Computer analysis of truss structures

Garrett Baker
Mentored by Dr. Richard Becker

Introduction

Truss structures are one of the most versatile and adaptable forms of engineered structures. Consisting of a set of beams connected to each other by joints (nodes), they have been used in a large variety of domains of structures from bridges and buildings to boats and roofs. Thus, the ability to simulate how these structures will react and deform to the loads that will be placed on them is useful for evaluating the safety and reliability of various construction projects. Similarly, an automated method for optimizing a truss structure to minimize cost or maximize stability would save countless person-hours during the development of these construction projects.

The goal of this project was to develop a program to analyze the mechanics of truss structures. Given a truss structure and a load, the program would calculate beam stress and the amount that the nodes move.

Materials and Methods

The programming language Python 3.7.17, using the libraries numpy, math, and re (regular expressions), was used for the codebase of this project. The Emacs IDE was used for the code development, and several Jupyter notebooks were used for verification. The simulation program took, as input, truss files of a customized file format, and output *.xmf files for viewing in the program Visit. Inkscape and LibreOffice Calc were used to encode the Burr truss known as Jericho bridge.

Over the summer, the foundational math for structural analysis was learned. This consisted of researching and attempting to prove various results using geometry, trigonometry, and some basic physics. This introduced the concept of beam matrices—matrices which describe the loads (or forces) put on a particular beam as a function of the displacement of each of the beam's nodes and the beam's structural properties. Once the school-year started and the proposal for the project was completed, the more complex portions of the math were learned. This consisted of learning how to combine multiple beam matrices in order to reflect the structural connections between each of the beams, creating a structure matrix equation, taking as input the nodal displacements and outputting the force vector on each node (Holzer, 1985).

Once this math was learned, a Python program consisting of an object class describing the properties of beams, an object class describing the properties of an entire structure, made up of beams, and a method allowing the input of a file and conversion of the data stored in that file into a structure object was written. Once a file was inputted, this structure object would generate each of the beam matrices and then use those beam matrices to generate the structure matrix. This matrix was constrained with some known forces and displacements, then the resulting equation was solved for the unknown displacements.

Materials and Methods (cont.)

After this part of the program was made, the basic equations relating the stress on each beam to the displacement of each node were learned and then coded into the program as well.

Jericho Bridge was then selected as the bridge the program's optimization capabilities would be tested on. A schematic of this bridge was found in the Library of Congress (Figure 1) (Forsyth, 1933), and the locations of each node, as well as the beam connections were recorded onto a *.csv file.

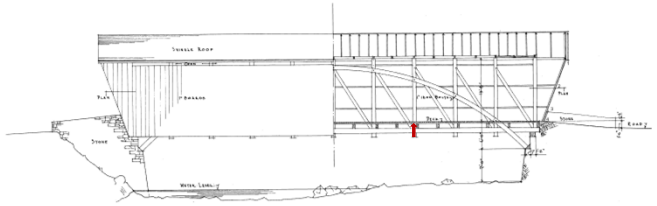

Figure 1: The original schematic of Jericho Bridge. Used for the recording of the node locations and beam connections. The red arrow indicates the node where the 2,000 N force was placed.

The data was fed into the created Python program, as well as a professional-grade program. A 2,000 N force was virtually placed on the node indicated by a red arrow in Figure 1, and each program calculated the resulting node displacement and beam stress values. The two data sets were then compared and visualized using Visit.

Results

The Visit visualization (Childs et al., 2012) of the outputted data of the created program can be viewed in Figure 2.

The resulting percent error of the *x* and *y* displacement values were 7.3×10^{-7}% and 6.9×10^{-7}% respectively, essentially zero. Both were less than the goal of 0.1% error, constituting a success of accurately calculating the displacement vectors of the structure. The resulting percent error of the beam stress values was 190%, significantly above the goal of 0.1% error, constituting a failure to accurately calculate the stress values of the structure. Plausible reasons why this error is so high are discussed in the conclusions section.

Results (cont.)

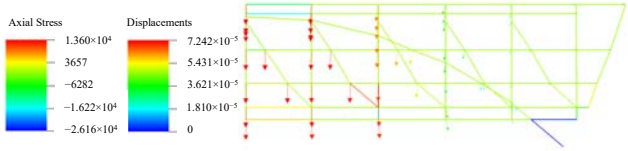

Figure 2: Resulting beam-stresses (colored lines) and node displacements (arrows) from the created program visualized in Visit. Notice how inconsistent and non-continuous the stress coloring on the beams are. This is an effect of the errors in the calculation.

Conclusion

The purpose of this project was to create a program which could accurately simulate the effects of a truss structure placed under loads within an error of 0.1%. This purpose was partially successful, as the final product could simulate the movement of the nodes with an error of 7.3×10^{-7}% in the *x*-direction, and 6.9×10^{-7}% in the *y*-direction but couldn't simulate the stress on the beams.

The most likely reason the stress calculations were unsuccessful was because the force vectors used in this calculation were those which were computed using the beam matrix and displacements vector. This makes most all forces at each node sum to zero, since the system is in equilibrium. Instead, forces should've been recalculated at each beam, based on the change in length of the beam as a result of the displacement vectors.

Future research could attempt to fix these problems to get an accurate simulation of the beam stresses on the truss. Alternatively, the product in its current form may be used to assist architects in constructing stable building, possibly by implementing a form of machine learning or iterative optimization algorithm which would attempt to minimize the average nodal displacement across the truss.

References

Childs H., Brugger E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K., Miller, M., Harrison, C., Weber, G., Krishnan, H., Fogal, T., Sanderson, A., Garth, C., Bethel, E.W., Camp, D., Rubel, O., Durant, M., Favre, J., Naviratil, P. (2012). VisIt: An end-user tool for visualizing and analyzing very large data. https://visit-dav.github.io/visit-website/

Forsyth, T., Pickering, E.H., Scarff, J.H., Ewald, L. (1933). Covered bridge, Jericho road spanning little gunpowder falls, Jerusalem, Baltimore County, MD. *Historic American Buildings Survey (Library of Congress).* https://www.loc.gov/resource/hhh.md0655.sheet/?sp=2

Holzer, S.M. (1985). Matrix displacement method: Plane structures. In *Computer analysis of structures* (pp.108–159). Elsevier.