

Analysis of a Truss System in 2D using Manual, Mathematical & Numerical Methods

Coursework Assignment

A report submitted to Coventry University's Branch Campus in Egypt at The Knowledge Hub Universities in partial fulfillment of the requirements for the module of

KH5034MAA: Analytical Modeling

a part of Year 2 in

BEng (Hons): Mechanical Engineering
Coventry University

Authors

Peter Metry - 202101686

Ali Hesham - 202200782

Rana Fawzy - 202101610

Shehab El-Debeiky - 202200616

Supervised by

Nour Ali, M.Sc. Reham Elsayah, M.Sc. Sameh Eid, PhD. Sara Abdelfattah, M.Sc.

December 5, 2023

Abstract

This report presents a comprehensive analysis of a truss structure using manual methods, numerical simulations with *ANSYS*, and a *MATLAB* implementation. The manual approach involves Free Body Diagrams to determine member forces, while *ANSYS* employs finite element analysis for detailed insights. The *MATLAB* implementation generates a global stiffness matrix, providing results consistent with *ANSYS*. The convergence of results from manual, numerical, and mathematical approaches enhance the overall reliability of the structural analysis.

Introduction

Structural analysis of truss systems is crucial for understanding their behavior under various loading conditions. This report combines manual, numerical, and mathematical methods to analyze a truss structure. The manual method uses Free Body Diagrams and Newton's 1st Law of Motion to determine forces in each member. *ANSYS* employs finite element analysis, while *MATLAB* implements a mathematical model to calculate nodal displacements and forces.

Aims and Objectives:

1. Conduct manual analysis to determine member forces in the truss structure.
2. Utilize *ANSYS* for numerical simulations to gain detailed insights into structural behavior.
3. Implement a mathematical model using *MATLAB* to calculate nodal displacements and forces.
4. Validate and compare results obtained from manual, numerical, and mathematical methods.

Problem Description: The truss structure under investigation consists of multiple members subjected to external forces. The objective is to understand the distribution of forces and displacements within the structure.

Formulation and Hypothesis: Manual analysis involves applying Newton's 1st Law of Motion to joints, assuming static equilibrium. *ANSYS* employs finite element analysis to model the truss, considering material properties, different cross sections, and boundary conditions. *MATLAB* formulates a global stiffness matrix to calculate nodal displacements and forces.

Results: The convergence of results from manual, numerical, and mathematical methods validate the structural analysis. *MATLAB* results closely match those obtained from *ANSYS*, demonstrating the reliability of the implemented mathematical model.

The Manual Method

Free Body Diagram

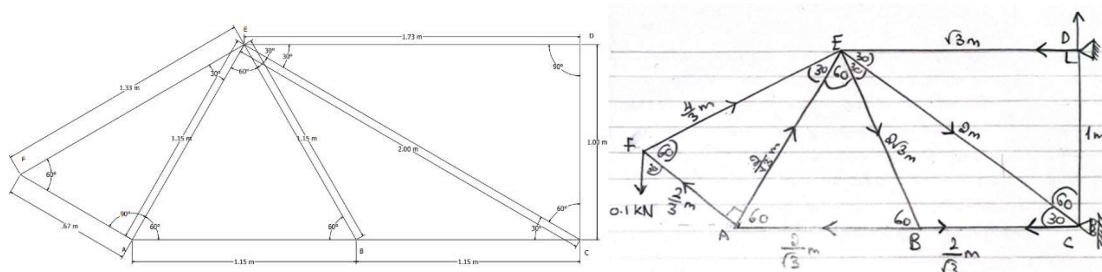


Fig 1.1 & Fig 1.2: Scaling the truss using AutoDesk AutoCAD (left) & The manual-sketched Free Body Diagram

FBD is a visual representation of Newton's 1st Law of Motion used to determine the forces acting on each individual member of the truss. The following truss consists of **9 members**. Given that $F_{fa} = -0.1 \text{ kN}$. The truss is at static equilibrium therefore, Forces are calculated by using Newton's 1st law $\sum F_x = \text{zero}$, $\sum F_y = \text{zero}$ *

*Kindly find the Free Body Diagram's calculations in the Appendix.

Results

The manual analysis provisionally shows a clear tension in ED, which can be considered the highest among the rest of the bars/elements. The Bar/Element BE also showed no stress compared to the rest of the system.

By referring to bar/element BE's location, we can easily anticipate that it is subjected to bending stress. **Considering the absence of any axial stress on this beam, we can deduce that it serves as the truss' neutral axis.**

By comparing and evaluating the stress and stiffness values of the truss system (See Appendix: Page 16) it was obvious that Bar/Element EC had higher stiffness but still had lower compression magnitude than Bar/Element BE, due to its bigger cross section. This validates the theory of bar/element BE being the neutral axis of the whole structure.

	EC	BE
δ	$\frac{EA}{L} = \frac{160 \times 10^9 \times \pi (7 \times 10^{-3})^2}{0.014}$ $\delta_{EC} = 1.76 \times 10^9 \text{ (N/m)}$	$\frac{EA}{L} = \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01}$ $\delta_{EC} = 1.26 \times 10^9 \text{ (N/m)}$
σ	$\frac{F}{A} = \frac{0.198 \times 10^3}{\pi (7 \times 10^{-3})^2}$ $\sigma_{EC} = 2.52 \text{ MPa}$	$\frac{F}{A} = \frac{0}{\pi (5 \times 10^{-3})^2}$ $\sigma_{EB} = 0 \text{ MPa}$

Safety Factor

Factor of safety is an important concept in the engineering design field, used to take into consideration any uncertainties, and unpredicted conditions, while construction of the structure, and used components. It works as a relation between the ultimate load that has been calculated and the load that can the structure hold realistically. The purpose of taking the factor of safety into consideration is to increase the reliability of the structure by making sure that the structure components have a margin to stand against any unpredictable situations.

As per the givens in the brief, we concluded that the material used for the truss is, **Steel, Structural**, which has Young's Modulus, $E = 160 \text{ GPa}$, Poisson ratio, $\nu = 0.3$, and Yield stress, $\gamma = 250 \text{ MPa}$.

To obtain the actual stress experienced in the truss the maximum force must be taken at point D with a force of 288.67 and divide it by the cross-sectional area with a diameter of 10 mm.

$$\text{Actual Stress} = \frac{\text{Force}}{\text{Area}} = \frac{288.67}{\pi (5 \times 10^{-3})^2} = 3.675 \times 10^6 \text{ Pa}$$

Therefore, the factor of safety that was used for this system can be obtained from the following equation:

$$SF = \frac{\text{Yield Stress}}{\text{Actual Stress}} = \frac{250}{3.675} = 68$$

The obtained factor of safety obtained is 68 which is a very high factor of safety, the reason for this however is the fact the structural steel is subjected to very low amounts of force so the resultant actual or working stress is very low in comparison with its yield point resulting in a large factor of safety.

In order to find out when the structure will fail if the force is gradually increased maximum yield strength should be used with the smaller cross-sectional area.

$$\text{Force} = (250 \times 10^6) \times \pi (5 \times 10^{-3})^2 = 19.63 \text{ kN}$$

When a force is gradually increased to the structure, it undergoes various stages of deformation. The material experiences elastic deformation, where it changes shape reversibly within the elastic limit, following Hooke's Law. As the force increases, the material reaches its yield point, marking the transition to plastic deformation. In this phase, the material undergoes permanent deformation even when the force is removed.

In the plastic region, which is relatively small for structural steel a significant increase in strain and minimal increase in stress occurs. This phase may involve localized deformation called necking, particularly in ductile materials like metals. The force continues to increase until the material reaches its ultimate strength, the maximum stress it can withstand.

If the force surpasses the ultimate strength, the material undergoes failure, resulting in fracture or rupture. The mode of failure depends on the material's properties, for this material because it is a metal ductile fracture occurs.

Understanding the stress-strain curve of a material is crucial in engineering and material science for designing structures capable of withstanding specific loads and conditions.

The Numerical Method

The finite element method (FEM) is considered as a numerical analysis technique for approximation solutions to a variety of design engineering problems and range of boundary problems. In the numerical analysis method is subdivided into several small elements that are joined at nodes and this process is known as **Discretization**.

To solve the truss, The Numerical Analysis Method uses computational matrix methods, and the accuracy of the results are dependent on the size and the number of elements. In this coursework, we will be using *ANSYS* to solve this truss by importing our designed *AutoCAD* sketch in *Fig 1.1* by referring to *Fig 3.1* using *ANSYS Workbench*.

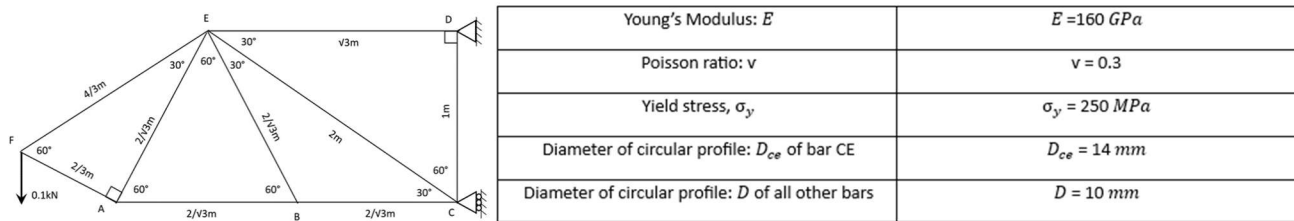


Fig 3.1: Given Truss Frame Properties (simulation restricted to Structural Steel due to Student License)

Starting the simulation by opening *Workbench*, then choosing the analysis system needed to do the simulation on, which is in this case **Static Structural Analysis**.

First Step (Sketching): - While it was possible to sketch the system on *DesignModeler*, we chose to sketch it using *AutoDesk AutoCAD* instead (See *Fig 1.1*), since it was found easier in both the process and ensuring accurate dimensions. The step of sketching was separated into two steps, First Step was sketching the truss alone by drawing lines with the required measurements and connect the lines to complete the shape of the truss. **The bar CE was split from the parent geometry to form a construction of one part from two bodies in order to assign different cross sections to the truss according to the data given in the coursework brief.**

NOTE: - Due to using the student version on *ANSYS*, which restricted the material to “Structural Steel”, we were unable to control the type of the material as mentioned in the coursework’s brief.

Second Step (Cross-Section): - After sketching the Truss, we are going to determine the type of Cross-Section needed and the size of it. In our case, we are using Circular Cross-Section with a 10mm Depth except for bar CE it has a depth of 14mm. by going to the concept menu → Cross-Section → select” Circular”, Then Choose the Depth for each part. **Third Step (Meshing and Applying Boundary Conditions):** - After Setting the Cross-Sections, we then moved to the Meshing section, which basically divides the elements of the Truss into small elements to get an accurate result over each element.

Applying the Boundary Conditions, as shown in *Fig 4.2*, as we have Fixed Support at Node D, Fixed Displacement at Node C which allows a free movement only in the Y-Axis, and a force with a magnitude of 0.1 KN or 100 N with a downwards direction at Node F.

Final Step (Solving and Simulating): - After Meshing and applying the Boundary Conditions, we are going to let *ANSYS* solve and do the simulation for the truss and get the deformation and the displacement of the elements and the nodes.

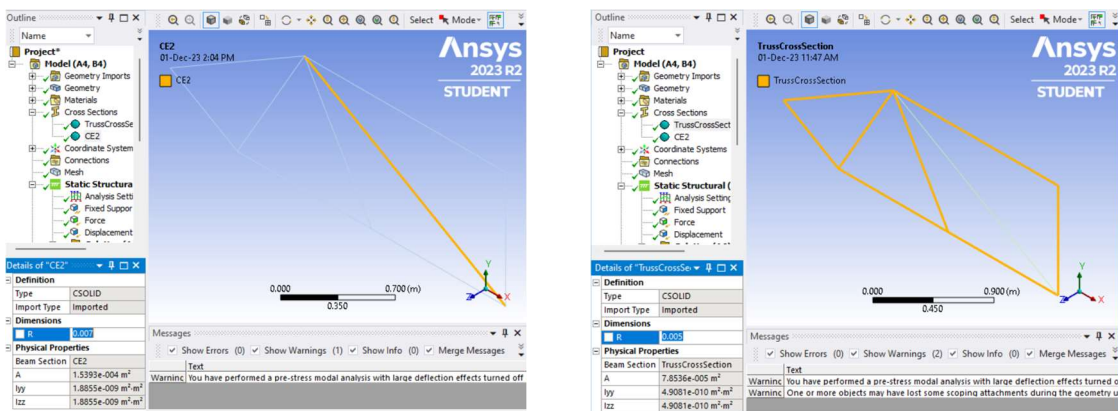


Fig 3.2 & 3.3: Isometric Views of the truss' geometry showing different cross sections

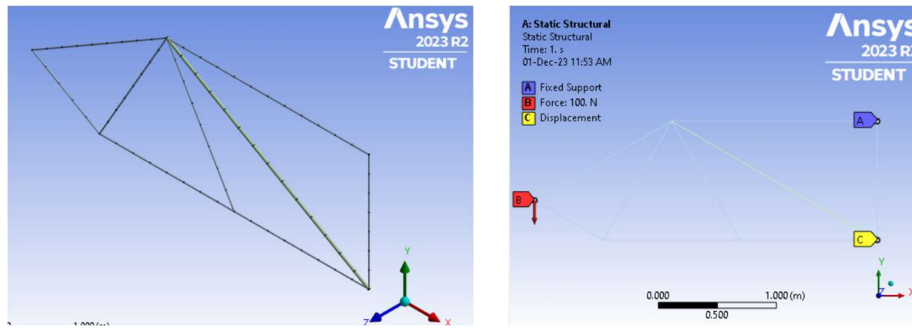


Fig 4.1 & 4.2: Isometric View (left) of the truss after Meshing & Front View (right) after Applying Boundary Conditions

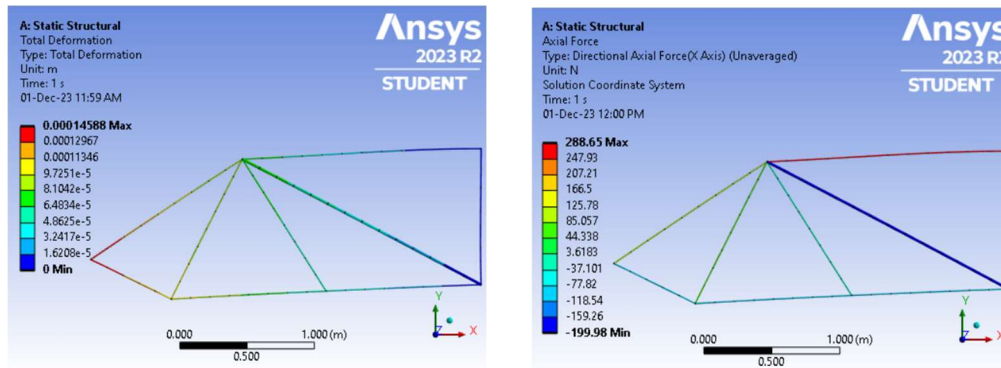


Fig 4.3 & 4.4: Front Views of the truss system showing Total Deformation/Nodal Displacement (left) and Directional Axial Force (right)

Results

Results	Minimum	Maximum	Units	Time (s)
Total Deformation	0.	1.4588e-004	m	1.
Nodal Euler XY Angle	0.	0.	°	1.
Elemental Euler XY Angle	-150.	150.	°	1.
Nodal Triads	0.	0.	°	1.
Axial Force	-199.98	288.65	N	1.
Directional Deformation	-3.1853e-005	1.6989e-005	m	1.

Probe: Reactions	X Magnitude	Y Magnitude	Z Magnitude	Total	Units	Time (s)
Force Reaction	288.67	100.	1.0917e-038	305.5	N	1.
Force Reaction 2	-288.67	0.	-8.9026e-022	288.67	N	1.

Fig 4.5: Result Summary (Extracted from ANSYS' Report)

The results visualized by ANSYS' simulation, while yielding different magnitudes, still confirms the obtained concepts from the Free Body Diagram's analysis, showing the highest tension at bar/element ED. The variance in magnitude was expected, due to the properties and the nature of Structural Steel, which makes it a rigid material and thus contributes to a higher magnitude in both tension and compression. However, the bar/element EC has shown higher compression magnitude, in contrast to the free body diagram's results. This might possibly be due to the widened cross-section of this bar/element, which may contribute to its increased compression compared to the predictions from the free body diagram analysis which did not take any cross sections into account.

The Mathematical Model

Using the concepts of the two previously mentioned methods, the process of modeling and analyzing the truss' system was nothing different, while being more like ANSYS since it's a process of numerical iterations. We've started by forming a Local Stiffness Matrix to each element before grouping them according to the nodes' sequence in a Global Stiffness Matrix to represent the whole truss.

Degrees of freedom

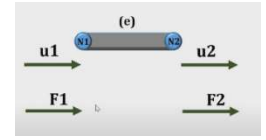
Degrees of freedom describe the number of independent motions or ways a system can move. For example, a particle in three-dimensional space has three translational degrees of freedom, while a rigid body may have both translational and rotational degrees of freedom. Understanding degrees of freedom is essential for making accurate statistical inferences, modelling physical systems, and solving engineering problems by accounting for the constraints and variations within a given system.

For a single node it can have three translational degrees of freedom in the X, Y, and Z direction and three translational degrees of freedom also around the X, Y and Z.

Adding up to a total of 6 degrees freedom for a singular node. However, in this case study it will be limited to two degrees of freedom at each node in the X and Y with two nodes for each element in the truss making it a total of 4 degrees of freedom per element.

Derivation of Global Stiffness Matrix

When analyzing a single static 1-D element containing two nodes considering only one degree of freedom per node. If the element is experiencing a force that is not in direct contact the forces at each node can be described as F_1 and F_2 consequently experiencing a displacement of u_1 and u_2



If the element is split and analyzed, an internal force is subjected within the element as a result of the external force.



Where from Hooke's law:

$$F_{int} = \sigma A$$

Where σ is the stress and A is the cross-sectional area of the bar.

$$\sigma = \epsilon E$$

ϵ is the Strain and E is the Young's Modulus representing the elasticity of the bar.

$$\epsilon = \frac{\Delta u}{L_{Original}}$$

Strain represents the ratio of deformation of the bar over the original length before deformation the longitudinal deformation is accounted as the displacements u_1 and u_2 .

$$F_{int} = \frac{EA}{L}(u_2 - u_1)$$

Because the system is in equilibrium:

$$F_1 + F_{int} = 0$$

Therefore:

$$F_1 = \frac{EA}{L}(u_1 - u_2), F_2 = \frac{EA}{L}(u_2 - u_1)$$

Finally, matrices can be made from this to solve for the system.

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \longrightarrow \quad \mathbf{F} = \mathbf{K}^e \mathbf{u}$$

This derivation has concluded that $\mathbf{F} = \mathbf{K}\mathbf{u}$ calculates the forces and displacements experienced by the elements, where \mathbf{K} is the stiffness matrix for an element which relates point forces, applied at a set of coordinates on the structure, to the displacements, at the same set of coordinates. Stiffness matrices of each element in the truss are combined in a single global matrix containing all the nodes of the truss which is used to calculate the needed missing forces and displacements.

It is important to emphasize that the matrix for the element above is for just the longitudinal (X) direction while in this case study another degree will be added which will increase it to 4 degrees per element making it a **4x4** stiffness matrix representing each node in each direction.

The elements in the truss are not all horizontal as naturally, there are some bars that will be at an angle and so by resolving the forces and displacements it will be derived that the stiffness matrix will be as follow:

Assembling Local Matrices

In this case study the truss in *Fig 3.1* is studied and a local stiffness matrix is obtained from each element, each having sperate material properties obtained from *Fig 3.1*.

Substituting the angles into the coordinate matrix for a 4x4 system as well as applying the properties of EA/L for each bar, the local matrices moving alphabetically through the nodes will be as in the following matrix:*

K1 (AB)				
Uax	Uay	Ubx	Uby	
10882796.19	0	-10882796.19	0	Uax
0	0	0	0	Uay
-10882796.19	0	10882796.19	0	Ubx
0	0	0	0	Uby

Table 6.1: A local stiffness matrix (for demonstration)

*The rest of the local stiffness matrices can be found in the Appendix section

Assembling The Global Stiffness Matrix

After obtaining all the local matrices for the 9 elements in the truss they are then combined in a single global matrix where the numbers are added for any overlapping between the local matrices, this represents all the degrees of freedom within the truss. Elements that share the same node such node E containing the most elements connected to it (5), all affecting the overall displacement experienced at node E. Therefore, the terms in the element stiffness matrices corresponding to node E should be summed for each degree of freedom. This process is done for any node sharing multiple elements.

To make this process easier the nodes are labelled in each local matrix for each of their respective degree of freedom. With this the global stiffness matrix should also be labelled in the same order of the nodes (alphabetically). This was done on Microsoft Excel to make the process of adding the overlapping terms easier using simple functions. The steps of achieving the global matrix via Excel are systematic. Place the cells of the local matrix in their respective places within the global matrix. This is done by allowing Excel to copy it in the form of a function that copies a cell.

Uay	Ubx	Uax	Uay
8 -3449708.156 -10882796.19	8	Uax	Uay
7 12874486.12 0	9	Ubx	Uby
9 0 24486291.43	10 10882796.19 0	=A3+A38+A45	-3449708.156
0 0 -4712388.982	11 0 0	-3449708.157	12874486.12
0 0 -4712388.982	12 10882796.19 0	-10882796.19	0
0 -A12 0			
0 0 0			

Fig 6.1, 6.2 & 6.3: Screenshots from Microsoft Excel while forming up the Global Stiffness Matrix, showing a cell as a summation of other cells

This in return simplifies the overlapping terms where it will be done by allowing Excel to add a cell with another as many times as it overlaps with other elements such as node E shown:

Uax	Uay	Ubx	Uby	Ucx	Ucy	Udx	Udy	Uex	Uey	Ufx	Ufy
27740662.18	-3449708.156	-10882796.19	0	0	0	0	0	-2720699.048	-4712388.982	-14137165.94	8162097.138
3449708.157	12874486.12	0	0	0	0	0	0	-4712388.982	-8162097.143	8162097.139	-4712388.98
-10882796.19	0	24486291.43	-4712388.982	-10882796.19	0	0	0	-2720699.048	-4712388.982	0	0
0	0	-4712388.982	8162097.143	0	0	0	0	-4712388.982	-8162097.143	0	0
0	0	-10882796.19	0	20119078.99	-5332570.13	0	0	-9236282.4	5332570.13	0	0
0	0	0	0	-5332570.13	15645131.41	0	0	-12566370.61	5332570.13	-3078760.8	0
0	0	0	0	0	0	0	0	-7255197.457	-7255197.457	0	0
0	0	0	0	0	0	0	0	-12566370.61	5332570.13	-3078760.8	0
0	0	0	0	0	0	0	0	0	0	0	0
-2720699.048	-4712388.982	-2720699.048	-4712388.982	-9236282.4	5332570.13	-7255197.457	0	29201461.42	-1251521.56	-7068583.471	-4081048.57
-4712388.982	-8162097.143	-4712388.982	-8162097.143	5332570.13	-3078760.8	0	0	-1251521.56	21759149.58	-4081048.57	-2356194.49
-14137165.94	8162097.138	0	0	0	0	0	0	-7068583.471	-4081048.57	2125750.41	-4081048.568
8162097.139	-4712388.98	0	0	0	0	0	0	-4081048.57	-2356194.49	-4081048.569	7068583.47

Fig 6.4 & 6.5: The Global Stiffness Matrix (left), and setting the matrix on MATLAB (right)

Solving the system's Global Stiffness Matrix using MATLAB

Using MATLAB in this coursework was clearly essential for the number of nodes in this matrix. We've formed a MATLAB code to implement a finite element analysis for the given truss' structure to obtain the values of the Nodal Forces & Nodal Displacement.

After introducing the Global Stiffness Matrix as shown in Fig 6.5 the matrix was defined in the code as "truss_matrix". The truss setup includes defining the nodes that do not experience displacement as "fixed_nodes", the node that has a pre-defined load "loaded_node", and the magnitude of the applied load "load_magnitude".

After having the Fixed Nodes identified, the corresponding rows and columns were removed from the stiffness matrix to account for the imposed constraints and to apply the boundary conditions, just like ANSYS.

```

22 % Extracting the number of nodes 25 % Formulate the global stiffness matrix K 34 % Applying forces (load at node 12)
23 num_nodes = size(truss_matrix, 1); 26 K = truss_matrix; 35 forces = zeros(num_nodes, 1);
36 forces(load_node) = load_magnitude;

```

Fig 7.1, 7.2 & 7.3: Defining the free nodes & applying the boundary conditions on MATLAB

A load is applied to a specific node, and the forces vector “**forces**” was initialized with zeros before being modified to include the applied load at the specified loaded node. The magnitude and direction of the applied load are given by the “**load_magnitude**” vector as shown in Fig 6.5.

The nodal displacements were then calculated by solving the system of equations using the reduced stiffness matrix and the applied forces.

```

22 % Extracting the number of nodes 25 % Formulate the global stiffness matrix K 28 % Applying boundary conditions
23 num_nodes = size(truss_matrix, 1); 26 K = truss_matrix; 29 free_nodes = setdiff(1:num_nodes, fixed_nodes);

```

Fig 7.4, 7.5 & 7.6: A snippet from the MATLAB code showing how the nodal displacements were calculated.

The complete displacement vector “**U**” is assembled by incorporating the calculated displacements for the free nodes. Nodal forces “**F**” are determined by multiplying the global stiffness matrix with the displacement vector. The code then concludes by displaying the number of nodes, the global stiffness matrix, nodal displacements, and nodal forces.

```

48 % Calculate nodal forces F = K * U
49 F = K * U;

```

Fig 7.7 A snippet from the MATLAB code showing how the nodal forces were calculated before printing the obtained results

Solving the matrix

In the context of resolving linear systems of equations, multiplication by the inverse of a matrix serves as a technique to isolate the variable on one side of the equation. Consider a linear system of equations represented in matrix form:

$$A \times X = B$$

Here, A denotes a square matrix, X is the column vector of unknowns, and B is another column vector. To ascertain X , we can perform multiplication on both sides of the equation by the inverse of matrix A , denoted as A^{-1} :

$$A^{-1} \times (A \times X) = A^{-1} \times B$$

The left side simplifies due to the property that $A^{-1} \times A$ equals the identity matrix I :

$$I \times X = A^{-1} \times B$$

Therefore

$$X = A^{-1} \times B$$

Hence, the multiplication by the inverse of matrix A allows for the isolation of the vector of unknowns X on one side of the equation. It is worth noting that not all matrices can have inverses. The square matrix A holds an inverse A^{-1} if and only if it is non-singular (having a non-zero determinant).

In the context of solving linear systems, matrices that are singular or nearly singular may introduce numerical instability issues during the computation of their inverses.

In our *MATLAB* code, the utilization of the backslash operator (\backslash) is employed for the direct solution of the linear system $K_r \times U_r = F_r$, for the explicit calculation of the inverse of “ K_r ”. This approach, according to *MATLAB*, is preferred due to its computational efficiency and numerical stability, particularly when dealing with large matrices. The backslash operator internally employs efficient numerical algorithms, such as LU decomposition, to determine the solution without explicitly forming the inverse of the matrix.

```

41 % Solve for nodal displacements using K_r * U_r = F_r
42 U_r = inv(K_r) * F_r;
43

```

⚠️ $\text{INV}(A)^*b$ can be slower and less accurate than $A \setminus b$. Consider using $A \setminus b$ for $\text{INV}(A)^*b$ or b/A for $b^* \text{INV}(A)$. [Details](#)

Fig 7.8: MATLAB recommending to compute $U_r = K_r \setminus F_r$ instead of $U_r = \text{inv}(K_r) * F_r$

Results

The obtained results from solving the truss’ system on *MATLAB* were almost identical to those obtained from *ANSYS*. Achieving identical results between *MATLAB* and *ANSYS* simulations is a significant validation of the accuracy and

reliability of the *MATLAB* implementation. This indicates that the implemented mathematical model accurately represents the structural behavior, and the numerical solution methods are consistent with those employed by *ANSYS*.

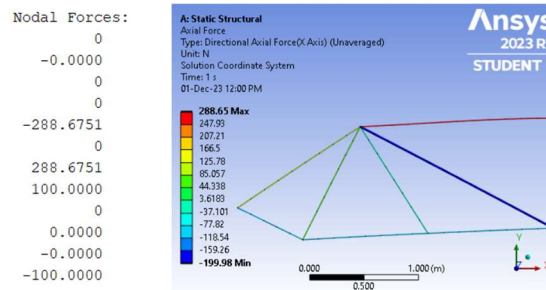


Fig 8.1 & 8.2: *MATLAB*'s Nodal Forces from U_{ax} till U_{fy} (left) & *ANSYS*' contours showing same values (right)

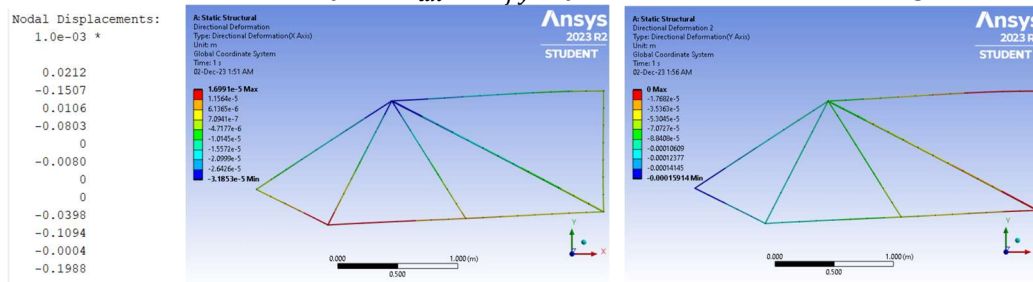


Fig 8.3, 8.4 & 8.5: *MATLAB* results of Nodal Displacement from U_{ax} till U_{fy} respectively (left) & *ANSYS*' contours showing almost-same values in X & Y Directions (middle & right)

Comparisons

In comparing the three methods used for solving the truss structure addressing the benefits and drawbacks of each:

Firstly, the **Manual Method**, static equilibrium calculations while they are reliable, they can be considered as not fully accurate as certain equations used possess a degree of inaccuracy due to oversimplification, it entails a theoretical framework with predefined equations to calculate forces. Furthermore, displacements at each node cannot be obtained from the static equilibrium equations without the addition of many more predefined equations and further decreasing the accuracy. Consequently, the manual method also requires a lot of hand calculation and manual work which when it is used regarding a much larger much more complicated structure it would not be very practical.

The **Numerical Software (FEA) method** utilizes sophisticated algorithms and meshing techniques to approximate solutions. In terms of accuracy, it is moderately accuracy and is dependent on the conditions set and the computing power. The mathematical model is theoretically robust but may deviate from real-world conditions. It is by far the fastest method and can be used for large scale projects with enough computing power to yield great results and visualization.

Finally, the **Mathematical Model** utilizes both manual work and software usage. The time it takes to complete may vary but is relatively moderate, the use of *MATLAB* then ensures extremely precise and accurate answers for both the forces and displacements at specific points to obtain in-depth results of each detail of the structure. It is the most accurate, but its usage on large scale structures might prove difficult due to its labor.

Conclusively, assessing the accuracy and reliability of each method in this study, considering factors such as accuracy, ease of application, and visualization and detail of results. In this case study both the numerical and mathematical model methods had the same results despite the mathematical method's advanced accuracy, this may be due to the simplicity of the structure allowing *ANSYS* to easily have high accuracy and thus validating the results. While on the other hand the manual method obtained less accurate answers compared with the other two and thus should be disregarded as the method of choice. Therefore, the choice of method depends on the specific requirements of the analysis, with the numerical FEA method often offering versatility and visualization and the mathematical model offering a higher degree of accuracy.

Summary

This report explores the behavior of a truss structure using manual methods, numerical simulations with *ANSYS*, and a *MATLAB* implementation. The main goal of this report was to validate and analyzing the output of each result and discussing the reasons that may have led to any differences in the outputs.

The manual method begins with a Free Body Diagram to determine forces in each truss member. The analysis involves applying Newton's 1st Law of Motion to establish equilibrium equations for each joint. Results from this method indicate tension and compression in different members, with a focus on bars ED and BE due to their high forces.

The *ANSYS* simulation involves using finite element analysis to model and analyze the truss structure. The process includes sketching, defining material properties, meshing, applying boundary conditions, and solving. The results from *ANSYS* confirm the results that were observed in the manual analysis, providing insights into the structural behavior under different loading conditions.

The *MATLAB* implementation involves creating a global stiffness matrix for the truss structure. The code applies boundary conditions, calculates nodal displacements and forces in separate arrays before displaying the results. Remarkably, the *MATLAB* results closely match those obtained from *ANSYS*, validating the accuracy and reliability of the *MATLAB* implementation for structural analysis.

Conclusions and Insights

- The manual analysis serves as a fundamental approach to understanding truss behavior, with specific attention to critical members.
- *ANSYS*, a numerical simulation tool, provides detailed insights into the structural response, validating trends observed in the manual method.
- *MATLAB*, used for solving our complex mathematical model, has successfully produced results consistent with *ANSYS*, ensuring how far our approach was realistic.

The report concludes by highlighting the significance of the visualized agreement between *MATLAB* and *ANSYS*' results, emphasizing the practical applications of each method, and showcasing a thorough understanding of structural analysis principles. The convergence of results from manual, numerical, and mathematical approaches strengthen the overall reliability of the structural analysis.

Future Recommendations

Areas of improvement and recommendations for future work can be to test these hypotheses with larger scale structures and forces to better see the difference between the numerical method (*ANSYS*) with the mathematical model method and conduct practical real-life experiments to further test these hypotheses. Furthermore, the usage of these methods to test how the structure would behave if specific beams were removed and how the structure can be improved to obtain best results with minimal material usage. Lastly, test the safety factors limits by pushing the working stress to the maximum to obtain a better understanding of its significance.

References

- Gavin, H. P. (2006). Mathematical properties of stiffness matrices. Lecture notes for Matrix Structural Analysis, CEE421L, *Department of Civil and Environmental Engineering, Duke University*, 2, 1-6.
- Bolton, W. C. (2006). Mechanical Science (3rd ed.). *Wiley-Blackwell*.
- El – Dakhakhni, W. M. (2010). Theory of Structures (14th ed.). *Dar Al-Maaref*.
- Hulse R., & Cain J. (2009). Structural Mechanics: Worked Examples (2009th ed.). *Red Globe Press*.
- Martin L., & Purkiss, J. A. (2007). Structural Design of Steelwork to EN 1993 and EN 1994, Third Edition (3rd ed.) *Butterworth-Heinemann*.
- Engineering Toolbox. (n.d.). Young's Modulus - Tensile and Yield Strength for Materials. *Engineering Toolbox*. Retrieved Month Day, Year, from https://www.engineeringtoolbox.com/young-modulus-d_417.html
- Byju's Exam Prep. (n.d.). Factor of Safety. *Byju's Exam Prep*. Retrieved Month Day, Year, from <https://byjusexamprep.com/gate-ce/factor-of-safety#toc-4>

Appendix

MATLAB Code & Results

```
% Given truss matrix
truss_matrix = [
27740662.18 -3449708.156 -10882796.19 0 0 0 0 0 -2720699.048 -4712388.982 -14137166.94
8162097.138;
-3449708.157 12874486.12 0 0 0 0 0 0 -4712388.982 -8162097.143 8162097.139 -4712388.98;
-10882796.19 0 24486291.43 -4712388.982 -10882796.19 0 0 0 -2720699.048 4712388.982 0 0;
0 0 -4712388.982 8162097.143 0 0 0 0 4712388.982 -8162097.143 0 0;
0 0 -10882796.19 0 20119078.59 -5332570.13 0 0 -9236282.4 5332570.13 0 0;
0 0 0 0 -5332570.13 15645131.41 0 -12566370.61 5332570.13 -3078760.8 0 0;
0 0 0 0 0 7255197.457 0 -7255197.457 0 0 0;
0 0 0 0 0 -12566370.61 0 12566370.61 0 0 0 0;
-2720699.048 -4712388.982 -2720699.048 4712388.982 -9236282.4 5332570.13 -7255197.457 0
29001461.42 -1251521.56 -7068583.471 -4081048.57;
-4712388.982 -8162097.143 4712388.982 -8162097.143 5332570.13 -3078760.8 0 0 -1251521.56
21759149.58 -4081048.57 -2356194.49;
-14137166.94 8162097.138 0 0 0 0 0 0 -7068583.471 -4081048.57 21205750.41 -4081048.568;
8162097.139 -4712388.98 0 0 0 0 0 0 -4081048.57 -2356194.49 -4081048.569 7068583.47
];

% Truss Setup
fixed_nodes = [5, 7, 8];
loaded_node = 12;
load_magnitude = -100;

% Extracting the number of nodes
num_nodes = size(truss_matrix, 1);

% Formulate the global stiffness matrix K
K = truss_matrix;

% Applying boundary conditions
free_nodes = setdiff(1:num_nodes, fixed_nodes);

% Removing rows and columns corresponding to fixed nodes
K_r = K(free_nodes, free_nodes);

% Applying forces (load at node 12)
forces = zeros(num_nodes, 1);
forces(loaded_node) = load_magnitude;

% Removing rows corresponding to fixed nodes
F_r = forces(free_nodes);

% Solve for nodal displacements using  $K_r * U_r = F_r$ 
U_r = K_r \ F_r;

% Assemble the complete displacement vector U
U = zeros(num_nodes, 1);
U(free_nodes) = U_r;

% Calculate nodal forces  $F = K * U$ 
F = K * U;

% Display results
disp('Number of Nodes:')
disp(num_nodes)
disp('Global Stiffness Matrix:')
disp(K)
disp('Nodal Displacements:')
disp(U)
disp('Nodal Forces:')
disp(F)
```

```
>> truss
```

```
Number of Nodes:
```

```
12
```

```
Global Stiffness Matrix:
```

```
1.0e+07 *
```

```

    2.7741   -0.3450   -1.0883         0         0         0         0         0
-0.2721   -0.4712   -1.4137    0.8162         0         0         0         0
   -0.3450    1.2874         0         0         0         0         0         0
-0.4712   -0.8162    0.8162   -0.4712         0         0         0         0
   -1.0883         0    2.4486   -0.4712   -1.0883         0         0         0
-0.2721    0.4712         0         0         0         0         0         0
         0         0   -0.4712    0.8162         0         0         0         0
    0.4712   -0.8162         0         0         0         0         0         0
         0         0   -1.0883         0    2.0119   -0.5333         0         0
-0.9236    0.5333         0         0         0         0         0         0
         0         0         0         0         0   -0.5333    1.5645         0
    0.5333   -0.3079         0         0         0         0         0   -1.2566
         0         0         0         0         0         0         0    0.7255
-0.7255         0         0         0         0         0         0         0
         0         0         0         0         0         0   -1.2566         0
         0         0         0         0         0         0         0    1.2566
-0.2721   -0.4712   -0.2721    0.4712   -0.9236    0.5333   -0.7255         0
2.9001   -0.1252   -0.7069   -0.4081    0.5333   -0.3079         0         0
   -0.4712   -0.8162    0.4712   -0.8162         0         0         0         0
-0.1252    2.1759   -0.4081   -0.2356         0         0         0         0
   -1.4137    0.8162         0         0         0         0         0         0
-0.7069   -0.4081    2.1206   -0.4081         0         0         0         0
         0.8162   -0.4712         0         0         0         0         0         0
-0.4081   -0.2356   -0.4081    0.7069         0         0         0         0

```

```
Nodal Displacements:
```

```
1.0e-03 *
```

```

    0.0212
   -0.1507
    0.0106
   -0.0803
         0
   -0.0080
         0
         0
   -0.0398
   -0.1094
   -0.0004
   -0.1988

```

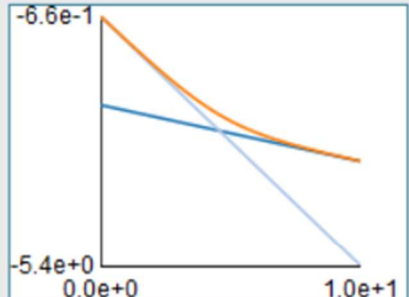
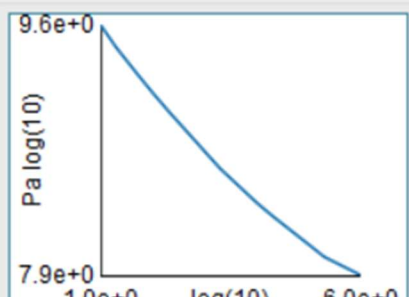
```
Nodal Forces:
```

```

0
-0.0000
0
0
-288.6751
0
288.6751
100.0000
0
0.0000
-0.0000
-100.0000
>>

```

Material Properties of “Structural Steel” used in ANSYS’ simulation

Structural Steel	
▼ Isotropic Elasticity	
Derive from	Young's Modulus and Poisson's Ratio
Young's Modulus	2e+11 Pa
Poisson's Ratio	0.3
Bulk Modulus	1.6667e+11 Pa
Shear Modulus	7.6923e+10 Pa
Isotropic Secant Coefficient of Thermal Expansion	1.2e-05 1/°C
Compressive Ultimate Strength	0 Pa
Compressive Yield Strength	2.5e+08 Pa
Strain-Life Parameters	
S-N Curve	
Tensile Ultimate Strength	4.6e+08 Pa
Tensile Yield Strength	2.5e+08 Pa

Free Body Diagram Calculations of all elements during The Manual Method

Joint "F":

$$\sum F_x = \text{zero}$$

$$F_{fe} \cos(30) + F_{fa} \cos(30) = 0$$

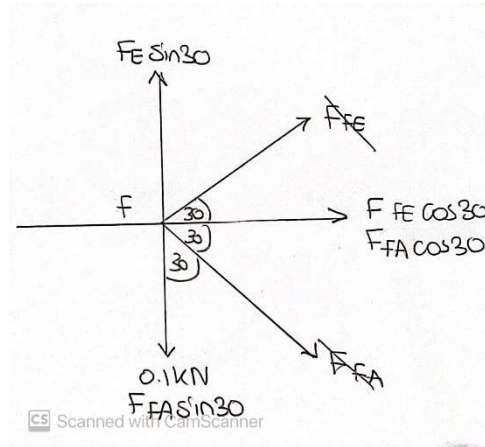
$$\sum F_y = \text{zero}$$

$$F_{fe} \sin(30) = 0.1 + F_{fa} \sin(30)$$

$$F_{fe} \sin(30) - F_{fa} \sin(30) = 0.1$$

$$\therefore F_{fe} = 0.1 \text{ kN (Tension)}$$

$$, F_{fa} = -0.1 \text{ kN (Compression)}$$



Joint "A":

$$\sum F_y = \text{zero}$$

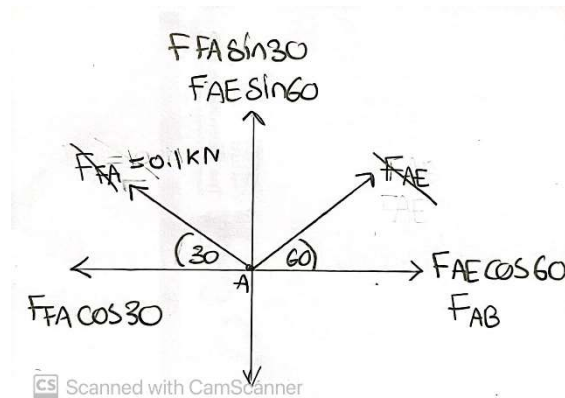
$$F_{ae} \sin(60) + 0.1 \sin(30) = 0$$

$$\therefore F_{ae} = -0.0577 \text{ kN (Compression)}$$

$$\sum F_x = \text{zero}$$

$$F_{ab} + -0.0577 \cos(60) - F_{fa} \cos(30) = 0$$

$$\therefore F_{ab} = 0.115 \text{ kN (Tension)}$$



Joint "B":

$$\sum F_y = \text{zero}$$

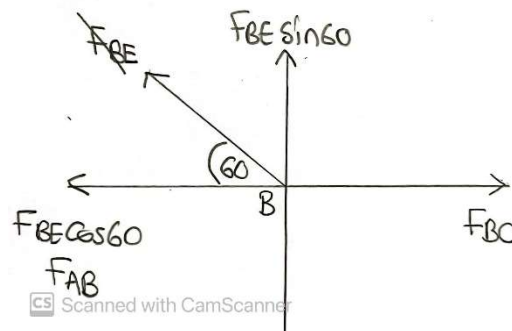
$$F_{be} \sin(60) = 0$$

$$\therefore F_{be} = 0 \text{ kN}$$

$$\sum F_x = \text{zero}$$

$$F_{bc} = 0.115 + F_{be} \cos(60)$$

$$\therefore F_{bc} = 0.115 \text{ kN (Tension)}$$



Joint "C":

A roller support is placed at point C having a reaction force only in the Y-direction.

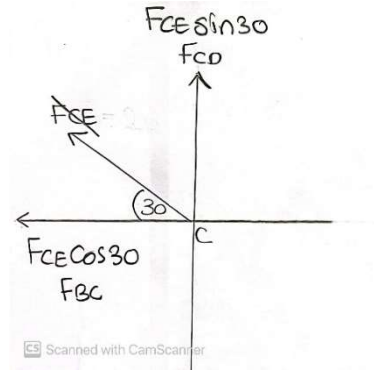
$$\sum F_y = \text{zero}$$

$$F_{ce} \sin(30) + F_{CD} = 0$$

$$\therefore F_{CD} = 0.099 \text{ kN (Tension)}$$

$$\sum F_x = \text{zero}$$

$$\therefore F_{ce} = -0.198 \text{ kN (Compression)}$$



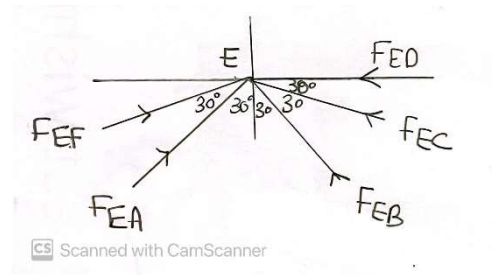
Joint "E":

A fixed support is placed at node D having two perpendicular reactions on the X and Y directions.

$$\sum F_y = \text{zero}$$

$$F_{ce} \sin(30) + 0 \sin(60) + 0.1 \sin(30) + 0.057 \sin(60)$$

$$F_{ce} = -0.198 \text{ kN (Compression)}$$



Given that:

$$F_{fe} = 0.1 \text{ kN} \quad F_{be} = 0 \text{ kN},$$

$$F_{ae} = 0.057 \text{ kN} \quad F_{ce} = -0.198 \text{ kN}$$

F_{ed} becomes the unknown force accordingly.

$$\therefore 0.1 \cos(30) + 0.057 \cos(60) = -0.198 \cos(30) + 0 \cos(60) + F_{ed}$$

$$\therefore F_{ed} = 0.286 \text{ kN (Tension)}$$

Local Stiffness Calculations of all elements during The Manual Method

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01} = 1.256637061 \times 10^9 \text{ (N/m)}$$
$$\sigma_{FE} = \frac{F}{A} = \frac{0.1 \times 10^3}{\pi (5 \times 10^{-3})^2} = 1.273239545 \times 10^6 \text{ Pa}$$

Bar AF:

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01} = 1.256637061 \times 10^9 \text{ (N/m)}$$
$$\sigma_{AF} = \frac{F}{A} = \frac{0.1 \times 10^3}{\pi (5 \times 10^{-3})^2} = 1.273239545 \times 10^6 \text{ Pa}$$

Bar AE:

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01} = 1.256637061 \times 10^9 \text{ (N/m)}$$
$$\sigma_{AE} = \frac{F}{A} = \frac{0.057 \times 10^3}{\pi (5 \times 10^{-3})^2} = 725.74654 \times 10^3 \text{ Pa}$$

Bar BA:

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01} = 1.256637061 \times 10^9 \text{ (N/m)}$$
$$\sigma_{BA} = \frac{F}{A} = \frac{0.115 \times 10^3}{\pi (5 \times 10^{-3})^2} = 1.464225476 \times 10^6 \text{ Pa}$$

Bar EB:

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01} = 1.256637061 \times 10^9 \text{ (N/m)}$$
$$\sigma_{EB} = \frac{F}{A} = \frac{0}{\pi (5 \times 10^{-3})^2} = 0 \text{ Pa}$$

Bar DE:

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (5 \times 10^{-3})^2}{0.01} = 1.256637061 \times 10^9 \text{ (N/m)}$$
$$\sigma_{DE} = \frac{F}{A} = \frac{0.286 \times 10^3}{\pi (5 \times 10^{-3})^2} = 3.641464098 \times 10^6 \text{ Pa}$$

Bar EC:

$$\delta = \frac{EA}{L}, \frac{160 \times 10^9 \times \pi (7 \times 10^{-3})^2}{0.014} = 1.759291886 \times 10^9 \text{ (N/m)}$$
$$\sigma_{EC} = \frac{F}{A} = \frac{0.198 \times 10^3}{\pi (7 \times 10^{-3})^2} = 2.521014299 \times 10^6 \text{ Pa}$$

We conclude from the stiffness value on bar EC that it has the highest Compression.

Local Stiffness Matrices

K1 (AB)

	Uax	Uay	Ubx	Uby	
	10882796.19	0	-10882796.19	0	Uax
	0	0	0	0	Uay
	-10882796.19	0	10882796.19	0	Ubx
	0	0	0	0	Uby

K2 (BC)

	Ubx	Uby	Ucx	Ucy	
	10882796.19	0	-10882796.19	0	Ubx
	0	0	0	0	Uby
	-10882796.19	0	10882796.19	0	Ucx
	0	0	0	0	Ucy

K3 (CD)

	Ucx	Ucy	Udx	Udy	
	0	0	0	0	Ucx
	0	12566370.61	0	-12566370.61	Ucy
	0	0	0	0	Udx
	0	-12566370.61	0	12566370.61	Udy

K4 (ED)

	Udx	Udy	Uex	Uey	
	7255197.457	0	-7255197.457	0	Udx
	0	0	0	0	Udy
	-7255197.457	0	7255197.457	0	Uex
	0	0	0	0	Uey

K5 (FE)

Uex	Uey	Ufx	Ufy	
7068583.471	4081048.57	-7068583.471	-4081048.57	Uex
4081048.57	2356194.49	-4081048.57	-2356194.49	Uey
-7068583.471	-4081048.57	7068583.471	4081048.57	Ufx
-4081048.57	-2356194.49	4081048.57	2356194.49	Ufy

K6 (AF)

Uax	Uay	Ufx	Ufy	
14137166.94	-8162097.138	-14137166.94	8162097.138	Uax
-8162097.139	4712388.98	8162097.139	-4712388.98	Uay
-14137166.94	8162097.138	14137166.94	-8162097.138	Ufx
8162097.139	-4712388.98	-8162097.139	4712388.98	Ufy

K7 (AE)

Uax	Uay	Uex	Uey	
2720699.048	4712388.982	-2720699.048	-4712388.982	Uax
4712388.982	8162097.143	-4712388.982	-8162097.143	Uay
-2720699.048	-4712388.982	2720699.048	4712388.982	Uex
-4712388.982	-8162097.143	4712388.982	8162097.143	Uey

K8 (BE)

Ubx	Uby	Uex	Uey	
2720699.048	-4712388.982	-2720699.048	4712388.982	Ubx
-4712388.982	8162097.143	4712388.982	-8162097.143	Uby
-2720699.048	4712388.982	2720699.048	-4712388.982	Uex
4712388.982	-8162097.143	-4712388.982	8162097.143	Uey

K9 (CE)

Ucx	Ucy	Uex	Uey	
9236282.4	-5332570.13	-9236282.4	5332570.13	Ucx
-5332570.13	3078760.8	5332570.13	-3078760.8	Ucy
-9236282.4	5332570.13	9236282.4	-5332570.13	Uex
5332570.13	-3078760.8	-5332570.13	3078760.8	Uey

The Global Stiffness Matrix

Uax	Uay	Ubx	Uby	Ucx	Ucy	Udx	Udy	Uex	Uey	Ufx	Ufy	
27740662.18	-3449708.156	-10882796.19	0	0	0	0	0	-2720699.048	-4712388.982	-14137166.94	8162097.138	Uax
-3449708.157	12874486.12	0	0	0	0	0	0	-4712388.982	-8162097.143	8162097.139	-4712388.98	Uay
-10882796.19	0	24486291.43	-4712388.982	-10882796.19	0	0	0	-2720699.048	4712388.982	0	0	Ubx
0	0	-4712388.982	8162097.143	0	0	0	0	4712388.982	-8162097.143	0	0	Uby
0	0	-10882796.19	0	20119078.59	-5332570.13	0	0	-9236282.4	5332570.13	0	0	Ucx
0	0	0	0	-5332570.13	15645131.41	0	-12566370.61	5332570.13	-3078760.8	0	0	Ucy
0	0	0	0	0	0	7255197.457	0	-7255197.457	0	0	0	Udx
0	0	0	0	0	-12566370.61	0	12566370.61	0	0	0	0	Udy
-2720699.048	-4712388.982	-2720699.048	4712388.982	-9236282.4	5332570.13	-7255197.457	0	29001461.42	-1251521.56	-7068583.471	-4081048.57	Uex
-4712388.982	-8162097.143	4712388.982	-8162097.143	5332570.13	-3078760.8	0	0	-1251521.56	21759149.58	-4081048.57	-2356194.49	Uey
-14137166.94	8162097.138	0	0	0	0	0	0	-7068583.471	-4081048.57	21205750.41	-4081048.568	Ufx
8162097.139	-4712388.98	0	0	0	0	0	0	-4081048.57	-2356194.49	-4081048.569	7068583.47	Ufy

Fig 8.4: The Global Stiffness Matrix