8408	-16.0098
-19.8778	5.61979	0.011338	0.029451	-0.3522	-0.03103	-0.00102	0.000391	-3.53E-06	0.000391	-0.00102	-1.56829	0.001273	503	1085	0.32804	2.65771	-65.4271	-13.205	-65.559	-15.9088
-18.9153	5.6134	0.006833	0.02828	-0.35692	-0.03115	-0.00105	0.000409	-6.85E-06	0.000409	-0.00105	-1.5661	0.001276	510	1085	0.320236	2.69483	-65.1845	-13.0675	-65.3063	-15.8065
-17.9522	5.60705	0.006713	0.027279	-0.36163	-0.03097	-0.00108	0.000427	-1.17E-05	0.000427	-0.00108	-1.56307	0.001287	517	1085	0.312989	2.73231	-64.9353	-12.9288	-65.0487	-15.7024
-16.9886	5.6008	0.011382	0.026356	-0.36615	-0.03048	-0.00111	0.000445	-1.74E-05	0.000445	-0.00111	-1.55962	0.001227	524	1085	0.305624	2.76879	-64.6788	-12.7904	-64.7849	-15.5971
-16.0265	5.59461	0.013968	0.025319	-0.37041	-0.03012	-0.00115	0.000466	-2.38E-05	0.000466	-0.00115	-1.55601	0.001151	531	1085	0.297658	2.8036	-64.4432	-12.6568	-64.541	-15.4942
-15.0636	5.5885	0.02189	0.024146	-0.37451	-0.02988	-0.00117	0.000487	-3.25E-05	0.000488	-0.00117	-1.55123	0.001057	538	1085	0.288931	2.83722	-64.2004	-12.5227	-64.2887	-15.3894
-14.1033	5.58213	0.023986	0.022682	-0.37842	-0.0301	-0.0012	0.000507	-4.60E-05	0.000508	-0.0012	-1.54391	0.001025	545	1085	0.279034	2.86814	-63.9947	-12.3886	-64.0707	-15.2843
-13.1449	5.57582	0.022722	0.021323	-0.38202	-0.03032	-0.00123	0.000504	-7.09E-05	0.000507	-0.00123	-1.52993	0.001014	552	1085	0.26985	2.89573	-63.8129	-12.2592	-63.8773	-15.1822
-12.1881	5.56947	0.019009	0.020182	-0.38536	-0.02973	-0.00125	0.000442	-0.00012	0.000451	-0.00125	-1.5016	0.001013	559	1085	0.26045	2.92151	-63.6493	-12.1324	-63.7052	-15.081
-11.2323	5.56338	0.014397	0.019106	-0.38848	-0.02885	-0.00126	0.000272	-0.00018	0.000293	-0.00128	-1.45645	0.001014	566	1085	0.250897	2.94683	-63.4998	-12.0129	-63.5482	-14.9845
-10.2768	5.55707	0.01179	0.017959	-0.39154	-0.02813	-0.00128	-3.20E-05	-0.00025	1.69E-05	-0.00133	-1.37884	0.001014	573	1085	0.241154	2.9722	-63.3553	-11.8901	-63.3954	-14.8837
-9.32261	5.55033	0.007655	0.016619	-0.39445	-0.02807	-0.0013	-0.00052	-0.00032	-0.00041	-0.00141	-1.2266	0.001002	580	1085	0.231374	2.9966	-63.231	-11.7634	-63.2604	-14.7771
-8.36899	5.5435	0.004294	0.015278	-0.39723	-0.02811	-0.00132	-0.00116	-0.00038	-0.00085	-0.00163	-0.8916	0.000962	587	1085	0.221748	3.01914	-63.1156	-11.6354	-63.1341	-14.6691
-7.41603	5.53733	0.001493	0.013835	-0.39986	-0.02837	-0.00134	-0.00195	-0.00045	-0.00111	-0.00219	-0.48508	0.001251	594	1085	0.211799	3.04125	-63.0107	-11.519	-63.0173	-14.5709
-6.46306	5.531	0.001702	0.012139	-0.403	-0.02751	-0.00135	-0.0029	-0.00052	-0.0012	-0.00306	-0.29494	0.002266	601	1085	0.197727	3.0735	-62.9061	-11.4002	-62.9007	-14.4686

Appendix IV: Python Code and Matlab

Mechanics of Materials Simulation Model

The Mechanics of Materials python solution referenced throughout the report can be found here as a .py file or below as text:

https://github.com/gkret123/Digital_Img_Correlation/tree/main/Data_Pipeline

----

Author: Gabriel Kret

Instructors: Dr. David Wootton and Dr. Kamau Wright

Course: ME-360-A: Engineering Experimentation

Date: Spring 2025

This script analyzes a simply supported beam subjected to a central load using Mechanics of Materials beam theory.

It computes the bending moment, shear force, deflection, and strain/stress distributions across the beam's cross-section.

It also visualizes the results using plots and heatmaps.

This script is intended to be used alongside the Digital Image Correlation (DIC) analysis of a 3-point bending test.

The results from this script can be compared with the DIC results to validate the mechanical behavior of the beam.

```
"""

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import os

"""
F_applied and X_coord are the user inputs for the loading condition at a specific x coordinate along the beam.
enter a value for F_applied to specify the loading condition you wish to analyze.
The value of F_applied is the force applied at the center of the beam.
The script will use this value to calculate the bending moment, shear forces, deflections, etc.
NOTE: It will also be used to approximate whether the case puts the beam in the elastic or plastic region.

enter a value for X_coord as the position along the beam where you want to analyze the bending moment, shear force, and deflection.
The value of X_coord is the distance from the left end of the beam to the point where you want to analyze the bending moment, shear force, and deflection.
"""
F_applied = 4000  # Applied force in Newtons (example value), This should generally be the maximum force you apply to the beam.

#For additional forces, enter a LIST of forces to be plotted along with the applied force. If you do not want to plot any additional forces, leave this as None or an empty list.
additional_force_to_plot = [1000] #additional forces (generally less than the "applied force" that will be plotted along with the applied force to show the curves at different loading conditions.
```

x_coord = 0.09 # Position along the beam in meters (example value, 6.25 cm) starting from the LEFT support.

Beam geometry and material properties

L = 0.12 # Beam length in meters (12 cm) (measured)

width = 0.01905 # meters (measured)

height = width # meters (measured)

Note: The inner dimensions are only used if the beam is hollow. write 'None' if the beam is solid.

inner_width = 0.0127 #measured

inner_height = inner_width # meters

#known/estimated material properties

poisson = 0.33 # Poisson's ratio (material property)

E = 69e9 # Young's modulus in Pascals (example value for Aluminum 6061) (Material property)

yield_strength = 276e6 # Yield strength in Pascals (example value for Aluminum 6061) (Material property)

G = E / (2 * (1 + poisson)) # Compute the shear modulus from Young's modulus and Poisson's ratio

```
def compute_inertia(width, height, inner_width = None, inner_height = None):
    """
    Compute the second moment of area (I) for a rectangular or hollow rectangular cross-section.
    """
    if inner_width is not None and inner_height is not None:
```

```
        # Hollow section
        I_outer = (width * height**3) / 12.0
        I_inner = (inner_width * inner_height**3) / 12.0
        I = I_outer - I_inner
    else:
        # Solid section
        I = (width * height**3) / 12.0
    return I


def bending_moment(x, L, F_applied):
    """
    Compute the bending moment at a distance x along a simply supported beam
    with a central load F_applied.
    For x <= L/2: M = F_applied*x/2; for x > L/2: M = F_applied*(L-x)/2.
    """
    if x <= L / 2:
        return F_applied * x / 2.0
    else:
        return F_applied * (L - x) / 2.0

def bending_moment_array(x_array, L, F_applied):
    """ Vectorized bending moment over an array of x coordinates """
    return np.array([bending_moment(x, L, F_applied) for x in x_array])

def shear_force(x, L, F_applied):
    """
```

```
    Compute the shear force at a distance x along the beam.
    For a simply supported beam with a central load:
      if x < L/2, shear V = F_applied/2; if x > L/2, V = -F_applied/2.
    (A jump exists at the midspan.)
    """
    if x < L / 2:
        return F_applied / 2.0
    else:
        return -F_applied / 2.0


def shear_force_array(x_array, L, F_applied):
    """  shear force over an array of x coordinates """
    return np.array([shear_force(x, L, F_applied) for x in x_array])


def deflection(x, L, F_applied, E, I):
    """
    Compute the vertical deflection at position x along the beam.
    For a simply supported beam with a central load, the deflection is given by:
      For 0 <= x <= L/2:
         v(x) = (F_applied*x*(3*L**2 - 4*x**2))/(48*E*I)
      For L/2 <= x <= L:
         v(x) = (F_applied*(L-x)*(3*L**2 - 4*(L-x)**2))/(48*E*I)
    """
    if x <= L / 2:
        return (F_applied * x * (3 * L**2 - 4 * x**2)) / (48 * E * I)
    else:
        return (F_applied * (L - x) * (3 * L**2 - 4 * (L - x)**2)) / (48 * E * I)
```

```
def deflection_array(x_array, L, F_applied, E, I):
    """ Vectorized deflection calculation along the beam """
    return np.array([deflection(x, L, F_applied, E, I) for x in x_array])

def strain_distribution(y, M, E, I):
    """
    Compute the axial strain at a point y (distance from the neutral axis) due to bending.
    εx(y) = - y * M / (E * I)
    """
    return - y * M / (E * I)

def shear_stress_distribution(y, V, width, height):
    """
    Compute the shear stress distribution in a rectangular cross-section at the neutral axis.
    For a rectangular beam:
     τ(y) = (3/2)*(V/(width*height))*(1 - (2*y/height)**2)
    """
    return (3/2) * (-V / (width * height)) * (1 - (2 * y / height)**2)

def elastic_or_plastic(F_applied, I):

    #Determine if the beam is in the elastic or plastic region for the force case applied

    c = height / 2  # Distance from the neutral axis to the outer fiber
    # Calculate the maximum bending stress allowed
    M_yield = yield_strength * I / c  # Yield moment
```

```
    # Calculate the maximum load that can be applied without yielding
    P_yield = 4 * M_yield / L  # Yield load, for a simply supported beam with a central load
    # Calculate the maximum bending stress induced by the applied load
    M = bending_moment(L / 2, L, F_applied)  # Maximum moment at the center of the beam
    sigma_max = M * c / I  # Maximum bending stress induced by the applied load

    print("The critical load is: ", P_yield, "N")
    print("The applied load is: ", F_applied, "N")
    print("The maximum bending stress from the applied load is: ", sigma_max/10**6, "MPa")
    print("The yield strength is: ", yield_strength/10**6, "MPa")

    #compare with yield strength
    if sigma_max > yield_strength:
        print("\nThe beam is in the plastic region, the force applied induced a bending stress higher than the yield strength. \n")

        print(f"The maximum bending stress is: {sigma_max/10**6} MPa, which is greater than the yield strength of {yield_strength/10**6} MPa.")
        print(f"The applied load of {F_applied} N is greater than the critical load of {P_yield} N, The beam is expected to yield at an applied load of {P_yield} N.")
        return "Plastic"
    else:
        print("\nThe beam is in the elastic region, the force applied does not induce a bending stress higher than the yield strength. \n")

        print(f"The maximum bending stress is: {sigma_max/10**6} MPa, which is less than the yield strength of {yield_strength/10**6} MPa.")
        print(f"An applied load of {P_yield} N would cause the beam to cross into the plastic region.")
```

```
        return "Elastic"


#Helper function to plot strain profiles for axial, lateral, and shear strains
def plot_strain_profiles(forces, y, x_coord, L, E, I, poisson, G):
    fig, axs = plt.subplots(1, 3, figsize=(24,10), sharey=True)
    for F in forces:
        M = bending_moment(x_coord, L, F)
        V = shear_force(x_coord, L, F)
        Epsilon_x = strain_distribution(y, M, E, I)
        Epsilon_y = -poisson * Epsilon_x
        τ  = shear_stress_distribution(y, V, width, height)
        gamma  = τ / G
        Epsilon_x_y = gamma / 2  # Shear strain (engineering shear strain)
        # Plotting the strain distributions
        axs[0].plot(Epsilon_x, y, label=f'εx, F={F:.0f}N')
        axs[1].plot(Epsilon_y, y, label=f'εγ, F={F:.0f}N')
        axs[2].plot(Epsilon_x_y, y,  label=f'εxγ, F={F:.0f}N')


    axs[0].set_xlabel("Axial Strain, εx")
    axs[1].set_xlabel("Lateral Strain, εγ")
    axs[2].set_xlabel("Shear Strain, εxγ")
    axs[0].set_ylabel("y (m)")
    for ax, title in zip(axs, ["Axial", "Lateral", "Shear"]):
        ax.set_title(f"{title} Strain Distribution")
        ax.grid(True)
        ax.legend()
        ax.tick_params(axis='x', rotation=45, labelsize = 16)  # Rotate x-tick labels by 45 degrees
```

```
        ax.tick_params(axis='y', labelsize = 16)
    plt.suptitle(f"Strain Profiles at x={x_coord:.3f} m")
    plt.tight_layout(rect=[0.0125,0,1,0.95])
    plt.show()

#Helper function to plot stress profiles for axial, lateral, and shear stresses
def plot_stress_profiles(forces, y, x_coord, L, E, I, poisson):
    fig, axs = plt.subplots(1, 3, figsize=(24,10), sharey=True)
    for F in forces:
        M = bending_moment(x_coord, L, F)
        V = shear_force(x_coord, L, F)
        Epsilon_x = strain_distribution(y, M, E, I)
        Epsilon_y = -poisson * Epsilon_x
        Sigma_x = E * Epsilon_x
        Sigma_y = E * Epsilon_y
        Tau = shear_stress_distribution(y, V, width, height)

        axs[0].plot(Sigma_x, y, label=f'σₓ, F={F:.0f}N')
        axs[1].plot(Sigma_y, y, label=f'σᵧ, F={F:.0f}N')
        axs[2].plot(Tau,  y, label=f'τ,  F={F:.0f}N')

    axs[0].set_xlabel("σₓ (Pa)")
    axs[1].set_xlabel("σᵧ (Pa)")
    axs[2].set_xlabel("τ  (Pa)")
    axs[0].set_ylabel("y (m)")
    for ax, title in zip(axs, ["Normal", "Lateral", "Shear"]):
        ax.set_title(f"{title} Stress Distribution")
```

```
        ax.grid(True)
        ax.legend()
    plt.suptitle(f"Stress Profiles at x={x_coord:.3f} m")
    plt.tight_layout(rect=[0.0125,0,1,0.95])
    plt.show()


#Helper function to plot shear and moment diagrams
def plot_shear_moment(forces, L, E, I):
    x_beam = np.linspace(0, L, 200)
    fig, axes = plt.subplots(2, 1, figsize=(8,6), sharex=True)
    for F in forces:
        V = shear_force_array(x_beam, L, F)
        M = bending_moment_array(x_beam, L, F)
        axes[0].plot(x_beam, V, label=f'V(x), F={F:.0f} N')
        axes[1].plot(x_beam, M, label=f'M(x), F={F:.0f} N')
    axes[0].set_title("Shear Force Diagram");  axes[0].set_ylabel("V (N)"); axes[0].grid(True);
axes[0].legend()
    axes[1].set_title("Bending Moment Diagram"); axes[1].set_xlabel("x (m)");
axes[1].set_ylabel("M (Nm)"); axes[1].grid(True); axes[1].legend()
    plt.tight_layout()
    plt.show()


#Helper function to plot deflection curves
def plot_deflection_curves(forces, L, E, I):
    x_beam = np.linspace(0, L, 200)
    plt.figure(figsize=(8,4))
    for F in forces:
        v = deflection_array(x_beam, L, F, E, I)
```

```
        plt.plot(x_beam, v, label=f'δ(x), F={F:.0f} N')
    plt.title("Deflection Curves")
    plt.xlabel("x (m)", fontsize = 16)
    plt.ylabel("δ (m)", fontsize = 16)
    plt.tick_params(axis = 'x', rotation=45, labelsize = 16)
    plt.tick_params(axis = 'y', labelsize = 16)
    plt.grid(True)
    plt.legend()
    plt.tight_layout()
    plt.show()


def main():
    # the user inputs for the loading condition at a specific x coordinate along the beam (Force and
x position) are defined above


    if type(additional_force_to_plot) == type(None) or additional_force_to_plot == []:
        forces = [F_applied] # list of forces to plot
    else:
        forces = additional_force_to_plot + [F_applied] # list of forces to plot


    # Calculate the moment of inertia
    I = compute_inertia(width, height, inner_width, inner_height)
    # Bending moment at the chosen x coordinate
    M = bending_moment(x_coord, L, F_applied)
    # Shear force at the chosen x coordinate
    V = shear_force(x_coord, L, F_applied)
```

```
# Cross-Section Analysis

# Define vertical positions (y) across the cross-section (neutral axis at y = 0)
y = np.linspace(-height / 2, height / 2, 100)
# Compute axial strain distribution at the given x coordinate
epsilon_x = strain_distribution(y, M, E, I)
# Lateral strain (due to Poisson effect)
epsilon_y = - poisson * epsilon_x
# Bending stress distribution (stress = E * strain)
sigma = E * epsilon_x
# Shear stress distribution across the height
tau = shear_stress_distribution(y, V, width, height)



print("\nSample strain/stress values (every 10th point) at x = {:.4f} m:".format(x_coord))
for i in range(0, len(y), 10):
    print(f"y = {y[i]:.4f} m, εₓ = {epsilon_x[i]:.6e}, εᵧ = {epsilon_y[i]:.6e}, σ = {sigma[i]:.6e} Pa, τ = {tau[i]:.6e} Pa")

print(f"\n--- Beam Analysis Results for F_applied = {F_applied} ---\n")
#bending region
print("Understanding the beams elastic and plastic regions:\n")
elastic_or_plastic(F_applied, I)

print("\nBeam Geometry and Material Properties:\n")

print(f"Moment of Inertia (I): {I:.6e} m^4")
```

```
print(f"Bending Moment (M) at x = {x_coord:.4f} m: {M:.6e} Nm")
print(f"Shear Force (V) at x = {x_coord:.4f} m: {V:.6e} N")

# Maximum deflection for a simply supported beam with a central load
delta_max = F_applied * L**3 / (48 * E * I)
print(f"Maximum Deflection: {delta_max:.6e} m")

#Plots 1-3:
plot_strain_profiles(forces, y, x_coord, L, E, I, poisson, G)
#Plots 4-6:
plot_stress_profiles(forces, y, x_coord, L, E, I, poisson)

# Plots 6-9: Heatmaps at the cross‑section specified
ny_cs, nz = 50, 50
y_cs = np.linspace(-height/2, height/2, ny_cs)
z_cs = np.linspace(-width/2, width/2, nz)
Z_cs, Y_cs = np.meshgrid(z_cs, y_cs)

# Strain fields
exx_cs = strain_distribution(Y_cs, M, E, I)
eyy_cs = - poisson * exx_cs

# Shear strain
V_cs = shear_force(x_coord, L, F_applied)
tau_cs = shear_stress_distribution(Y_cs, V_cs, width, height)
gamma_cs = tau_cs / G
epsilon_x_y = gamma_cs / 2
```

```
fig, axs = plt.subplots(1, 3, figsize=(18, 5), constrained_layout=True)

# Axial Strain
cp0 = axs[0].contourf(Z_cs, Y_cs, exx_cs, 200, cmap='gist_rainbow')
fig.colorbar(cp0, ax=axs[0], label='Axial Strain, εx')
axs[0].set_xlabel("Horizontal coordinate, z (m)")
axs[0].set_ylabel("Vertical coordinate, y (m)")
axs[0].set_title(f"Axial Strain (εx)\nCross-Section at x = {x_coord:.4f} m")

# Lateral Strain
cp1 = axs[1].contourf(Z_cs, Y_cs, eyy_cs, 200, cmap='gist_rainbow')
fig.colorbar(cp1, ax=axs[1], label='Lateral Strain, εγ')
axs[1].set_xlabel("Horizontal coordinate, z (m)")
axs[1].set_ylabel("Vertical coordinate, y (m)")
axs[1].set_title(f"Lateral Strain (εγ)\nCross-Section at x = {x_coord:.4f} m")

# Shear Strain
#cp2 = axs[2].contourf(Z_cs, Y_cs, gamma_cs, 200, cmap='gist_rainbow')
cp2 = axs[2].contourf(Z_cs, Y_cs, epsilon_x_y, 200, cmap='gist_rainbow') #changed to include exy to match VIC data
fig.colorbar(cp2, ax=axs[2], label='Shear Strain, εxγ')
axs[2].set_xlabel("Horizontal coordinate, z (m)")
axs[2].set_ylabel("Vertical coordinate, y (m)")
axs[2].set_title(f"Shear Strain (εxγ)\nCross-Section at x = {x_coord:.4f} m")

plt.suptitle(f"Heatmap Display of Strain Distributions in Cross-Section\nat x = {x_coord:.4f} m", fontsize=16)
```

```
# Beam Diagram Plots (along x)
x_beam = np.linspace(0, L, 200)
M_beam = bending_moment_array(x_beam, L, F_applied)
V_beam = shear_force_array(x_beam, L, F_applied)
v_beam = deflection_array(x_beam, L, F_applied, E, I)

# Plot 10-11: Shear and bending moment diagrams
plot_shear_moment(forces, L, E, I)

# Plot 12: Deflection Curve
plot_deflection_curves(forces, L, E, I)

# Plots 13-15: stacked heatmap subplots for axial strain, lateral strain, and shear strain across entire beam.
# Create a grid covering the entire beam: x from 0 to L and y from -height/2 to height/2.
nx_entire, ny_entire = 2000, 500
x_entire = np.linspace(0, L, nx_entire)
y_entire = np.linspace(-height/2, height/2, ny_entire)
X_entire, Y_entire = np.meshgrid(x_entire, y_entire)
# Compute bending moment at each x location (note: M is independent of y)
M_entire = np.array([bending_moment(x, L, F_applied) for x in x_entire])
# Broadcast M_entire along y-direction
M_grid = np.tile(M_entire, (ny_entire, 1))
# Compute axial strain distribution over the entire beam:
epsilon_x_entire = - Y_entire * M_grid / (E * I)
# Lateral strain distribution using Poisson's ratio:
```

```
epsilon_y_entire = - poisson * epsilon_x_entire
# shear strain distribution using shear force:

V_entire = np.array([shear_force(x, L, F_applied) for x in x_entire])
V_grid = np.tile(V_entire, (ny_entire, 1))
# First compute shear stress distribution
tau_entire = shear_stress_distribution(Y_entire, V_grid, width, height)
# Convert shear stress to shear strain
gamma_entire = tau_entire / G
epsilon_x_y_entire = gamma_entire / 2  # Shear strain (engineering shear strain)

# create subplots with automatic spacing for titles/colorbars
fig, axs = plt.subplots(3, 1, figsize=(16, 10), constrained_layout=True)

# Axial strain
cp3 = axs[0].contourf(X_entire, Y_entire, epsilon_x_entire, 500, cmap='gist_rainbow')
cb0 = fig.colorbar(cp3, ax=axs[0], label='Axial Strain, εₓ')
vmin0, vmax0 = epsilon_x_entire.min(), epsilon_x_entire.max()
ticks0 = np.linspace(vmin0, vmax0, 8)
cb0.set_ticks([vmin0, *ticks0, vmax0])
cb0.ax.yaxis.set_major_formatter(ticker.FormatStrFormatter('%.2e'))
axs[0].set_xlabel("Beam length, x (m)")
axs[0].set_ylabel("Vertical position, y (m)")
axs[0].set_title("Axial Strain Distribution Across Entire Beam", pad=12)

# Lateral strain
cp4 = axs[1].contourf(X_entire, Y_entire, epsilon_y_entire, 500, cmap='gist_rainbow')
```

```
    cb1 = fig.colorbar(cp4, ax=axs[1], label='Lateral Strain, εᵧ')
    vmin1, vmax1 = epsilon_y_entire.min(), epsilon_y_entire.max()
    ticks1 = np.linspace(vmin1, vmax1, 8)
    cb1.set_ticks([vmin1, *ticks1, vmax1])
    cb1.ax.yaxis.set_major_formatter(ticker.FormatStrFormatter('%.2e'))
    axs[1].set_xlabel("Beam length, x (m)")
    axs[1].set_ylabel("Vertical position, y (m)")
    axs[1].set_title("Lateral Strain Distribution Across Entire Beam", pad=12)


    # Shear strain
    #cp5 = axs[2].contourf(X_entire, Y_entire, gamma_entire, 500, cmap='gist_rainbow')
    cp5 = axs[2].contourf(X_entire, Y_entire, epsilon_x_y_entire, 500, cmap='gist_rainbow')
#changed to include exy to match VIC data
    cb2 = fig.colorbar(cp5, ax=axs[2], label='Shear Strain, εₓᵧ')
    #vmin2, vmax2 = gamma_entire.min(), gamma_entire.max()
    vmin2, vmax2 = epsilon_x_y_entire.min(), epsilon_x_y_entire.max() #changed to include exy
to match VIC data
    ticks2 = np.linspace(vmin2, vmax2, 8)
    cb2.set_ticks([vmin2, *ticks2, vmax2])
    cb2.ax.yaxis.set_major_formatter(ticker.FormatStrFormatter('%.2e'))
    axs[2].set_xlabel("Beam length, x (m)")
    axs[2].set_ylabel("Vertical position, y (m)")
    axs[2].set_title("Shear Strain Distribution Across Entire Beam", pad=12)
    plt.show()


if __name__ == "__main__":
    main()
```

Python Code for Strain Distributions

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from scipy.interpolate import griddata

# Set global plot font style
plt.rcParams.update({'font.size': 10, 'font.family': 'sans-serif'})

# Function to load strain data from a DIC file, interpolate it, and extract a vertical slice at a specific x-location
def load_strain_slice(file_label, strain_type, target_x):
    file_path = f"/Users/jonasmargono/Documents/DIC Data/DIC_{file_label}.xlsx"
    df = pd.read_excel(file_path)

    # Clean column names and remove one known erroneous point, check excel sheet to remove point with no data, there should only be one
    df.columns = df.columns.str.strip().str.replace('"', '')
    df = df[~((df['X'] == 44.0074) & (df['Y'] == 41.7513) & (df['Z'] == 483.018))]

    # Keep only nonzero, valid strain data
    df_strain = df[df[strain_type] != 0].dropna(subset=['X', 'Y', strain_type])

    # Interpolate onto a uniform grid
    xi = np.linspace(df_strain['X'].min(), df_strain['X'].max(), 300)
    yi = np.linspace(df_strain['Y'].min(), df_strain['Y'].max(), 300)
```

```
    Xg, Yg = np.meshgrid(xi, yi)
    Z = griddata((df_strain['X'], df_strain['Y']), df_strain[strain_type], (Xg, Yg), method='linear')

    # Reshape to long-form DataFrame
    df_long = pd.DataFrame(Z, index=yi, columns=xi).stack().reset_index()
    df_long.columns = ['Y', 'X', strain_type]
    df_long = df_long.dropna()

    # Get the data at the X-location closest to the desired cross-section
    nearest_x = df_long['X'].iloc[(np.abs(df_long['X'] - target_x)).argmin()]
    return df_long[df_long['X'] == nearest_x], nearest_x

# Function to plot both the strain-vs-y fit and the residuals at a particular x-location
def plot_strain_comparison(label, strain_type, poly_order, target_x, ylabel, residual_color):
    # Load interpolated slices for 1000N and 4000N tests
    slice_1000N, x_1000N = load_strain_slice(f"{label}_1000N", strain_type, target_x)
    slice_4000N, x_4000N = load_strain_slice(f"{label}_4000N", strain_type, target_x)

    # Fit polynomial of specified order to strain data
    coeffs_1000N = np.polyfit(slice_1000N['Y'], slice_1000N[strain_type], poly_order)
    coeffs_4000N = np.polyfit(slice_4000N['Y'], slice_4000N[strain_type], poly_order)
    fit_1000N = np.polyval(coeffs_1000N, slice_1000N['Y'])
    fit_4000N = np.polyval(coeffs_4000N, slice_4000N['Y'])

    # Calculate residuals = actual - fitted strain
    residuals_1000N = slice_1000N[strain_type] - fit_1000N
    residuals_4000N = slice_4000N[strain_type] - fit_4000N
```

```
    # Format polynomial equation as a readable string
    def poly_to_string(coeffs):
        terms = [f"{coeff:.2e}·y^{poly_order - i}" if poly_order - i > 1
                 else f"{coeff:.2e}·y" if poly_order - i == 1
                 else f"{coeff:.2e}" for i, coeff in enumerate(coeffs)]
        return " + ".join(terms)

    eq_1000N = poly_to_string(coeffs_1000N)
    eq_4000N = poly_to_string(coeffs_4000N)

    # ---- STRAIN FIT PLOT ----
    plt.figure(figsize=(8, 5))
    # Plot raw data and fitted curves for both loads
    plt.scatter(slice_1000N[strain_type], slice_1000N['Y'], s=2, alpha=0.5, color='red',
label='1000N Raw')
    plt.plot(fit_1000N, slice_1000N['Y'], linestyle='--', color='maroon', linewidth=2, label='1000N
Fit')
    plt.scatter(slice_4000N[strain_type], slice_4000N['Y'], s=2, alpha=0.5, color='blue',
label='4000N Raw')
    plt.plot(fit_4000N, slice_4000N['Y'], linestyle='--', color='navy', linewidth=2, label='4000N
Fit')

    # Display equations of fitted polynomials
    plt.text(1.02, 0.95, f"1000N Fit:\n{eq_1000N}",
             transform=plt.gca().transAxes, fontsize=8, color='maroon',
             verticalalignment='top', bbox=dict(facecolor='white', alpha=0.7, edgecolor='none'))
    plt.text(1.02, 0.65, f"4000N Fit:\n{eq_4000N}",
```

```
        transform=plt.gca().transAxes, fontsize=8, color='navy',
        verticalalignment='top', bbox=dict(facecolor='white', alpha=0.7, edgecolor='none'))

    # Label axes and format
    plt.xlabel(f"{ylabel} (×10⁻⁴)")
    plt.ylabel("y (mm)")
    plt.title(f"{ylabel} vs y at x ≈ {x_1000N:.2f} mm (Combined {title} Tests)")
    plt.grid(True)
    plt.legend(markerscale=2, loc='lower right', fontsize=9)
    plt.gca().xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: f"{x*1e4:.2f}"))
    plt.tight_layout()
    plt.xlim(left=min(slice_1000N[strain_type].min(), slice_4000N[strain_type].min()) * 0.95,
        right=max(slice_1000N[strain_type].max(), slice_4000N[strain_type].max()) * 1.05)
    plt.ylim(slice_1000N['Y'].min(), slice_1000N['Y'].max())
    plt.show()

    # ---- RESIDUALS PLOT ----
    plt.figure(figsize=(6, 3))
    plt.scatter(residuals_1000N, slice_1000N['Y'], s=5, alpha=0.6, color=residual_color,
label='1000N Residual')
    plt.scatter(residuals_4000N, slice_4000N['Y'], s=5, alpha=0.6, color='blue', label='4000N
Residual')
    plt.axvline(0, color='black', linestyle='--')  # Reference line at zero residual
    plt.xlabel(f"Residual {ylabel} (×10⁻⁴)")
    plt.ylabel("y (mm)")
    plt.title(f"Residuals of {ylabel} Fit at x ≈ {x_1000N:.2f} mm ({title})")
    plt.grid(True)
    plt.legend(fontsize=9)
```

```
    plt.gca().xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: f"{x*1e4:.2f}"))
    plt.tight_layout()
    plt.show()


# Set beam and test info
#Make sure to label file DIC_label_1000N.xlsx
title = "Hollow Beam"
label = "HollowBox"


# Plot εxx vs y using a linear (1st order) polynomial
plot_strain_comparison(label, strain_type="exx", poly_order=1, target_x=30, ylabel="εxx", residual_color='red')


# Uncomment the next line to analyze shear strain instead, using a 2nd order fit
# plot_strain_comparison(label, strain_type="exy", poly_order=2, target_x=30, ylabel="εxγ", residual_color='red')
```

MATLAB Code to Generate Heatmaps for VIC Data

```
% ME360 Engineering Experimentation
% Team A2 - Digital Image Correlation


clear all;
% Load the Excel file, note: must change exported csv files to xlsx file
% NOTE: when plotting data for different beams, also change x_val, y_val, z_val
% Hollow Box Aluminum
% filepath for 1000N:
file_path = "C:\Users\maria\Downloads\DIC_DataProcessing\HollowBox\DIC_HollowBox_1000N.xlsx";
% filepath for 4000N:
```

```
%file_path =
"C:\Users\maria\Downloads\DIC_DataProcessing\HollowBox\DIC_HollowBox_4000N.xlsx";


% Small Solid Aluminum
% file path for 1000N:
%file_path =
"C:\Users\maria\Downloads\DIC_DataProcessing\SmallBox\DIC_SmallBox_1000N.xlsx";

% file path for 4000N:
%file_path =
"C:\Users\maria\Downloads\DIC_DataProcessing\SmallBox\DIC_SmallBox_4000N.xlsx";


% Large Solid Aluminum
% file path for 1000N:
%file_path =
"C:\Users\maria\Downloads\DIC_DataProcessing\LargeBox\DIC_LargeBox_1000N.xlsx";

% file path for 4000N:
%file_path =
"C:\Users\maria\Downloads\DIC_DataProcessing\LargeBox\DIC_LargeBox_4000N.xlsx";

opts = detectImportOptions(file_path);
opts.VariableNamingRule = 'preserve';
opts.Sheet = 1; % Adjust if needed
data = readtable(file_path, opts);

% Ensure column names are valid MATLAB identifiers
```

```
data.Properties.VariableNames = matlab.lang.makeValidName(data.Properties.VariableNames);

% % Hollow Box - coordinates to remove, note: these values change for each beam
x_val = 44.0074;
y_val = 41.7513;
z_val = 483.018;

% % Small Solid - coordinates to remove:
% x_val = 12.9336;
% y_val = 23.2437;
% z_val = 528.295;

% % Large Solid - coordinates to remove
% x_val = 35.4785;
% y_val = 17.7537;
% z_val = 483.516;

% Check if required columns exist to avoid errors
if all(ismember({'X', 'Y', 'Z'}, data.Properties.VariableNames))
   % Filter out exact match row
   rowsToRemove = (data.X == x_val) & (data.Y == y_val) & (data.Z == z_val);
   data(rowsToRemove, :) = [];

   % Display number of remaining rows
   disp(['Remaining rows: ', num2str(height(data))]);
else
   disp("Error: Required columns are missing from the dataset.");
```

```
end

% Scatter plot X vs Y
figure;
scatter(data.X, data.Y, 1, 'filled'); % Marker size set to 1
xlabel('X');
ylabel('Y');
title('Scatter Plot of X vs Y');
grid on;

% Display number of rows in the dataset
disp(['Number of rows: ', num2str(height(data))]);

% Heat Map of exx Strain Field
figure;
scatter(data.X, data.Y, 10, data.exx, 'filled'); % Marker size set to 10
colormap('parula'); % MATLAB does not have 'inferno', using 'parula' as an alternative
colorbar;
xlabel('X');
ylabel('Y');
title('Heat Map of Axial Strain Field');
grid on;

% Heat Map of eyy Strain Field
figure;
scatter(data.X, data.Y, 10, data.eyy, 'filled'); % Marker size set to 10
colormap('hot'); % MATLAB does not natively support 'plasma', using 'hot' as an alternative
```

```
colorbar;
xlabel('X');
ylabel('Y');
title('Heat Map of Lateral Strain Field');
grid on;

% Heat Map of exy Strain Field
figure;
scatter(data.X, data.Y, 10, data.exy, 'filled'); % Marker size set to 10
colormap('jet'); % MATLAB does not natively support 'viridis', using 'jet' as an alternative
colorbar;
xlabel('X');
ylabel('Y');
title('Heat Map of Shear Strain Field');
grid on;

% Clean column names
data.Properties.VariableNames = matlab.lang.makeValidName(data.Properties.VariableNames);

% Interpolated Shear Strain (ε_xy)
figure;
%subplot(3,1,3);
data_filtered = data(~isnan(data.exy) & data.exy ~= 0, :);
xi = linspace(min(data_filtered.X), max(data_filtered.X), 300);
yi = linspace(min(data_filtered.Y), max(data_filtered.Y), 300);
[X_grid, Y_grid] = meshgrid(xi, yi);
Z = griddata(data_filtered.X, data_filtered.Y, data_filtered.exy, X_grid, Y_grid, 'linear');
```

```
contourf(X_grid, Y_grid, Z, 100, 'LineStyle', 'none');
colormap('hsv');
colorbar;
xlabel('x (mm)');
ylabel('y (mm)');
title('Interpolated Shear Strain \epsilon_{xy}');
axis equal;
grid on;

% Interpolated Lateral Strain (ε_yy)
%subplot(3,1,2);
figure;
data_filtered = data(~isnan(data.eyy) & data.eyy ~= 0, :);
xi = linspace(min(data_filtered.X), max(data_filtered.X), 300);
yi = linspace(min(data_filtered.Y), max(data_filtered.Y), 300);
[X_grid, Y_grid] = meshgrid(xi, yi);
Z = griddata(data_filtered.X, data_filtered.Y, data_filtered.eyy, X_grid, Y_grid, 'linear');
contourf(X_grid, Y_grid, Z, 100, 'LineStyle', 'none');
colormap('hsv');
colorbar;
xlabel('x (mm)');
ylabel('y (mm)');
title('Interpolated Lateral Strain \epsilon_{yy}');
axis equal;
grid on;

% Interpolated Axial Strain (ε_xx)
```

```
%subplot(3,1,1);
figure;
data_filtered = data(~isnan(data.exx) & data.exx ~= 0, :);
xi = linspace(min(data_filtered.X), max(data_filtered.X), 300);
yi = linspace(min(data_filtered.Y), max(data_filtered.Y), 300);
[X_grid, Y_grid] = meshgrid(xi, yi);
Z = griddata(data_filtered.X, data_filtered.Y, data_filtered.exx, X_grid, Y_grid, 'linear');
contourf(X_grid, Y_grid, Z, 100, 'LineStyle', 'none');
colormap('hsv');
colorbar;
xlabel('x (mm)');
ylabel('y (mm)');
title('Interpolated Axial Strain \epsilon_{xx}');
axis equal;
grid on;
```

Appendix V: SolidWorks Simulations

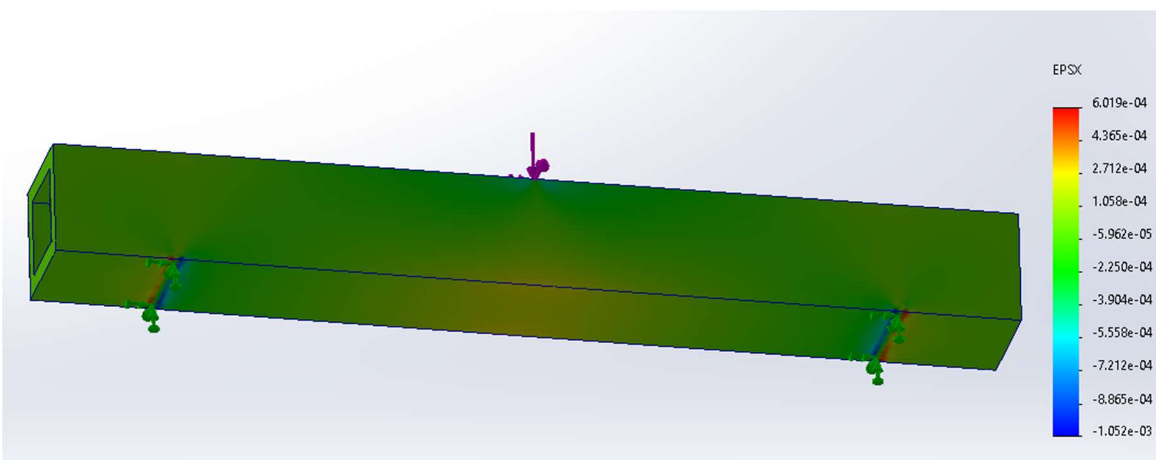
General Guidelines for Developing a Finite Element Model of Three-Point Bending

1. After sketching a square, extrude the square to be 16 cm long.
2. Select the top surface of the beam for a new sketch.
3. Sketch a line directly at half of the beam's length, 8 cm.
4. Select the bottom surface of the beam to create a new sketch.
5. Sketch two new lines on the bottom surface, each 6 cm from the previous line made at half of the beam's length.
6. Set Aluminum 6061 as the material.
7. Create a new study:
    - Select force and create a new split for the top-half line.
        1. After creating split, create new reference axis on the top by selecting reference geometry.
        2. Use this axis to apply a line force in -y direction.
    - Select fixed geometry and create a new split for the two-bottom lines.
        1. After creating split, create new reference axes on the two bottom lines by selecting reference geometry.
        2. Use these axes to create fixed geometry at the two new axes.
8. Create a new mesh set to the finest resolution.
9. Run the study.

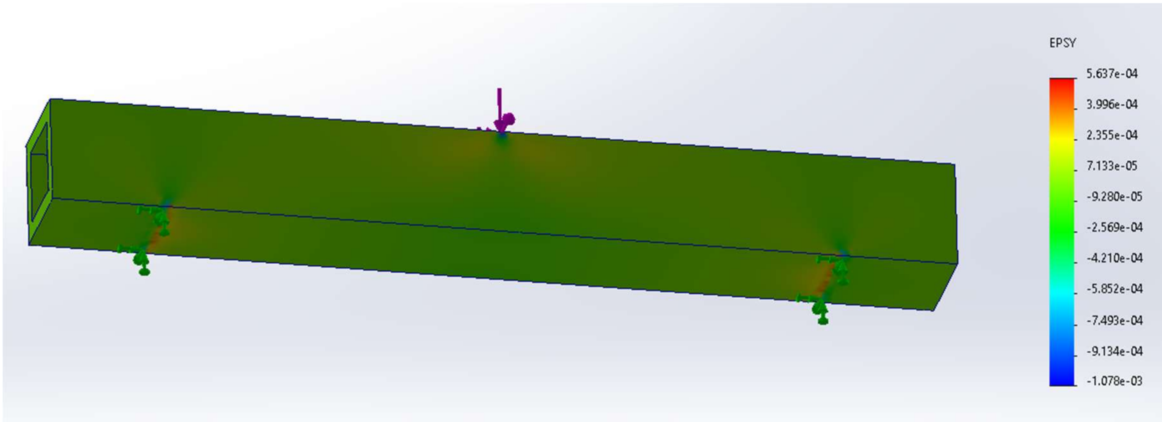
Hollow Box Beam Simulations

1000 N

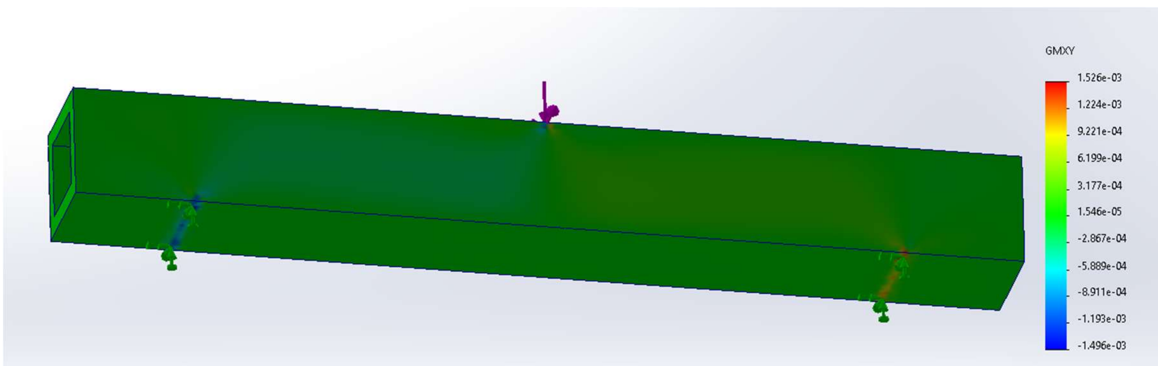
X-Normal Strain



Y-Normal Strain

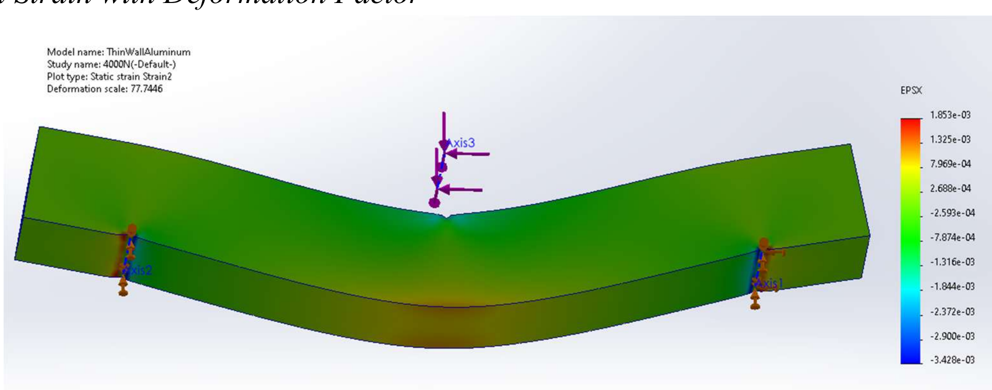


Shear Strain

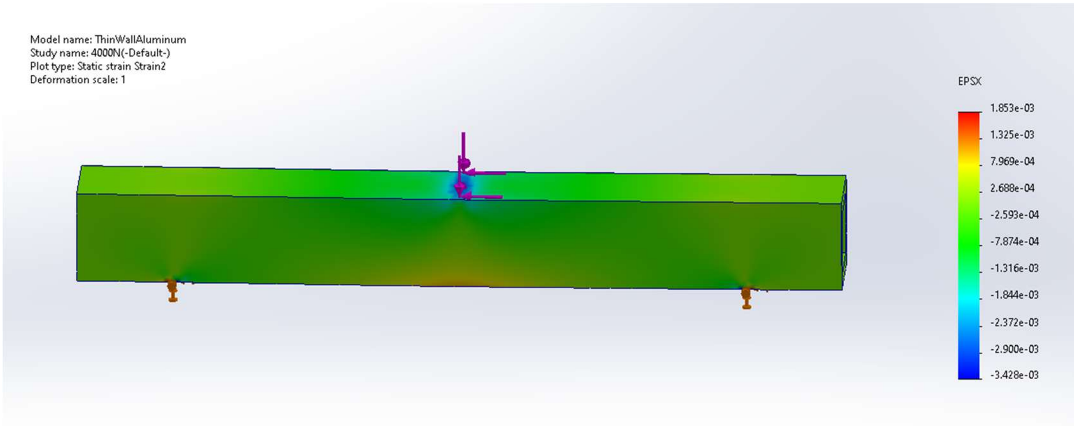


4000 N

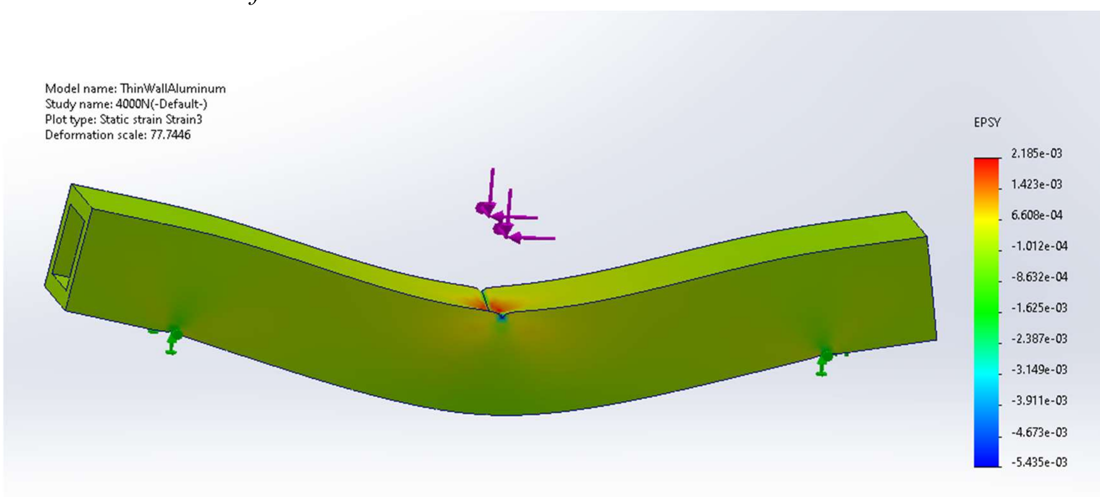
X-Normal Strain with Deformation Factor



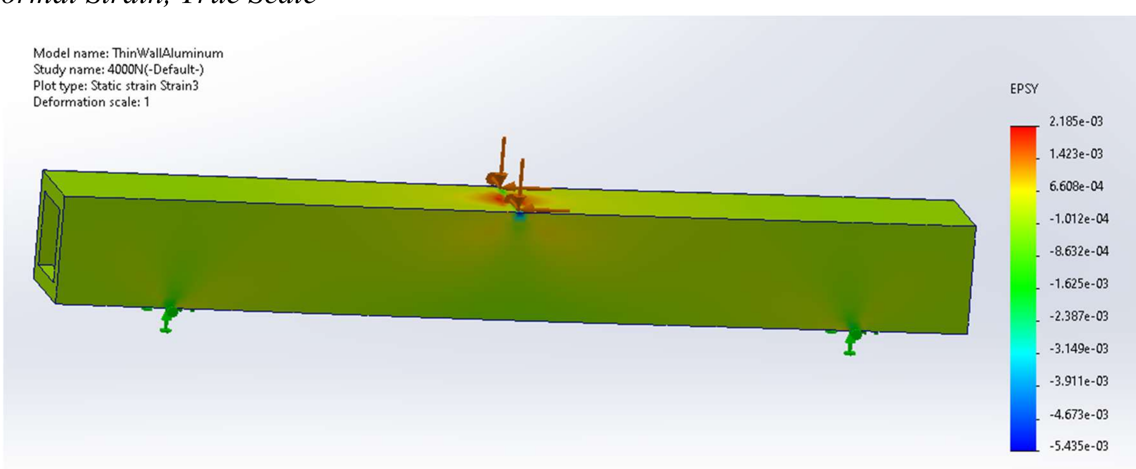
X-Normal Strain, True Scale



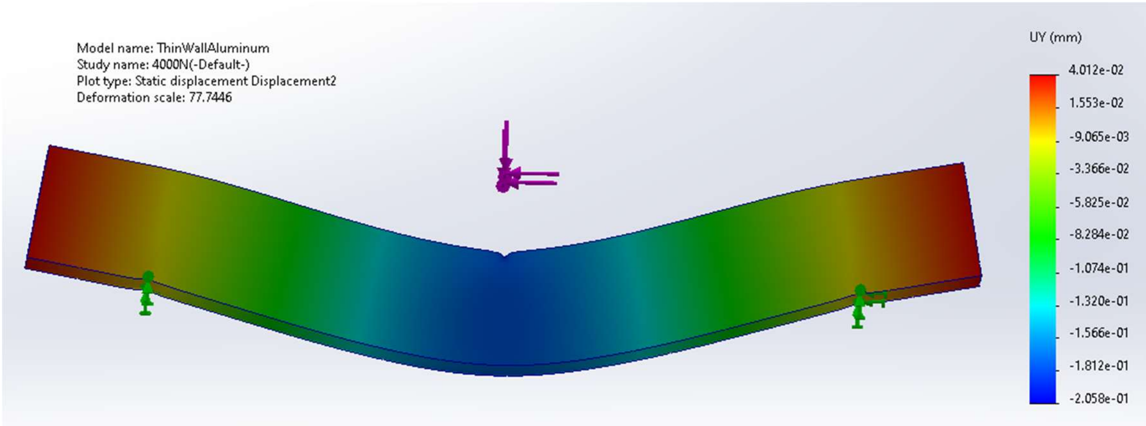
Y-Normal Strain with Deformation Factor



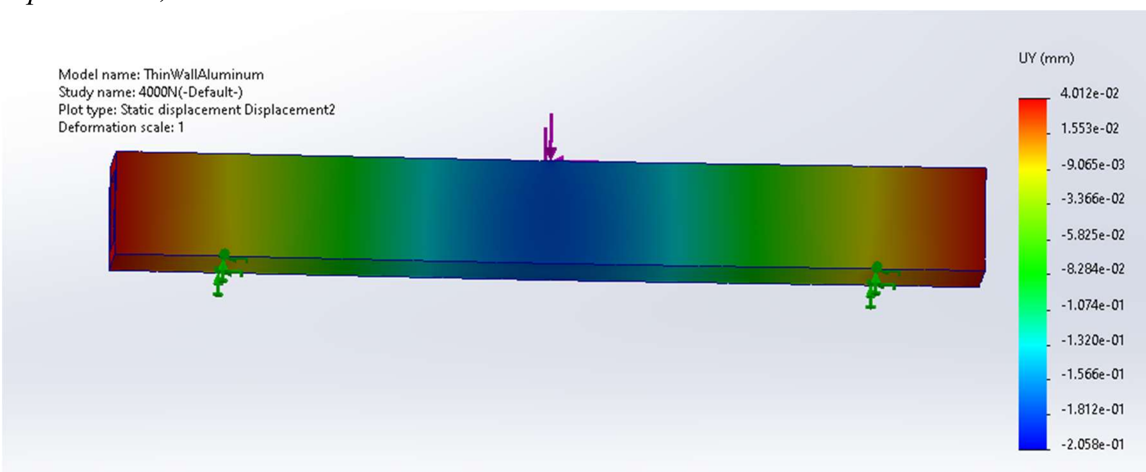
Y-Normal Strain, True Scale



Y-Displacement with Deformation Factor



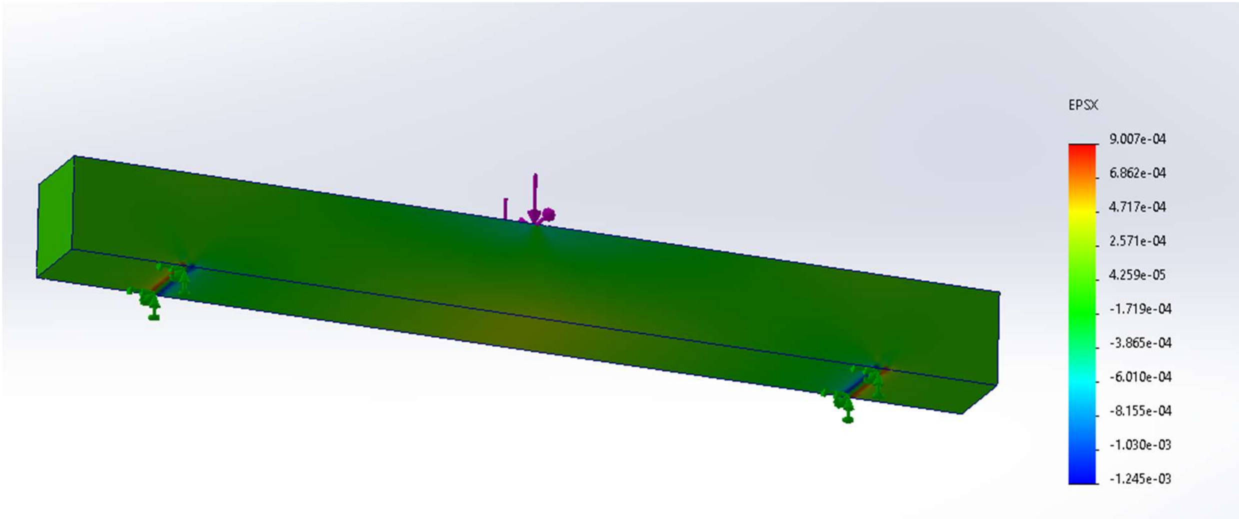
Y-Displacement, True Scale



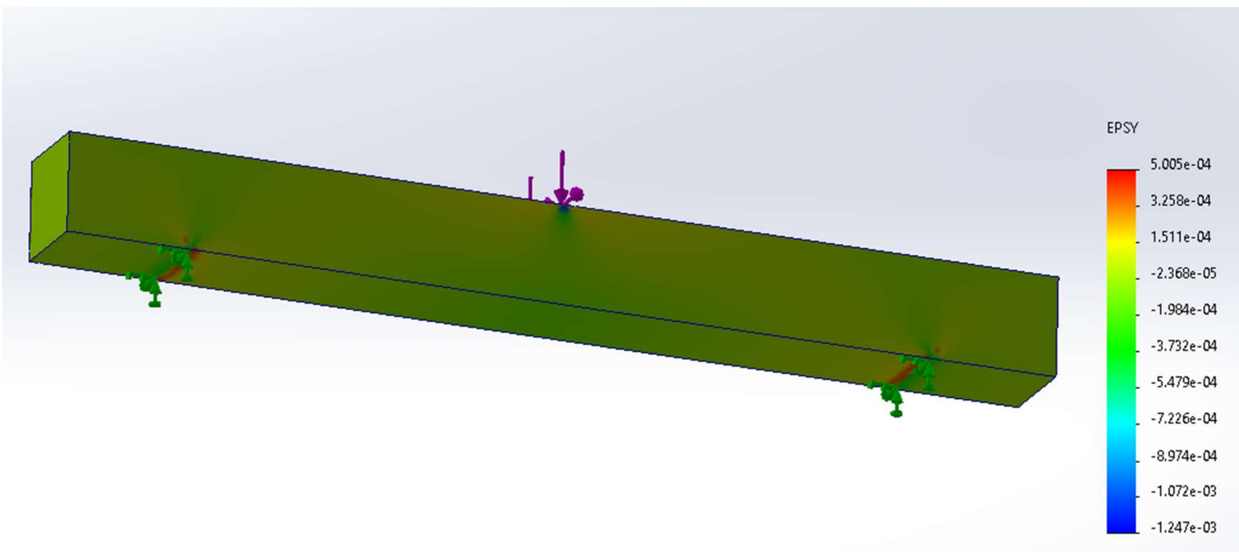
Small Solid Beam Simulations

1000 N

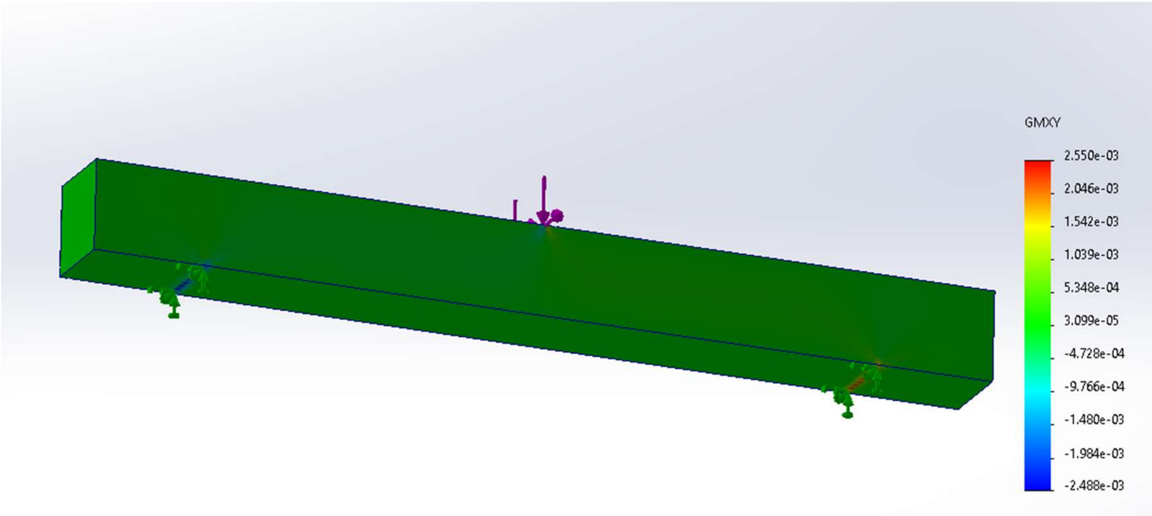
X-Normal Strain



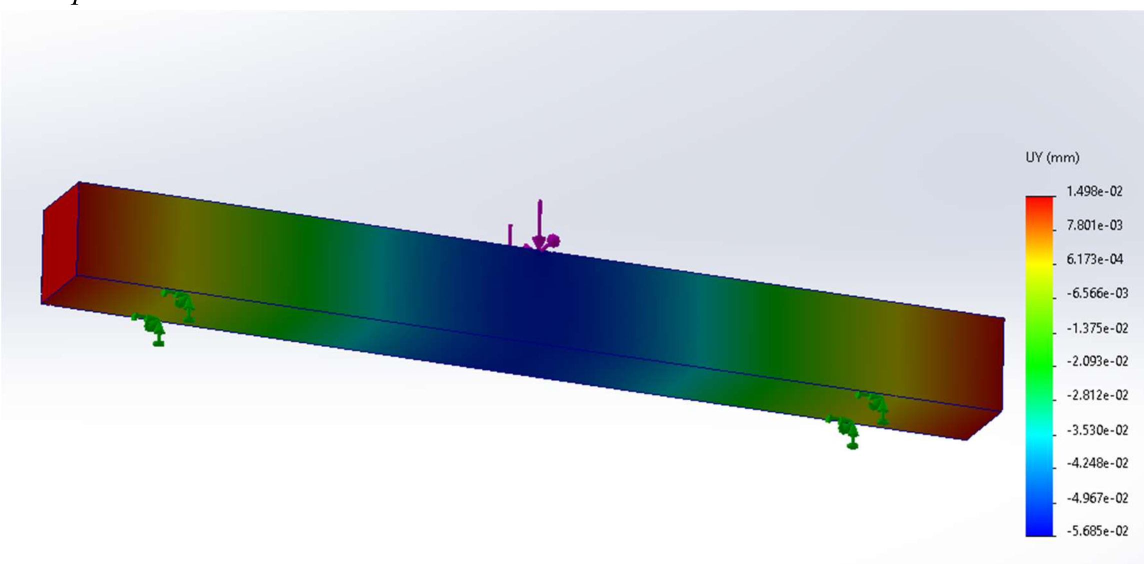
Y-Normal Strain



Shear Strain

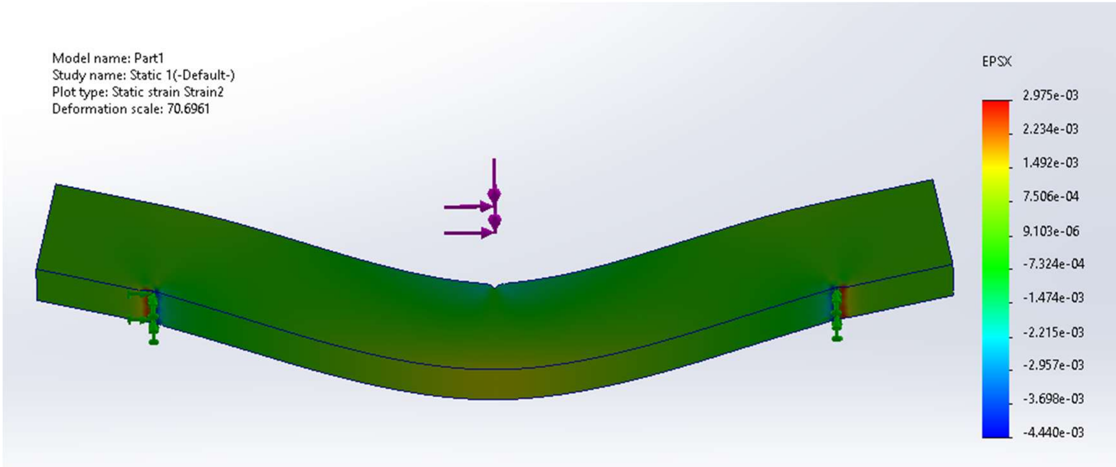


Y-Displacement

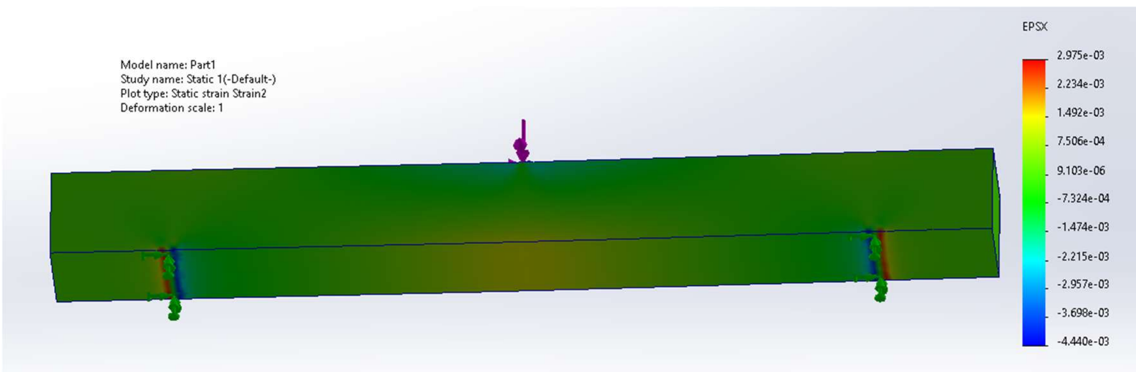


4000 N

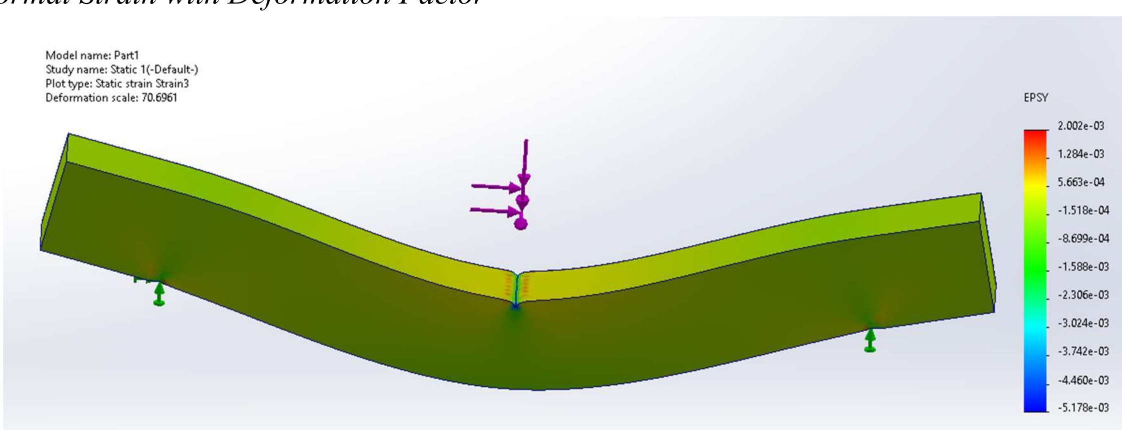
X-Normal Strain with Deformation Factor



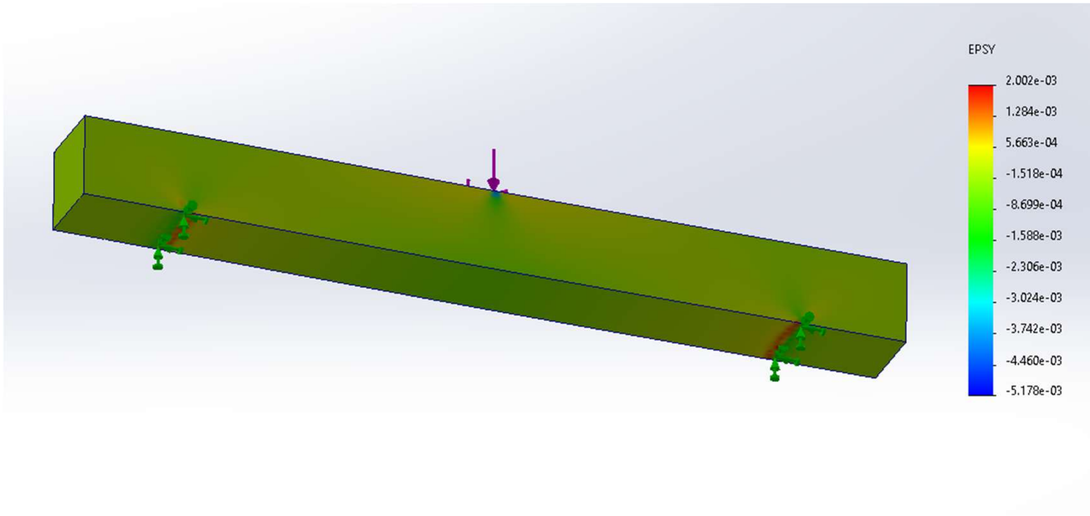
X-Normal Strain, True Scale



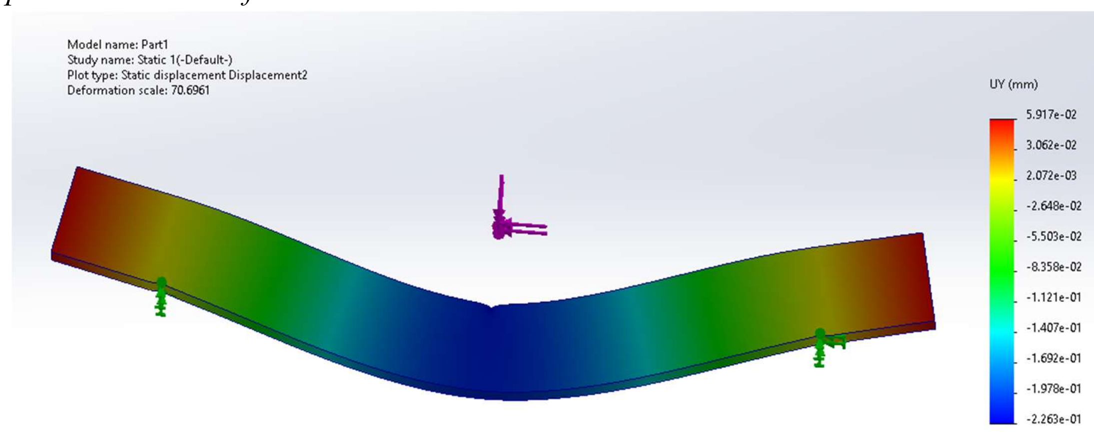
Y-Normal Strain with Deformation Factor



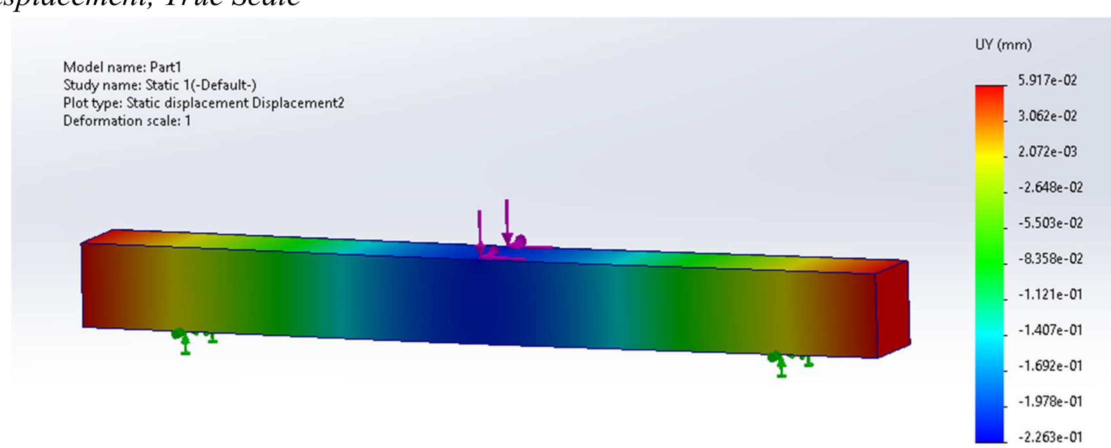
Y-Normal Strain, True Scale



Y-Displacement with Deformation Factor



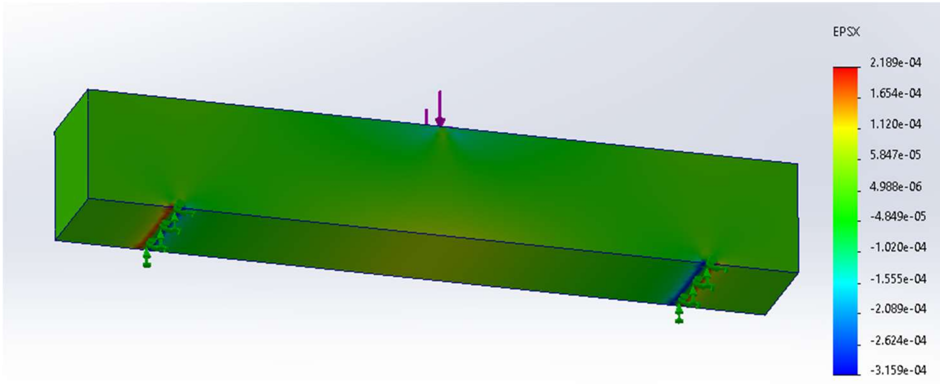
Y-Displacement, True Scale



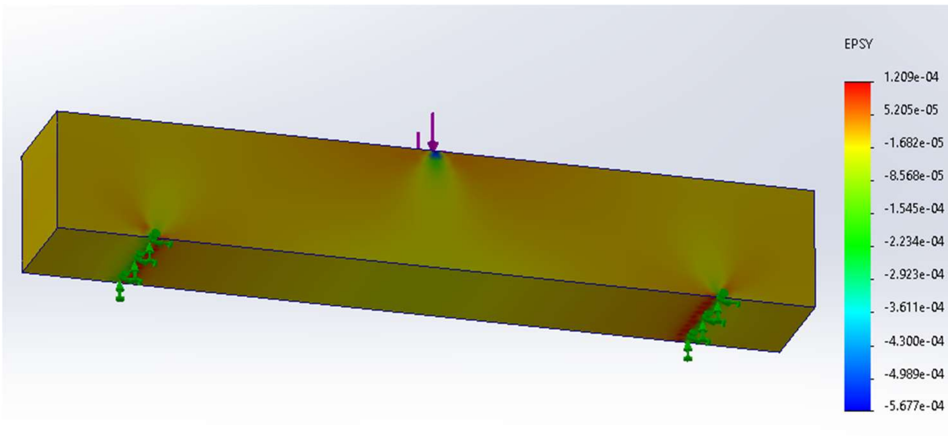
Large Solid Beam Simulations

1000 N

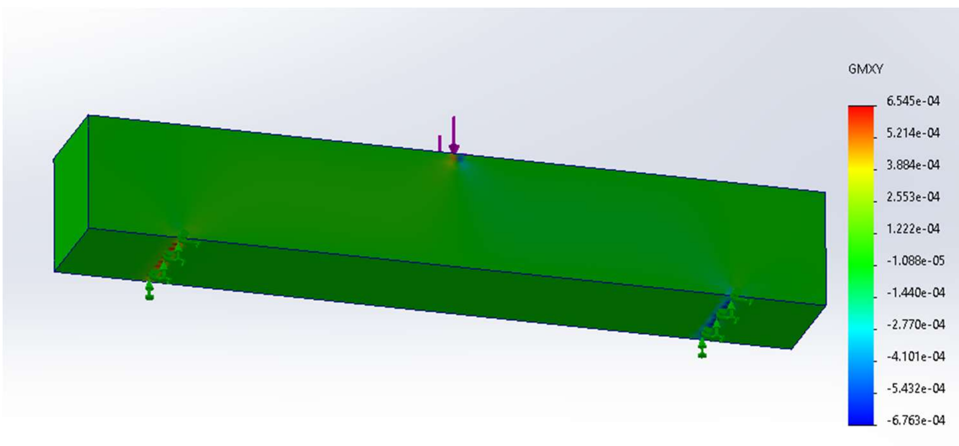
X-Normal Strain



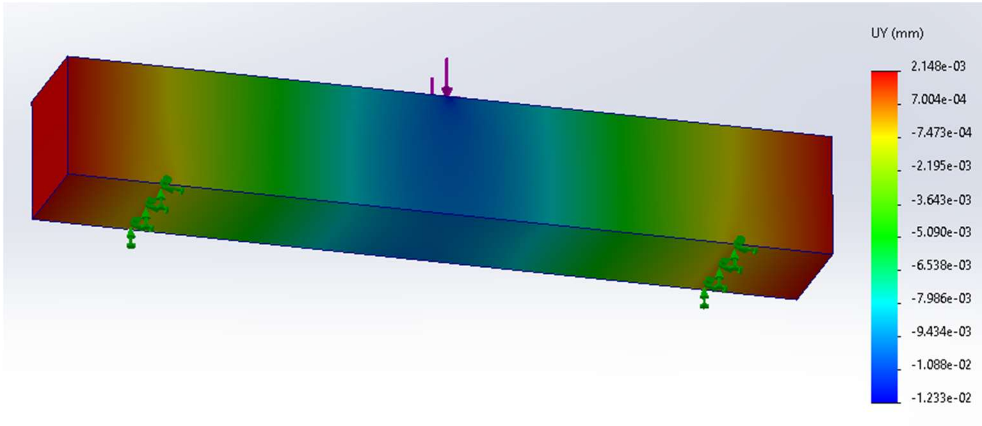
Y-Normal Strain



Shear Strain

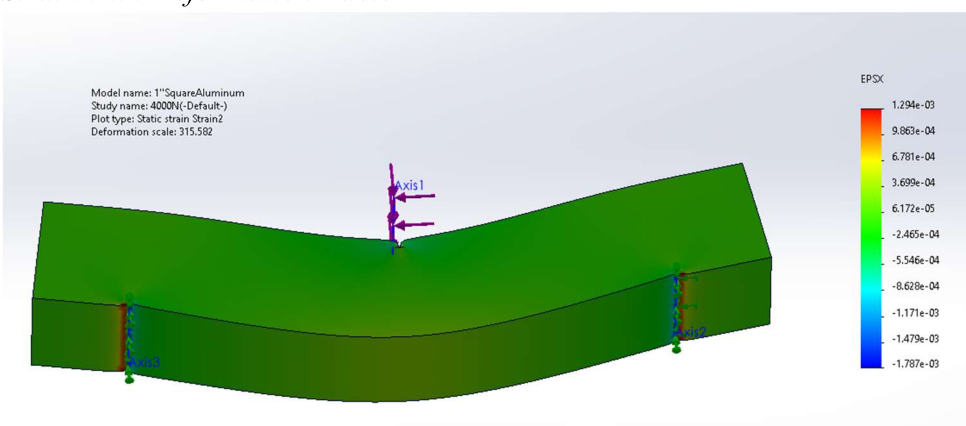


Y-Displacement

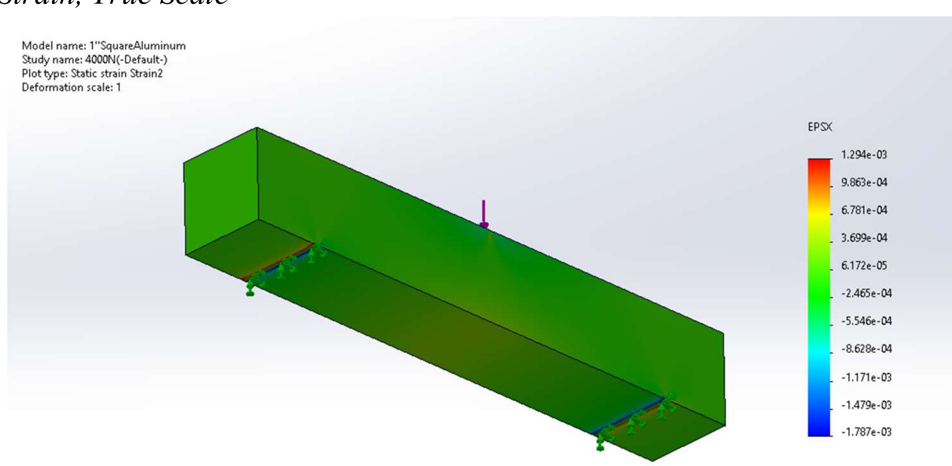


4000 N

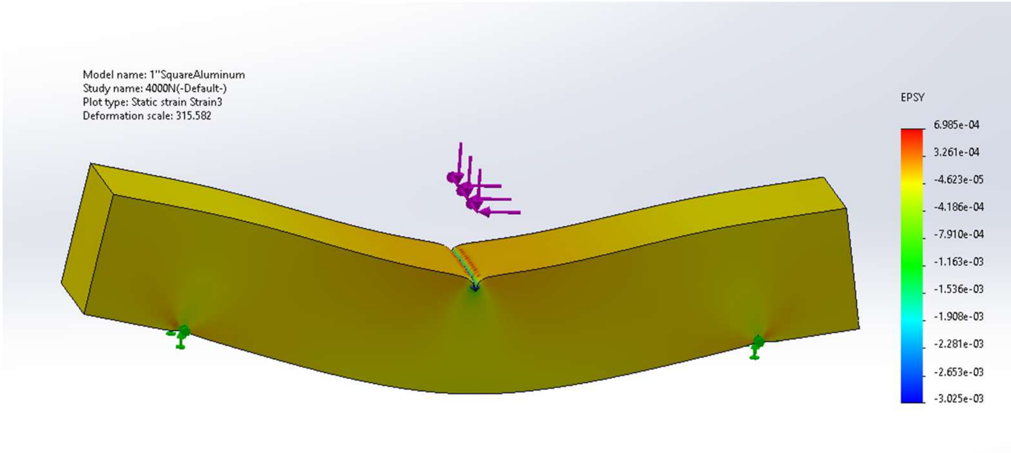
X-Normal Strain with Deformation Factor



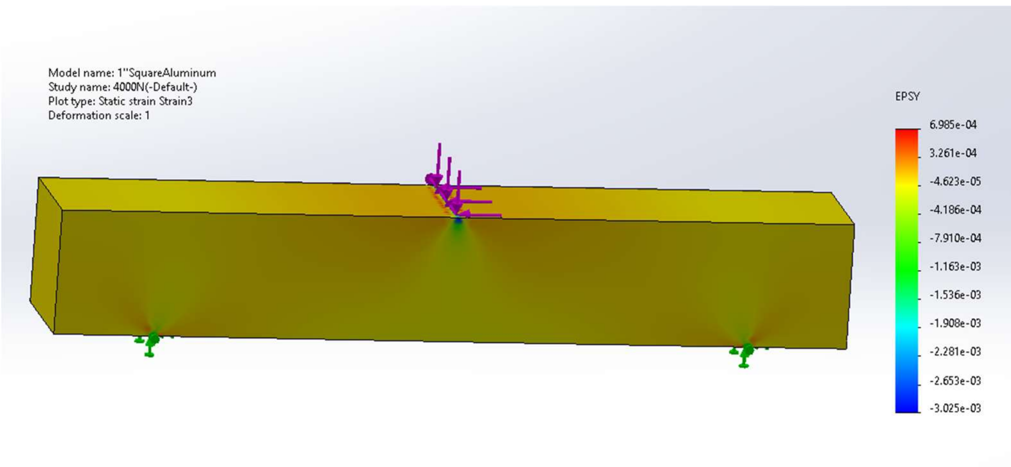
X-Normal Strain, True Scale



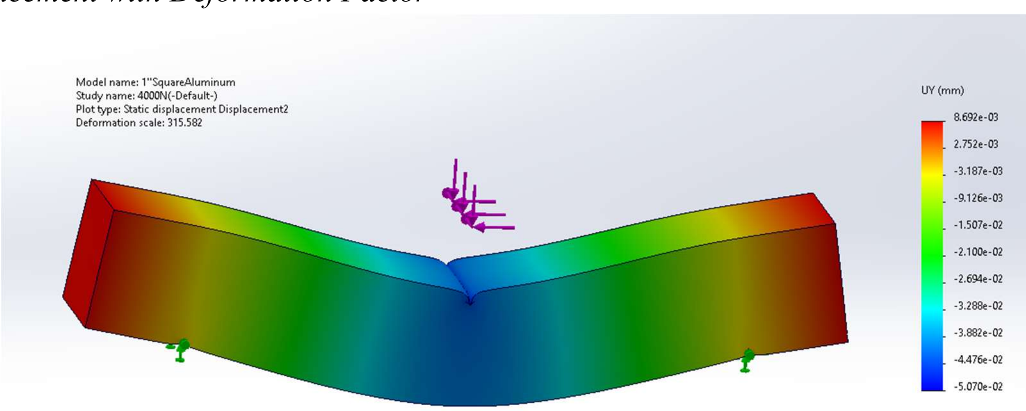
Y-Normal Strain with Deformation Factor



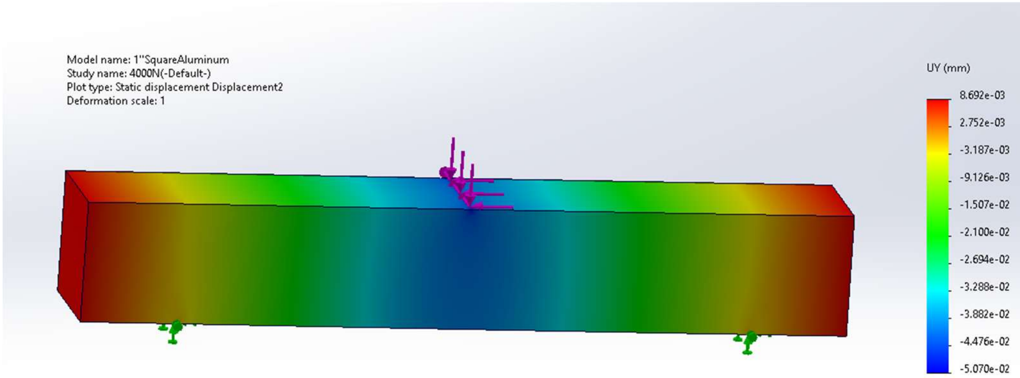
Y-Normal Strain, True Scale



Y-Displacement with Deformation Factor



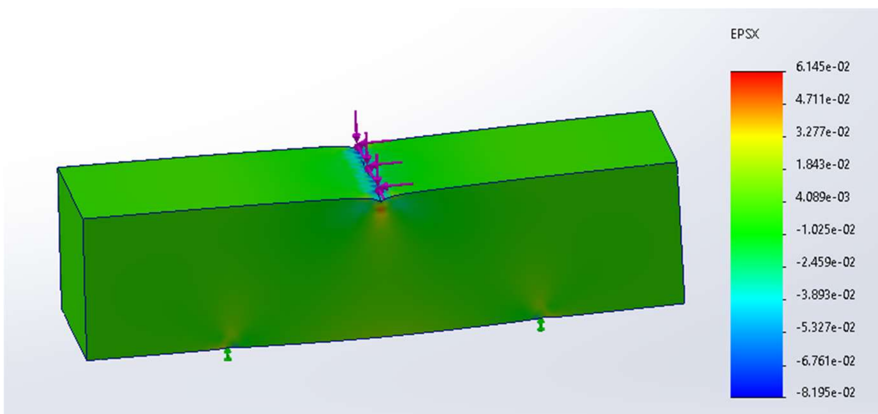
Y-Displacement, True Scale

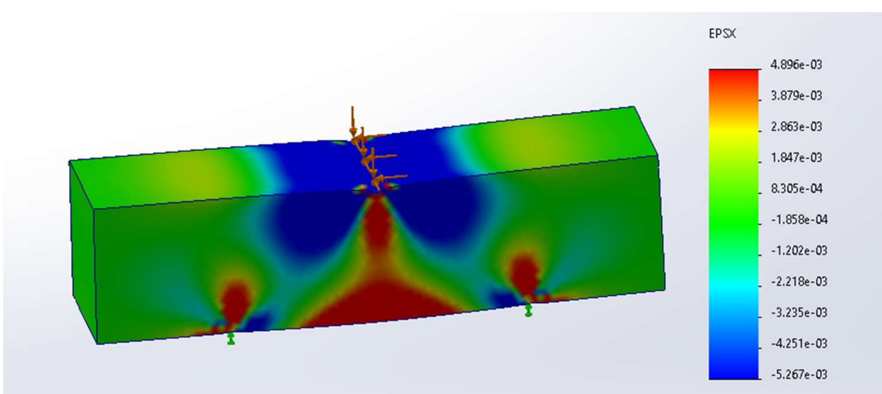


Polyurethane Solid Beam Simulations

38 N

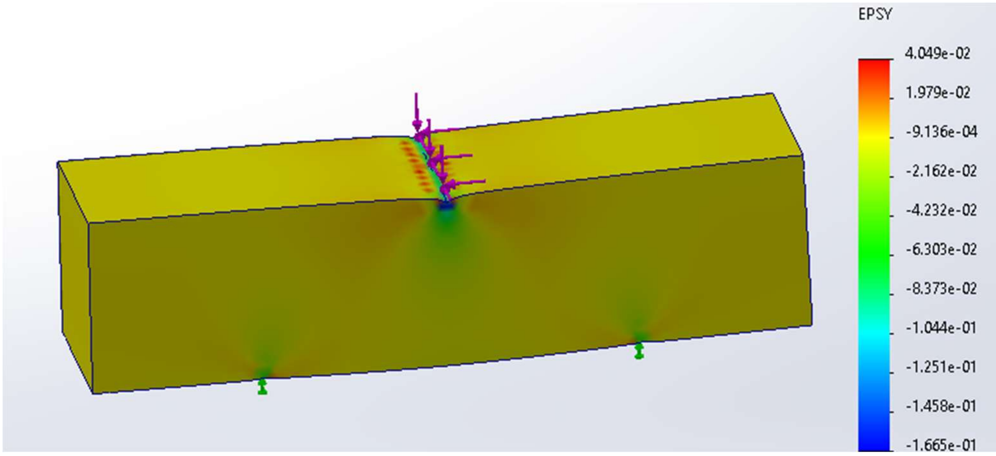
X-Normal Strain



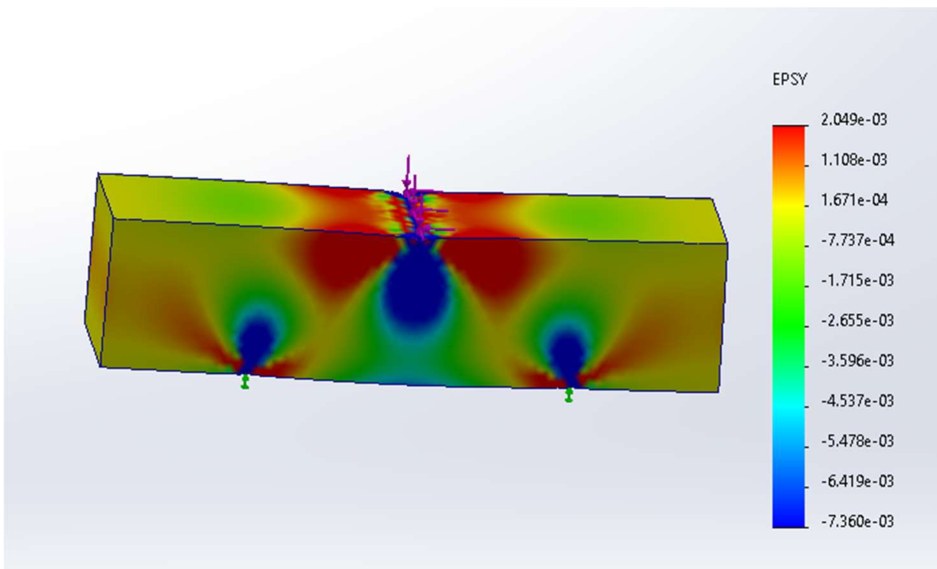


X-Normal strain with color bar limits set to experimental values

Y-Normal Strain

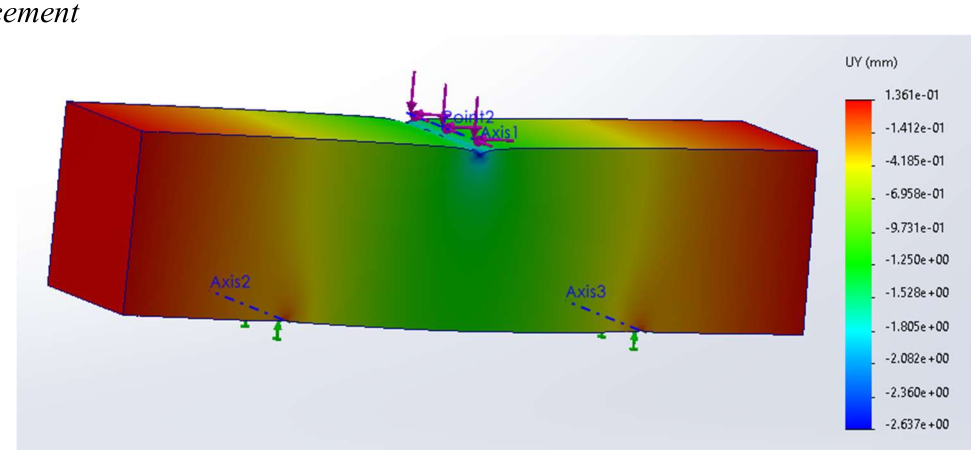


Top View



Y-Normal Strain with color bar limits set to experimental values

Y-Displacement



Appendix VI: M1 Style Procedure and Datasheet

Repeated Bending Tests with Digital Image Correlation

Broad Goal: Experimentally visualize and validate the principles of stress and strain by applying Digital Image Correlation (DIC) in a controlled bending test.

Specific Objective: Measure and compare the experimental strain distribution on an aluminum beam with theoretical predictions.

Test Setup:

Procedure:

<u>Data Collection</u>

1. Measure the beam dimensions and record them in the table provided in the *Bending Tests with Digital Image Correlation Datasheet.*
2. Roll the roller stamp on the ink pad thoroughly before applying the speckle pattern to the sample. Firmly apply a random speckle pattern to a painted side of an aluminum sample using the roller stamp. Use a ruler to make the center of the beam so that the load can be place appropriately.
    - **Do not touch** the speckle pattern as it may smudge.
    - If the pattern is too sparse or smudged, rotate to another clean, painted side and attempt to reapply the speckle pattern.
3. Mount the beam horizontally in a three-point bending fixture on the Instron testing machine, ensuring the span of the supports are aligned with the desired test length, the beam is centered on

the test stand, and the face of the beam with the speckle pattern is facing the camera. Make sure the loading head will contact the top face of the beam evenly to avoid unintended torsional effects.

4. Open the Vic-Snap 9 software and create a new project folder to store captured images. Set this folder as the working directory.
    - VIC-Snap is used to take images, VIC-3D (as explained later) will be used to process the images and extract data.
5. In Vic-Snap, switch to Speckle Images mode.
6. Perform a hand-calculation to determine the max load that could be applied before the beam reaches the plastic region.
    - Note: The python script includes these calculations and will serve as a check. This value, divided by a factor of safety, will be used as the maximum limit for the force applied by the Instron.
7. Decide on a reasonable load increment value such that there will be 20 evenly spaced load increments, this will depend on the maximum load calculated.
    - For example, if you have a max load of 4000 N, the load should be in 200 N increments starting from 0 N.
8. With no load applied, capture a reference image of the undeformed sample by pressing the space bar.
9. Ensure you are wearing appropriate safety equipment as directed by Lab Staff and Instructors.
10. Under the supervision of an instructor or qualified laboratory technician, use the Instron software to apply force incrementally at the center of the beam.
    - To zero the Instron on the beam, apply a 1N load. This will not cause any meaningful deflection to the beam and will serve as the zero point.
    - The maximum load applied should be approximately 50% of the calculated load. This is a reasonable factor of safety that will ensure that the beam does not undergo plastic deformation.
    - Do not exceed the material's yield strength, **the goal is to remain within the linear elastic region**.
11. After each static load increment, pause the Instron and capture a corresponding image.
    - Avoid taking images during active motion to ensure frame alignment.
    - Make sure to take note of the load the Instron applies. It will be similar, but not exactly the same as the load prescribed.
    - Make sure to fill out the Datasheet on the next page as appropriate.
12. Continue the loading sequence until sufficient data points across the elastic range are captured. Stop well before plastic deformation to preserve the sample for repeated testing.
13. Release the load on the sample and take a final image again under a zero-load condition.

Data Extraction

14. Open Vic-3D and load the previously captured calibration images. Select Calibrate stereo system from the Calibration menu and verify that the calibration score is low (Ideally less than 0.1).
15. Load the full set of speckle images taken from VIC-Snap into VIC-3D.

16. Select an Area of Interest (AOI) that includes the portion of the beam contacting the mounts and loading points.
17. Use the Suggest Subset Size tool, then set the step size.
18. Run the DIC analysis and inspect the resulting strain fields.
19. Save all project files and export relevant data for reporting.
20. Exporting the data: The data can be exported by selecting the Export item from the data menu. Export the data as a comma-separated variable. Save the .csv as LastName-Load.csv.
21. To analyze data via python script: Copy the path for the .LastName-Load.csv and paste into python script. Run the script.
22. The analysis script can analyze strains along cross sections.
23. If desired, samples can be unloaded and reused in future trials, provided they remain undamaged and within the elastic regime.

Bending Tests with Digital Image Correlation Datasheet

Beam Material	
Length	
Width	
Height	
Thickness (if applicable)	

Trial	Image Number	Force Input (N)	Force Applied (N)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			