3D Finite Element Method Rigid/Pin Joint Python Program

Deepak Singh

Contents

1 Introduction

This three script Python program allows the computation of the displacements and rotations of beam elements under axial, shear, and torsional (moment) loads in three dimensions (3D). The finite element method (FEM) allows for numerical computation of the displacements instead of analytical process of computing displacements and reactions in beams/rods which is usually studied in early engineering courses for instance *Strength of materials*. The program is built from the assumptions from Euler Bernoulli Theory. Further information for Euler Bernoulli Theory can be found in *Beam Structures: Classical and Advanced Theories.* [1].

Depending on the initial and boundary conditions, further discussed in the Pre-Processor section, either rigid or pin joints can be implemented into the program. Rigid joints allow for the transmission of both axial, shear and torsion loads, while pin joints do not transmit torsional loads. The Main-Process script computes displacements by generating the global stiffness matrix of the structure. The computation of the nodal displacements and reaction forces is also computed using the global stiffness matrix. Finally, the Post-Processor script displays the beam structure with the displaced beam structure.

1.1 Required Libraries

The program is written in Python 3. The various libraries required to run the program are:

- NumPy
- Pandas
- JSON
- Matplotlib

2 Pre-Processor

2.1 .XLS Document

A XLS spreadsheet document with two sheets can be used to type in the nodal and element properties. The document will be opened in the pre-processor script and the contents are organised into JSON file format for the main-processor script. An empty XLS document can be seen in the folder named *blank.xls*. The first sheet is for the nodal properties and the second sheet is for the element/ bar properties.

2.2 Script Information

The script sequence can be described as:

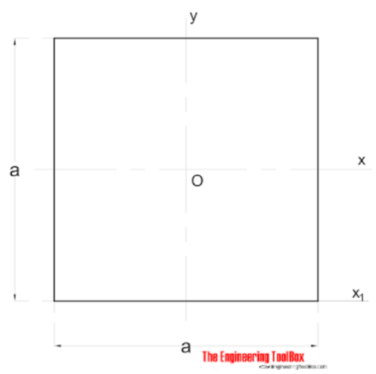
1. Import Libraries
2. Importing and opening XLS file
3. Node and Element Functions
4. Using previous functions to organise data into JSON format
5. Outputting JSON file for Main-Processor
6. Displays structural problem described in XLS file

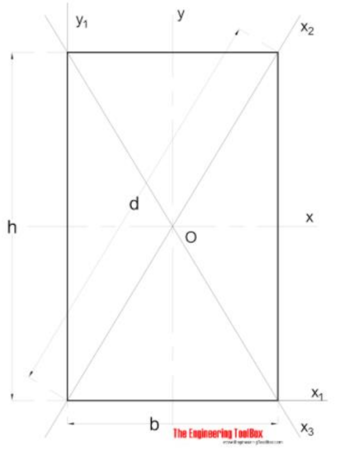
In the fifth described part of the pre-processor scrip the name of the output file is default set up as *result.json* but can be changed to create multiple of output files. Other files can be implemented into the program if required. For instance, CSV file type can instead be used instead.

2.3 Other Useful information

Various mechanical properties of the beam structures can be found from various resources externally. Below is the moment of inertia (I_y,I_z) and the torsion constant of some various cross sections which can be found in beam structures:

Table 1 Different Cross Sections Mechanical Properties [2][3]

Cross section Type	Moment of Inertia	Torsion Constant
 Square Cross Section	$I_y = I_z = \frac{a^4}{12}$	$J = \frac{9\ a^4}{4}$



Rectangular Cross Section

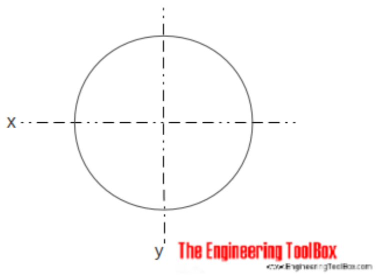
$$I_y = \frac{bh^3}{12}$$

$$I_z = \frac{b^3 h}{12}$$

$$J = \beta b^3 h \; (b \leq h)$$

$$\beta = \frac{1}{3} - \cdots$$

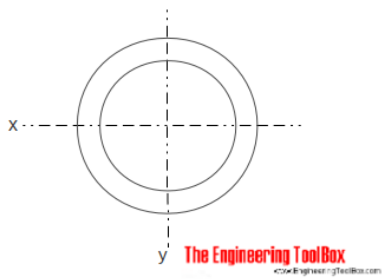
$$\ldots 0.21 \frac{b}{h}\left[1 - \frac{1}{12}\left(\frac{b}{h}\right)^4\right]$$



Circular Cross Section

$$I_y = I_z = \frac{\pi r^4}{4}$$

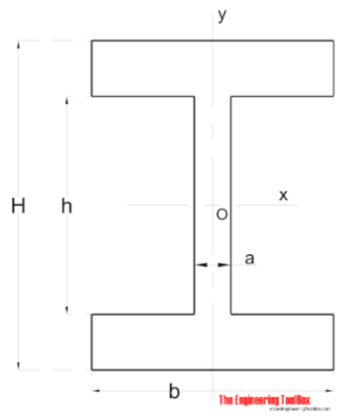
$$J = \frac{1}{2}\pi r^4$$



Hollow cylindrical cross ection

$$I_y = I_z = \frac{\pi\left(d_o^4 - d_i^4\right)}{64}$$

$$J = 2\pi r^3 (r_o - r_i)$$

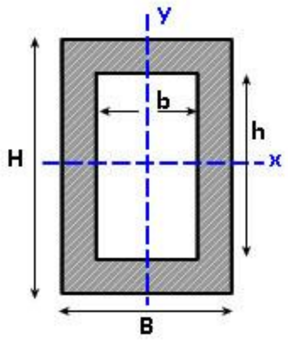


Symmetrical I-beam Cross Section

$$I_y = \frac{ah^3}{12} + \frac{b}{12}(H^3 - h^3)$$

$$J = \frac{1}{3}\left[2b\left(\frac{H-h}{2}\right)^3 + Ha^3\right]$$

$$I_z = \frac{a^3 h}{12} + \frac{b^3}{12}(H - h)$$



$$I_y = \frac{BH^3}{12} - \frac{bh^3}{12} \qquad J = \frac{2B^2H^2}{2\left(\frac{B}{B-b} + \frac{H}{H-h}\right)}$$

$$I_z = \frac{HB^3}{12} - \frac{hB^3}{12}$$

Hollow Rectangle Cross section

Websites for online mechanical properties of shapes include:

- https://www.engineeringtoolbox.com/area-moment-inertia-d_1328.html
- https://amesweb.info/section/second-moment-of-area-calculator.aspx
- https://skyciv.com/free-moment-of-inertia-calculator/

For further information on more complex cross sections can be found in various literature including:

- *Mechanics for Engineers: Statics [4]*
- *Strength of Materials & Structures (With an introduction to Finite Element Methods) [5]*

2.4 Initial Conditions and Boundary conditions

Rigid and Pin joints can be differentiated from the boundary and initial conditions described in the XLS document. As previously discussed, rigid joints can transmit both axial, shear and torsional loads. Therefore, rigid joint can displace and rotate. The rigid joint which allows for Newton's third law to be achieved, e.g. the rigid joint which connects to wall/ ground, will not displace nor rotate. A pin joint will not be able to transmit torsional loads, however, this still means pin joints displace and rotate. A pin joint end will differentiate with a rigid end due to pin joints still rotating, while they do not displace like the rigid end. The boundary and initial conditions for rigid and pin joints can be seen in Table 2.

Table 2 Initial Conditions and Boundary Conditions

Type	F_x, F_y, F_z	M_x, M_y, M_z	$\Delta x, \Delta y, \Delta z$	$\Delta\theta_x, \Delta\theta_y, \Delta\theta_z$
Rigid End	Reaction ('Free')	Reaction ('Free')	B.C **	B.C**
Rigid Middle	I.C	I.C	Reaction ('Free')	Reaction ('Free')
Pin End	Reaction('Free')	0	B.C**	Reaction ('Free')
Pin Middle	I.C	0	Reaction ('Free')	Reaction ('Free')

*** Boundary Conditions default is 0. Values with real numbers are for problems where the beam structures dimensions are smaller or larger than required. Therefore, the structure is forced into place causing the structure to be under tension/ compression as the structure is forced to stretched/ compressed.*

3 Main-Processor

3.1 Script Information

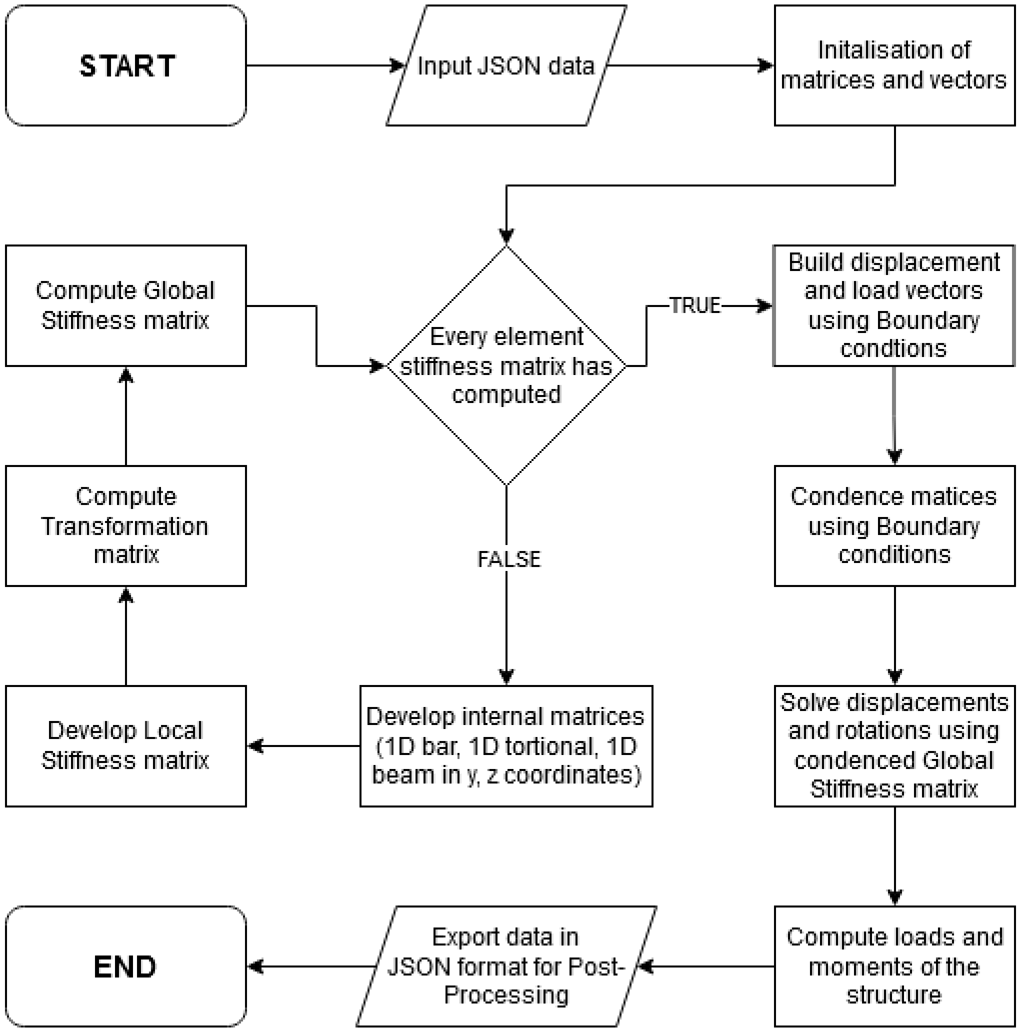


Figure 1 Main-Process script flow chart

The basic algorithm of the main processor can be seen in Figure 1.

3.2 FEM Beam Elements

The FEM beam stiffness matrices are developed from 3 different local stiffness matrices. The superposition scheme can be seen in Figure 2.

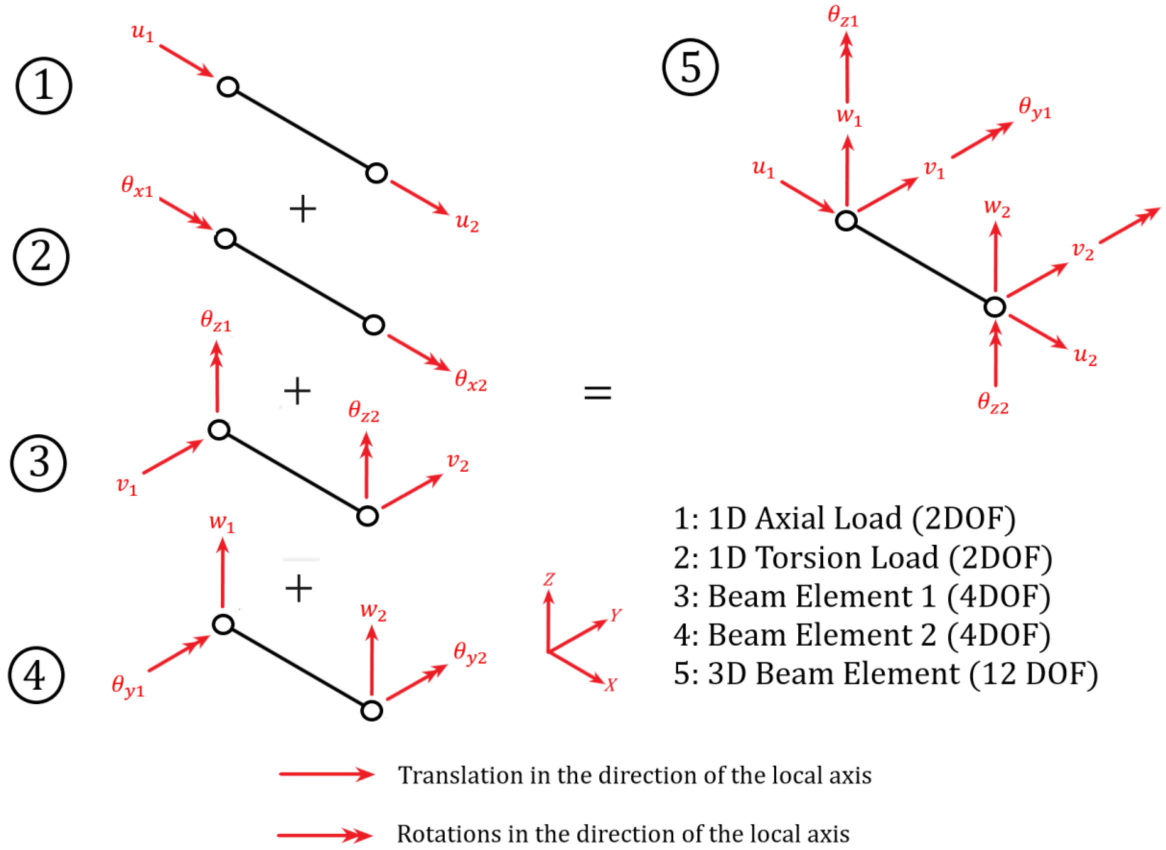


Figure 2 Superposition scheme for a 3D beam

3.3 Main Processor Results and Outputs

As shown in Figure 1 the Main processor will solve the missing displacements and rotations. The global stiffness matrix [A] developed is reduced allowing for missing displacements [U] to be solved. The inverse of the reduced stiffness matrix is multiplied with the forces and moments [F] applied onto the structure:

$$[\mathbf{A}]^{-1}[\mathbf{F}] = [\mathbf{U}]$$

After all the displacements have been found, the reaction forces and moments [P] on the structure can be obtained. This requires the full global stiffness matrix [K] and displacements found previously.

$$[\mathbf{K}][\mathbf{U}] = [\mathbf{P}]$$

The data developed from this file is outputted via Json format; the data is used for the post-processor file.

4 Post-Processor

The post-processor script allows for the deformed structure to be viewed interactively. Due to the limitations of matplotlib tools, the 3D plots produce is not to scale, and the plots are an exaggerated view of deformed truss structure. Therefore, all the axes have been automatically turned off. The displacements can be manually scaled to allow for the deformed plot to be adjusted. Furthermore, 3D quivers have been used to display the applied loads and reaction forces. However, it is best if the loads scale is set to a very small value. This will therefore makes the loads plot redundant. This is due to the difference in the size of the structure and quivers. Depending on the deformed structure, it can be very hard to display visually appealing quivers with the correct arrowhead sizes. This can be seen in Figure 3.

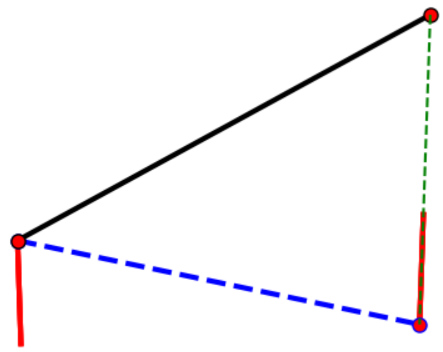


Figure 3 Deformed Cantilever Structure with Load applied at the end.

5 Aircraft Wing Truss Structure Case Study

The document named "EG-353-FinalPaper" is a research paper produced during my final year of my undergraduate degree. The three-script program was developed to investigate the structure of an aircraft wing under aerodynamic loads. The version of the program differs, as the pre-processor uses the aerodynamic data for the applied loads and moments. The research paper also validates the program in section 4. There are three different validation cases investigated and compared with the analytical theoretical results.

References

[1] Carrera E, Giunta G, Petrolo M. Beam Structures : Classical and Advanced Theories. Hoboken: John Wiley & Sons, Incorporated; 2011, pp9-22

[2] Area Moment of Inertia - Typical Cross Sections I [Internet]. Engineeringtoolbox.com. 2021 [cited 12 August 2021]. Available from: https://www.engineeringtoolbox.com/area-moment-inertia-d_1328.html

[3] Second Moment of Area Calculator [Internet]. Amesweb.info. 2021 [cited 12 August 2021]. Available from: https://amesweb.info/section/second-moment-of-area-calculator.aspx

[4] Beer, F., Johnston, E. and Flori, R., 2008. *Mechanics for engineers: Statics*. New York: McGraw-Hill.

[5] Case, J., Chilver, L. and T.F. Ross, C., 1993. Strength of Materials & Structures. 3rd ed. London: Arnold