ES 221 MECHANICS OF SOLIDS

REPORT
BY GROUP 8

Rupak Banerjee Nishit Mistry Utkarsh Srivastava Vidhi Shah Hardik Jain

OBJECTIVE

Our project mainly focuses on the various parameters to consider when a beam is under a load. Through this code, we have allowed engineers and students to explore and use a flexible tool for determining shear force diagrams, bending moment diagrams, slope, and deflection. When a beam is under load, deflection of the beam takes place. Pondering upon this can be very useful for the design makers so that they can make informed decisions.

PROBLEM STATEMENT

Our project is concerned with integrating the loads on the beam over the whole beam under various loading scenarios, such as point loads, uniformly distributed loads, triangular distributed loads, and moments. We aim to offer a thorough beam deflection analysis by utilizing integration techniques and getting the output. This will help students and engineers understand structural behavior and make well-informed design decisions for various engineering applications.

METHODOLOGY

Based on the code is the listed breakdown of the methodology used,

➢ The methodology used here is the integration method. The integral() function's trapezoidal rule is used in numerical integration to calculate the definite integrals of supplied functions.

$$\frac{dv}{dx} = F \qquad [1]$$

$$\frac{dm}{dx} = V \qquad [2]$$

$$M(x) = \int V dx \qquad [3]$$

Let the deflection be y,

$$EI\frac{d^2y}{dx^2} = M(x) \qquad [4]$$

$$EI\frac{dy}{dx} = \int M(x) + C \qquad [5]$$

$$EI\, y = \int\int M(x) + Cx + D \qquad [6]$$

➢ The formula [6] gives the deflection of the beam, while the formula [1] and [2] gives us the diagrams for the shear force and bending moment.
➢ By integrating the force distribution and the distance from the specified point over a certain period, the m( ) function determines the moment of a force distribution about a specific point. The values( ) function uses numerical integration and numerical techniques to iteratively calculate the beam characteristics by adding up the contributions from each individual load.
➢ First, we will configure starting variables and request the user's beam length and support locations.
➢ Then, ask the user to choose the load type, and necessary information such as load strength and the position of loading.
➢ Utilize the values function to determine the shear force, bending moment, slope, and deflection along the beam when the user signals that they are finished adding loads.
➢ Using Matplotlib, we plot the computed results for the shear force, bending moment, slope, and deflection and use the st.pyplot function in Streamlit to see the plots.

SIMPLIFICATION STEPS

We pondered various ways in which it could become a user-friendly tool and be used by laymen. So here are some of the ways that simplify the problem and

Initialization: The code initializes several session state variables and parameters, including load kinds, load values, support locations, and beam length.

User Interaction: Streamlit widgets, such as sliders and number input areas, enable users to enter parameters like beam length, support locations, and load kinds within the application.

Load Addition: By using the "Add load" option, users can apply loads to the beam. Session state variables hold the load parameters corresponding to the specified load type, which can be point moment, point force, constant force profile, or triangular force profile.

Real-Time Visualization: On adding a load to the beam, it generates an image depicting the location of the supports and the load. It clearly depicts the triangle force profiles, constant force profiles, point moments, point forces and their locations.

Calculations: When the user clicks the "Done with Force" button, the load parameters they have provided are used to calculate the shear force, bending moment, slope, and deflection along the beam. This process begins when the user has finished adding loads. Matplotlib plots are then used to depict these computed values, and the Streamlit program displays the graphs.

Integration Methods: We have done the basic processes from scratch. We didn't use any direct formulas, but the most basic and essential formulas, from which all this is derived from the integration using small sections of the beam, allow every possible case of the beam loading to be solved.

Iterative Approach: Session stating variables are updated in real-time as users add loads and interact with the application, guaranteeing that the computations and visualizations are up to current with the most recent user input.

NUMERICAL IMPLEMENTATION/ EXPERIMENTAL DETAILS

Link to the google colab which stores the entire code:
https://colab.research.google.com/drive/1bHQnmExkyZSRN6zHlI9_G-O8CYNOCbou#scrollTo=EAmQfF-TPwVz

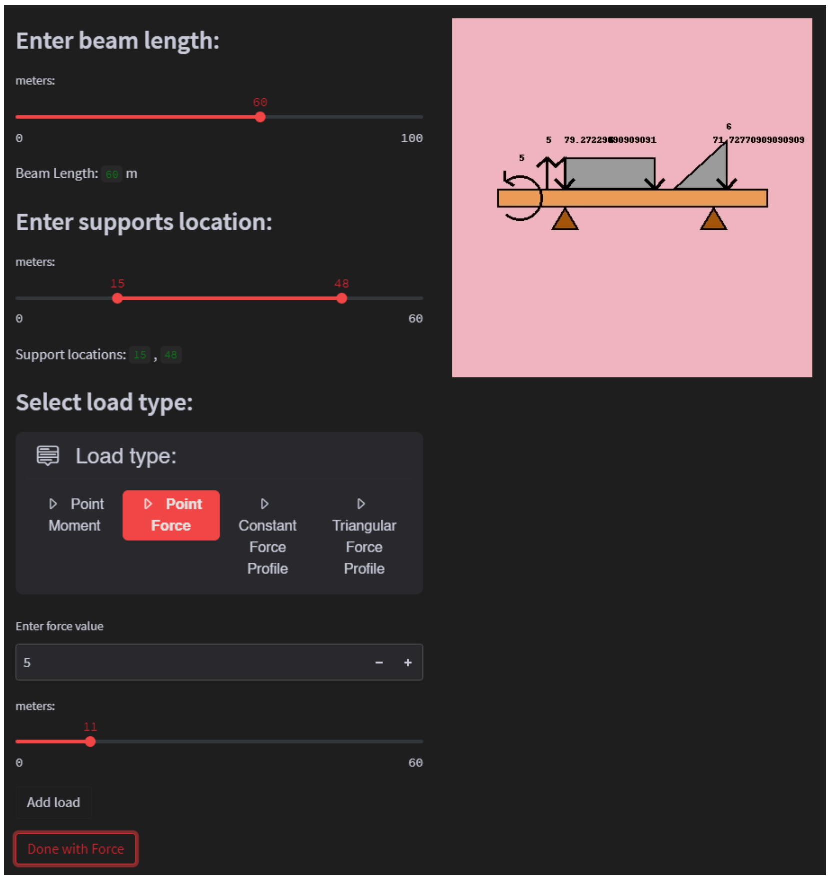
Here is the step-by-step explanation of the code:

1. To create the web application and produce visualizations, we imported streamlit, numpy, matplotlib.pyplot, and PIL (Pillow).
2. We began with inputting the beam length, support locations, and all the variety of loads.
3. We began by creating functions for mathematical operations like **integral** for integration and **m** for moment computation.
4. The types of input loads are categorized into 4 types, e.g.

- Point Moments - Stored in array f1
- Point Forces - Stored in array f2
- Continuous Load Profile - Stored in f3
- Triangular Load Profiles - Stored in f4

5. The length is stored in variable beam_length. The supports are located in variables, **support_1** and **support_2**.
6. These are then inputted into the values function that returns shear force, bending moment, slope (dy/dx) and deflection (y)
7. Matplotlib plots the computed numbers, which are then shown in distinct plots.
8. Lastly, the function uses the draw_beam from **PIL** to create a schematic drawing of the beam construction with the supplied loads. This function generates an image depicting the location of the supports and the different applied forces and moments acting on the beam. It clearly depicts the triangle force profiles, constant force profiles, point moments, point forces and their locations.

RESULTS AND DISCUSSION



Enter beam length:

meters: 60 (0 – 100)

Beam Length: 60 m

Enter supports location:

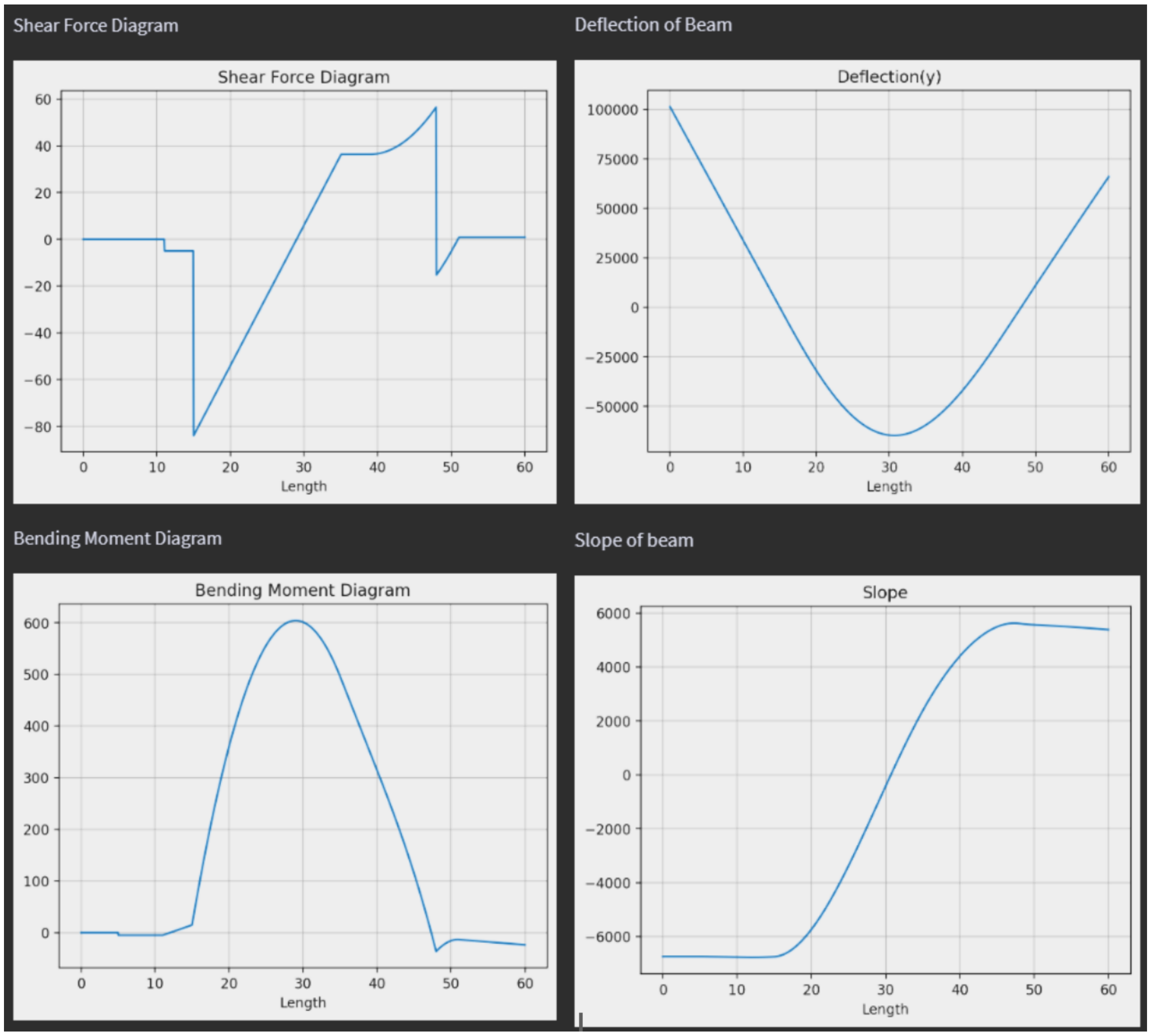
meters: 15, 48 (0 – 60)

Support locations: 15 , 48

Select load type:

Enter force value: 5

meters: 11 (0 – 60)



We have achieved the formation of the shear force diagram, Bending moment diagram, and the deflection of the beam by doing the complex coding combined with the graphical user interface.

Different Loading:

★ We have analyzed different types of loads, which tell us which kind of load is optimum for the given beam initial conditions.

★ The slope and deflection profiles are displayed to show how the beam deforms when loads are applied. The evaluation of structural integrity, the identification of possible failure sites, and the optimization of beam design to satisfy performance standards are all aided by these plots.

CONCLUSION

In conclusion, the code we developed for assessing beam deflection under varied loads was a good outcome of our project. We calculated the shear force, bending moment, slope, and deflection by using numerical integration techniques, giving engineers and students important insights into the behavior of the structure. Its dependability was confirmed by comparing the tool's accuracy to analytical solutions.

Future improvements involve enhancing numerical techniques and including other load kinds. Our code helps optimize structural designs in all the areas considered, guaranteeing efficiency and safety in engineering projects.

THANK YOU!