ES:221 MECHANICS OF SOLID
2D TRUSS SOLVER
GROUP-9

Manas Kalal - 22110138
Nikhil Kumar Lal - 22110167
Mallepogula Charanteja - 22110136
Nandkishor kumar pandit - 22110164

Abstract: These reports include our finding of building a code for solving 2D truss structure using python. The output prints a table which shows the members which along with the points and the force which the members is bearing also it shows that if it is in compression or tension. Along with the colorful visualization plot of the truss.

I.OBJECTIVES

We all had once seen a truss in our life whether it is the kite structure(Fig.1) in the main academic building area or any other structure and ever wondered how and why there are so many rods in various orientations holding up a heavy load.

Fig 1. Academic Area

The basic idea of truss is to distribute large loads over distanced supports or joints. A simple basic truss could start form a triangle and go to a very complex geometry. There are various types of truss which are listed in fig.2 and all the truss mentioned in the image could be solved using our code.

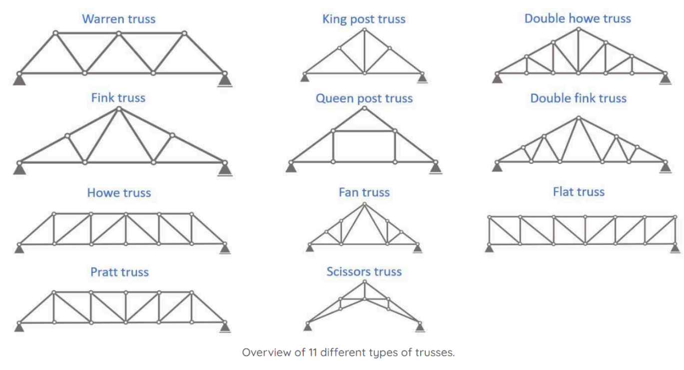

Fig 2. Types of truss[2]

Our main objective in this project is to write an algorithm which helps our fellow mates and juniors to analyze 2D trusses and get a better understanding of how trusses. Basically we will be developing a python code which takes input from the user or various parameters of the truss like joint coordinates, members and external loads etc and provides the analysis of the truss.

II.PROBLEM STATEMENT

The project focuses on analyzing forces within truss structures for the given truss parameters . It aims to develop a python code that accurately calculates internal forces in truss members (tension or compression) based on applied loads and support input from the user.

III.METHODOLOGY

The methodology used in the code revolves around the traditional method of solving truss as taught in class. As the truss was just 2D we had neglected the mass of the members and area of it etc while coding.

Load Class: Defines a load and connects it to point (joint index) finds angle (measured from positive x-axis), and magnitude (force magnitude). All these parameters are taken as input and if the load is made at an angle this class has functions which calculate the x and y components of the load applied at that point.

Status Class: Represents a truss member with attributes for start and end points, force (calculated), and whether the member is tension or compression.

Truss Class: It is the main class which calculates the force at individual point

- **c_moment()**: Computes the moment due to a load around a reference point.
- **s_rxn()**: Determines the support reactions at fixed and roller supports.
- **is_determinate():** This function checks if we can figure out all the forces and support reactions in the truss by comparing the number of unknowns (forces we don't know yet) like the reactions with the equations we have (from physics) using moment and force balance equations.
- **get_members_from_point():** If you give it a point in the truss, it tells you which parts (members) of the truss connect to that point.
- **is_point_solvable()** and is_point_solved(): Check if a point's member forces can be solved or are already solved.
- **solve_point()**: Calculates member forces at a specific point using equilibrium equations at individual points using the rules of force balance and knowing the forces acting on that point.
- **solve_all():** It keeps using the combination of the above function for every point in the truss until we know all the forces everywhere and the supports reactions, so the whole truss is solved
- **plot_truss()**: Generates a graphical representation of the solved truss structure, including member forces and loads using matplotlib.

Input Handling: User input is taken for defining the bridge geometry (points and members), applied loads, and support types (roller and pin supports).

IV.NUMERICAL IMPLEMENTATION

The numerical implementation of the truss solver involves the following key steps and algorithms:

Input: User gives input such as defining the bridge geometry (points and members), specifying applied loads, and identifying support types (roller and pin supports).

Initialization: The Bridge class initializes the truss structure based on user input, creating instances of points, members, and loads.

Static Determinacy Check: The is_determinate() function verifies if the truss structure is statically determinate, ensuring solvability of member forces and support reactions.

Support Reaction Calculation: The solve_support_reactions() method calculates the support reactions at pin and roller supports using moment equilibrium equations.

Member Force Calculation: The solve_point() function iteratively calculates member forces at each point using equilibrium equations.

Iterative Solution: The solve_all() method iteratively solves for support reactions and member forces until we get the force for each member, ensuring equilibrium across the entire truss structure.

Graphical Representation: The plot_truss() function generates a graphical representation of the solved truss, depicting member forces, applied loads, and support types for visualization and analysis.

Output: The solver outputs member forces (in Newtons) and identifies whether they are in tension or compression using the member class functions.

V.RESULTS AND DISCUSSION

Upon implementing the truss solver and analyzing a sample bridge structure, the following results were obtained:
We had solved 2 truss problems using this first is as follows

Question 1 Find all internal forces in the truss ?

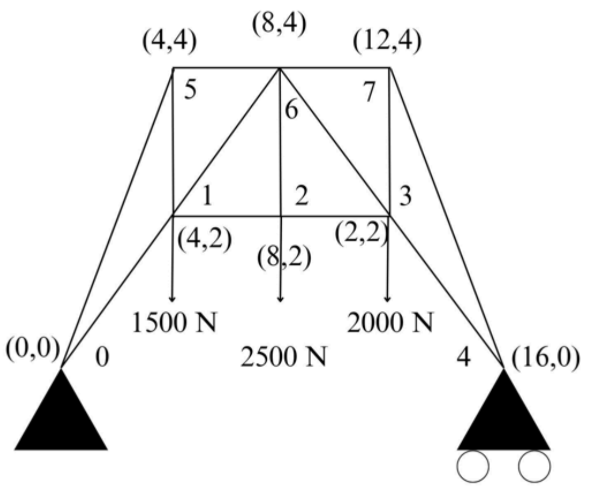

Fig 3. Truss question-1[1]

```
Please provide the following input for your bridge:
Enter the number of points in your bridge: 8
Enter x and y coordinates for point 0: 0 0
Enter x and y coordinates for point 1: 4 2
Enter x and y coordinates for point 2: 8 2
Enter x and y coordinates for point 3: 12 2
Enter x and y coordinates for point 4: 16 0
Enter x and y coordinates for point 5: 4 4
Enter x and y coordinates for point 6: 8 4
Enter x and y coordinates for point 7: 12 4
Enter the number of members in your bridge: 13
Enter start and end points for member 0: 0 1
Enter start and end points for member 1: 0 5
Enter start and end points for member 2: 1 2
Enter start and end points for member 3: 1 5
Enter start and end points for member 4: 1 6
Enter start and end points for member 5: 2 6
Enter start and end points for member 6: 2 3
Enter start and end points for member 7: 3 6
Enter start and end points for member 8: 3 4
Enter start and end points for member 9: 4 7
Enter start and end points for member 10: 3 7
Enter start and end points for member 11: 6 7
Enter start and end points for member 12: 6 5
Enter the number of loads on your bridge: 3
Enter the point where load 0 is applied: 1
Enter the angle of load 0 (in radians): -1.57
Enter the magnitude of load 0: 1500
Enter the point where load 1 is applied: 2
Enter the angle of load 1 (in radians): -1.57
Enter the magnitude of load 1: 2500
Enter the point where load 2 is applied: 3
Enter the angle of load 2 (in radians): -1.57
Enter the magnitude of load 2: 2000
Enter the index of the roller support: 4
Enter the index of the pin support: 0
```

Fig 4. Terminal screenshot for an sample user input

Fig.3 shows how we had given input to the code for our question like
First it asks the number of points or joints and after that we need to give the x,y coordinate separated by a space then it asks the number of members then we need to specify the number of members or rods and after that we need to give the input of which points make the member like a member would be represented by 2 points so for that we had made 0 1 that means point 0 connects to point 1 and there is a member in between them. Next the code asks for the number of external loads after that it take 3 information of the load that is at which point the load is acting like 0 or 9 etc then we need to specify the angle that it makes with the x axis in radians for example a downward force is having a angle of pi/2 = -1.57 then it asks the point where is roller joint and fixed joint and then it displays the result.

Theoretical Results

Member	Force(N)	Types
(0,1)	64367.23	Tension
(0,5)	8134.86	Compression
(1,2)	8500.005	Tension
(1,5)	5752.82	Tension
(1,6)	3074.598	Compression
(2,3)	8501.60	Tension
(2,6)	2498.64	Compression
(3,4)	6988.74	Tension
(3,6)	2514.59	Compression
(3,7)	6249.56	Tension
(4,7)	8838.23	Compression
(6,7)	6249.56	Compression
(6,5)	5750	Compression

Table 1. Theoretical results question 1

Member	Force (N)	Type
(0, 1)	6438.04	Tension
(0, 5)	8136.79	Compression
(1, 2)	8505.97	Tension
(1, 5)	5753.58	Tension
(1, 6)	3073.26	Compression
(2, 6)	2500	Tension
(2, 3)	8503.98	Tension
(3, 6)	2516.91	Compression
(3, 4)	6989.05	Tension
(4, 7)	8840.52	Compression
(3, 7)	6251.19	Tension
(6, 7)	6251.19	Compression
(6, 5)	5753.58	Compression

Fig 4. Table code output for question-1

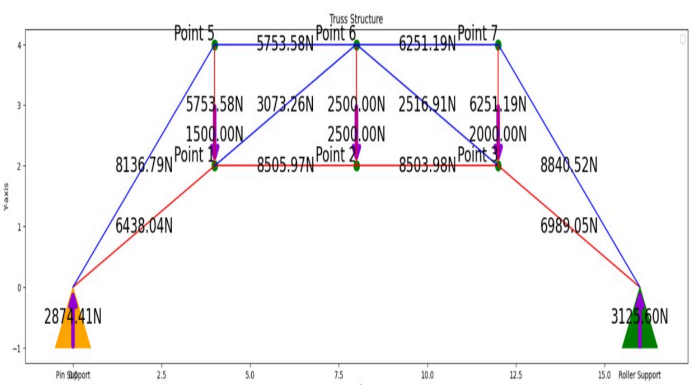

Fig 5. Truss output from code for question-1

In the above plot the red color members indicate the member in tension while the blue color members indicate the member in compression, the orange triangle is a fixed joint while the green triangle is a roller joint the forces are represented using a purple color arrow the values of force in Newton are also mentioned alongside. A bigger image of the same is attached in the end (Fig10).

Question-2:- Find all internal forces in the truss?

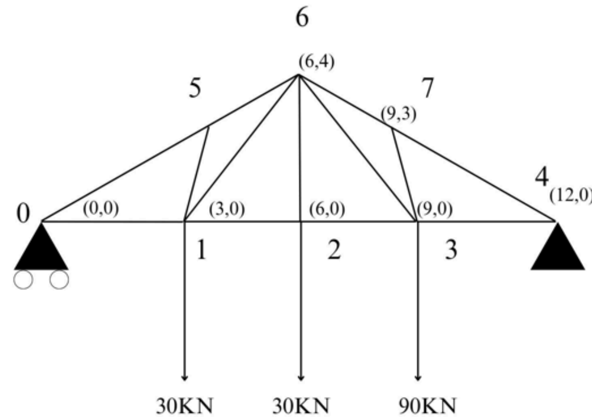

Fig 6. Truss question-2

```
Please provide the following input for your bridge:
Enter the number of points in your bridge: 8
Enter x and y coordinates for point 0: 0 0
Enter x and y coordinates for point 1: 3 0
Enter x and y coordinates for point 2: 6 0
Enter x and y coordinates for point 3: 9 0
Enter x and y coordinates for point 4: 12 0
Enter x and y coordinates for point 5: 3 3
Enter x and y coordinates for point 6: 6 4
Enter x and y coordinates for point 7: 9 3
Enter the number of members in your bridge: 13
Enter start and end points for member 0: 0 1
Enter start and end points for member 1: 0 5
Enter start and end points for member 2: 1 2
Enter start and end points for member 3: 1 5
Enter start and end points for member 4: 1 6
Enter start and end points for member 5: 2 3
Enter start and end points for member 6: 2 6
Enter start and end points for member 7: 3 4
Enter start and end points for member 8: 3 6
Enter start and end points for member 9: 3 7
Enter start and end points for member 10: 4 7
Enter start and end points for member 11: 5 6
Enter start and end points for member 12: 6 7
Enter the number of loads on your bridge: 3
Enter the point where load 0 is applied: 1
Enter the angle of load 0 (in radians): -1.57
Enter the magnitude of load 0: 30000
Enter the point where load 1 is applied: 2
Enter the angle of load 1 (in radians): -1.57
Enter the magnitude of load 1: 30000
Enter the point where load 2 is applied: 3
Enter the angle of load 2 (in radians): -1.57
Enter the magnitude of load 2: 90000
Enter the index of the roller support: 0
Enter the index of the pin support: 4
```

Fig 7. Terminal screenshot for an sample user input

Member	Force(N)	Types
(0,1)	59998.63	Tension
(0,5)	84850.66	Compression
(1,2)	67474.98	Tension
(1,5)	39997.23	Tension
(1,6)	12598.6	Compression
(2,3)	67450.8	Tension
(2,6)	29998.89	Tension
(3,4)	89879.5	Tension
(3,6)	37547	Tension
(3,7)	59997.98	Tension
(4,7)	127277.80	Compression
(6,7)	94866.67	Compression
(6,5)	63244.76	Compression

Table 2. Theoretical results question 2

Member	Force (N)	Type
(0, 1)	60000	Tension
(0, 5)	84852.8	Compression
(1, 2)	67476.1	Tension
(1, 5)	40000	Tension
(1, 6)	12500	Compression
(2, 3)	67452.2	Tension
(2, 6)	30000	Tension
(3, 4)	89880.5	Tension
(3, 6)	37500	Tension
(3, 7)	60000	Tension
(4, 7)	127279	Compression
(5, 6)	63245.5	Compression
(6, 7)	94868.3	Compression

Fig 8. Table code output for question-2

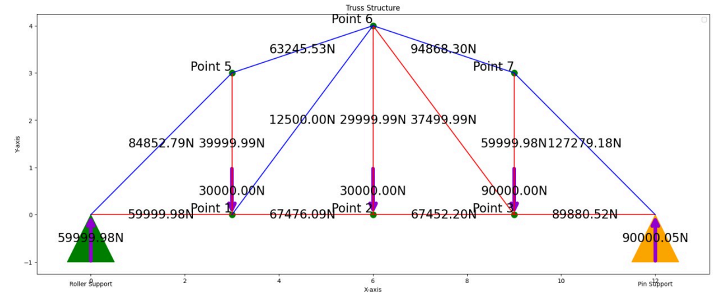

Fig 9. Truss output from code for question-2

A bigger image of the plot is attached in the end(Fig.11).

VI.CONCLUSION

In conclusion we are confident that using this code we can easily find the solution to truss structure and also while implementing truss structure in our physical world one could easily try making a different setup and could verify using this code if it is a better solution or not.

Also we had learn a lot of new things in this project like how to collaborate in team and code together a lot of time we use to correct one other like while brainstorming for the code if one person defines a algorithm rest all of us try to see the worst case scenario and see if it could cater all the possible cases or not.

VII.ACKNOWLEDGMENT

We would like to thank Professor Harmeet Singh for teaching us.

VIII.REFERENCE

[1] T. A. (2020, August 23). *TRUSS BY SECTION METHOD SOLVED PROBLEM 2 IN ENGINEERING MECHANICS IN HINDI*. YouTube. https://www.youtube.com/watch?v=u_4OdfNfgCc
YouTube.
https://www.youtube.com/watch?v=u_4OdfNfgCc

[2] Ernst, L., Ernst, L., & Ernst, L. (2023, November 21). *11 types of trusses [The MOST used]*. Structural Basics - Structural Engineering for Everyone. https://www.structuralbasics.com/types-of-trusses

BIGGER PLOTS

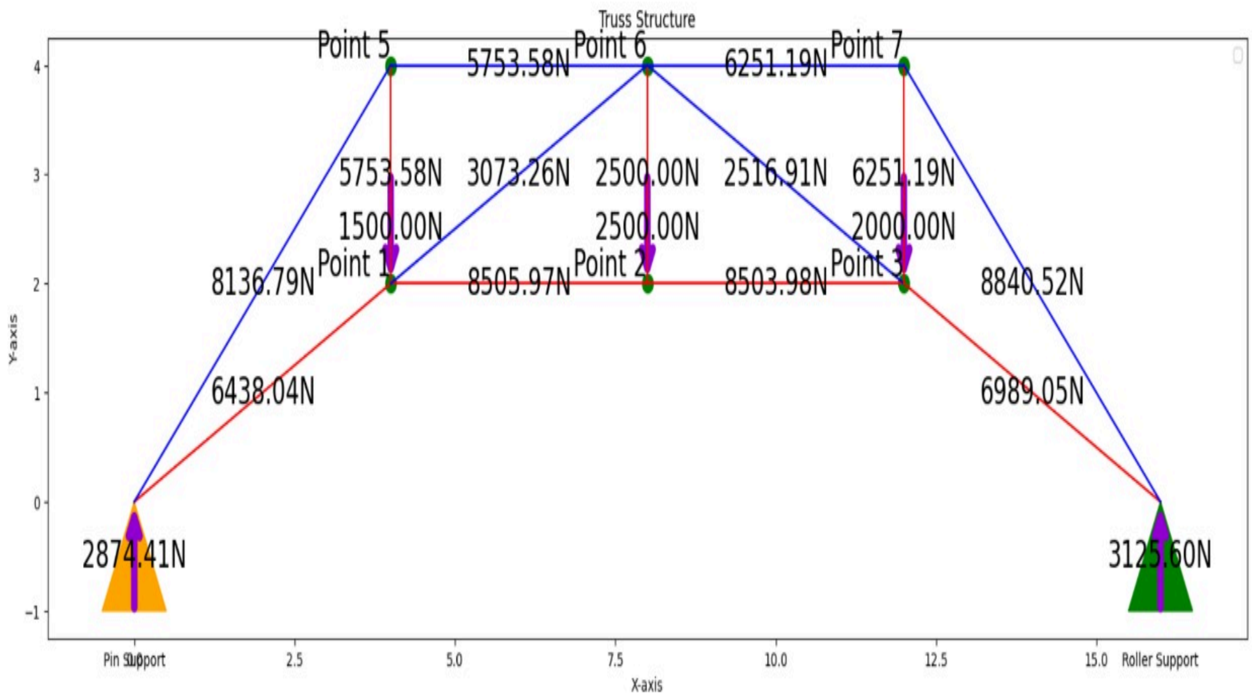

Fig 10. Truss output of code for question-1(bigger plot)

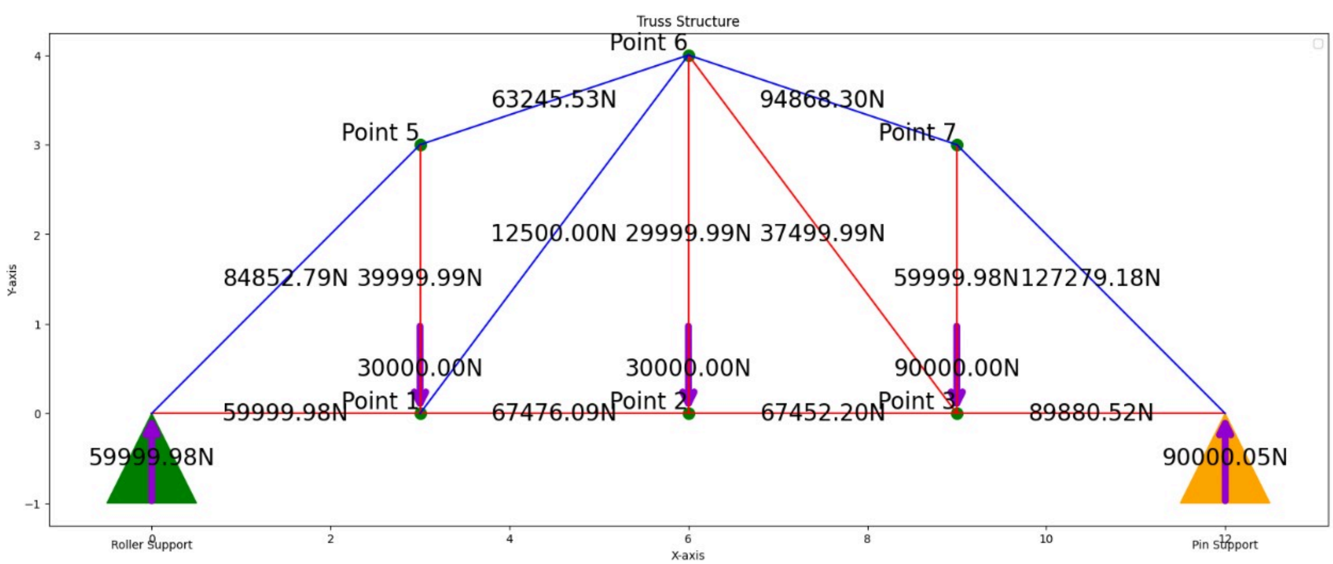

Fig 11. Truss output of code for question-2(bigger plo