Interactive Structural Analysis of Drawn Structures

Allan Pendergrast

Abstract

Many architects and engineers use best guesses, approximations, and other simplifying assumptions to design structures. While understandable due to the complexity of analyzing an entire complex structure, this frequently results in over-engineering or overlooking critical points. This research presents a solution to this problem by making a tool available to architects and engineers that shows the critical parts of the structure that are under the highest stress. This tool is presented as a plug-in to the popular design tool Google Sketchup. The current implementation suffers from a number of limitations, but further work should solve these issues.

1 Related Work

This implementation is built off of a method of solving systems of equations using matrices that was developed in the 18th and 19th centuries. It is very simple, but limited to systems where the number of equations is exactly equal to the number of variables. This method is taught in freshman-level statics classes for small 2D trusses, but is very difficult to work out by hand for larger structures. Papers on related topics of solving more complex series of equations will be used in further research but have not been referenced for this paper.

2 Algorithm

The core algorithm of this program assembles a series of equations using information about the vertices and edges that is readily available from Google Sketchup then solves the series of equations using a simple equation. The equation construction consists of iterating through the vertices of the model. At each vertex, the program calculates coefficients for the supporting forces supplied by each connected edge and forces applied by the weight of these edges. The supporting force coefficients are put into matrix $[A]$, while the applied forces are put into column matrix $[B]$. Thus, matrix $[A]$ contains the coefficients of the variable forces and matrix $[B]$ contains the right hand sides of the equations. The values of the supporting forces are then calculated through the equation $[X] = [A]^{-1}[B]$, where $[X]$ is a column matrix holding the forces of all members and ground supports. Once the forces are calculated, the members are colored according to the compression or tension forces applied to them. Higher compression yields a redder shade, while higher tension yields a greener shade. A member with internal forces close to 0 are colored black.

3 Results

This algorithm provides accurate results for models that are properly formed. As can be seen in Figures 1 and 2, the members that one would expect to have a higher compression or tension force are more intensely colored than the ones with lower forces. The algorithm is also fast and analyzes the 18-vertex structure shown in Figure 1 within a few seconds. The 4-vertex structure in Figure 2 is analyzed even faster, taking less than a second to be colored. However, structures such as the ones seen in Figures 3-5 cannot be solved using the current algorithm or return unusual results. The reasons these cannot be solved properly are discussed in section 5.

4 Conclusions

This algorithm does what it was designed to do in analyzing drawn structures quickly and accurately. The output is very readable and interactive, clearly showing to the architect or engineer the most critical members of the structure. When used properly, it can quickly and visually show the user where the critical parts of the structure are so that they can be strengthened properly. It also shows where the non-critical parts are so that they can be removed or reduced in strength in order to reduce cost. The software has a number of limitations, but these limitations will be overcome with further work.

5 Limitations

Despite its accomplishments, this method is very limited in the types of structures it can analyze. Firstly, the structure must have exactly three points touching the ground. Any more or any fewer and the resulting structure has multiple solutions, rendering it indeterminate, as seen in Figure 3. Secondly, all faces must be triangles. Otherwise, the structure is unstable and indeterminate, as seen in Figure 4. Also, if too many forces lie directly along an axis, the structure might be indeterminate or give skewed results. Figure 5 is an example of such a case, in which the software gave an answer that was completely incorrect, showing members in tension rather than compression and putting undue force on certain members. These limitations could be avoided if a more complex algorithm was used in place of the $[X] = [A]^{-1}[B]$ equation, but that was outside of the scope of this project.

6 Further Work

This research will be continued in the author's master's thesis and expanded to work on all types of structures. Support for non-triangulated meshes through the insertion of shear forces that are provided by sheathing or other exterior materials will be added. Support for structures with more than 3 points touching the ground will also be added by using a different solving method. The method used in this paper would experience significant slowdown for structures with a very high number of vertices, so tweaks to the algorithm's speed is essential. The most computationally intensive part of the algorithm is the inverse function, so a different solving method would alleviate, if not completely solve, that problem. There are many things that can be done to improve this software, and they will be done over the coming months.

7 References

[1] *A new object-oriented finite element analysis program architecture*, Archer, Fenves, Thewalt

[2] *Creating Models of Truss Structures with Optimization*, Smith, Hodgins, et al

[3] *Optimal Truss Design Using Ant Colony Optimization*, Symeon Christodoulou

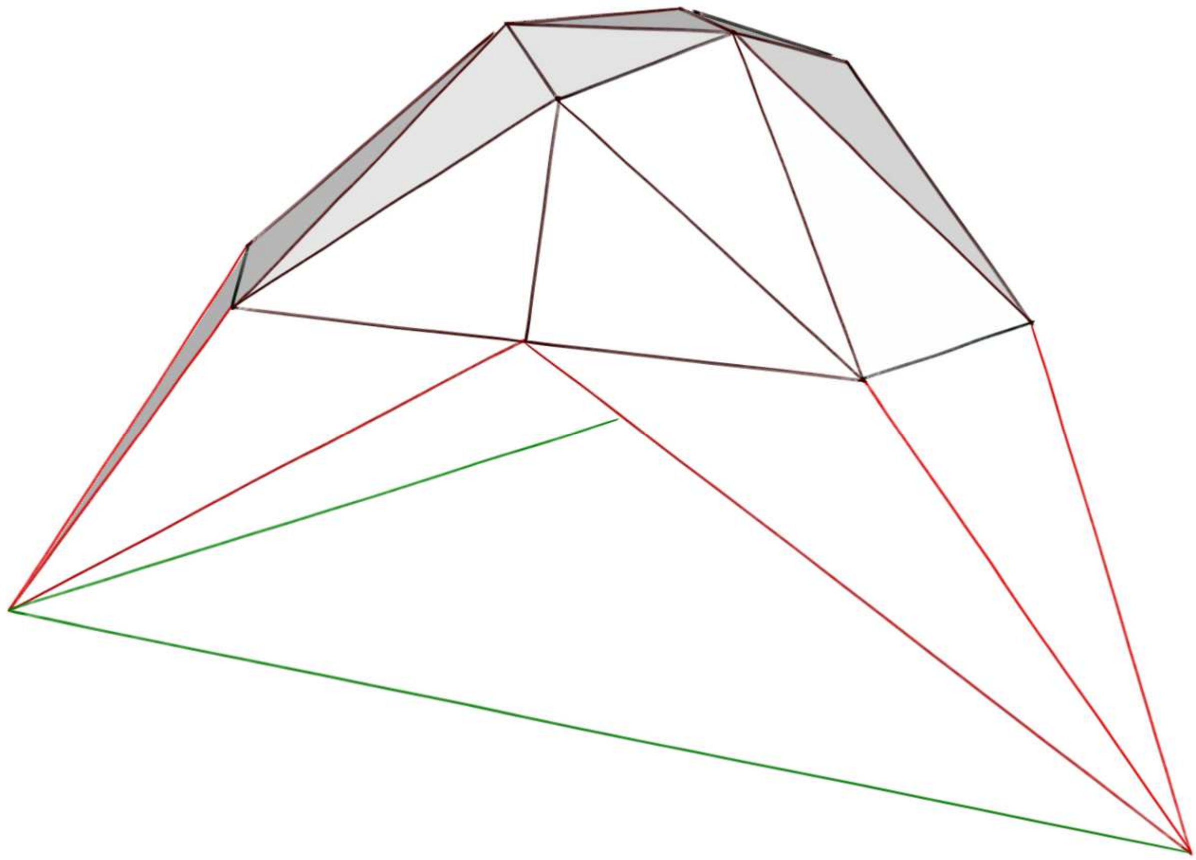

Figure 1: Dome shell structure

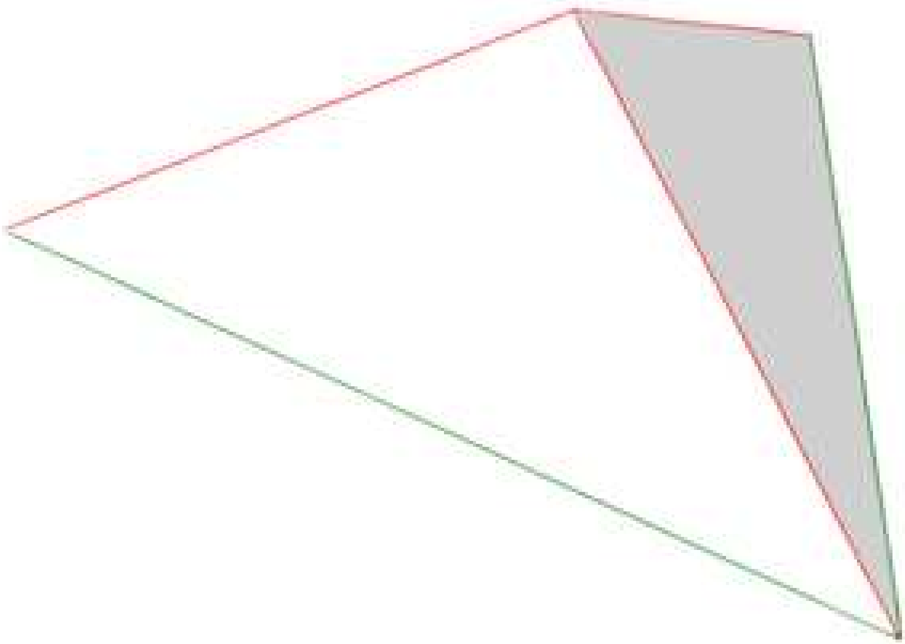

Figure 2: Simple tetrahedral structure

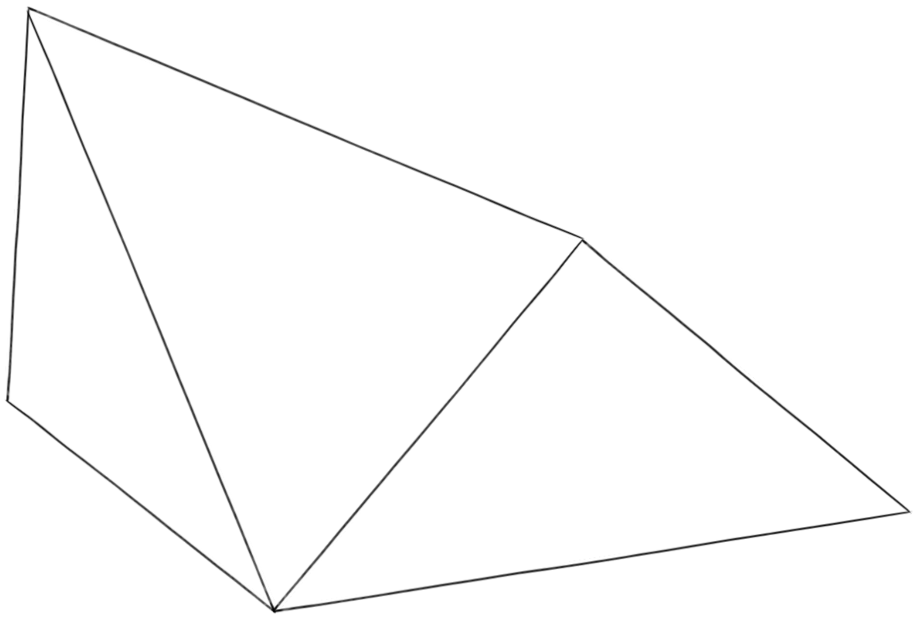

Figure 3: Unsolvable structure with four points on the ground

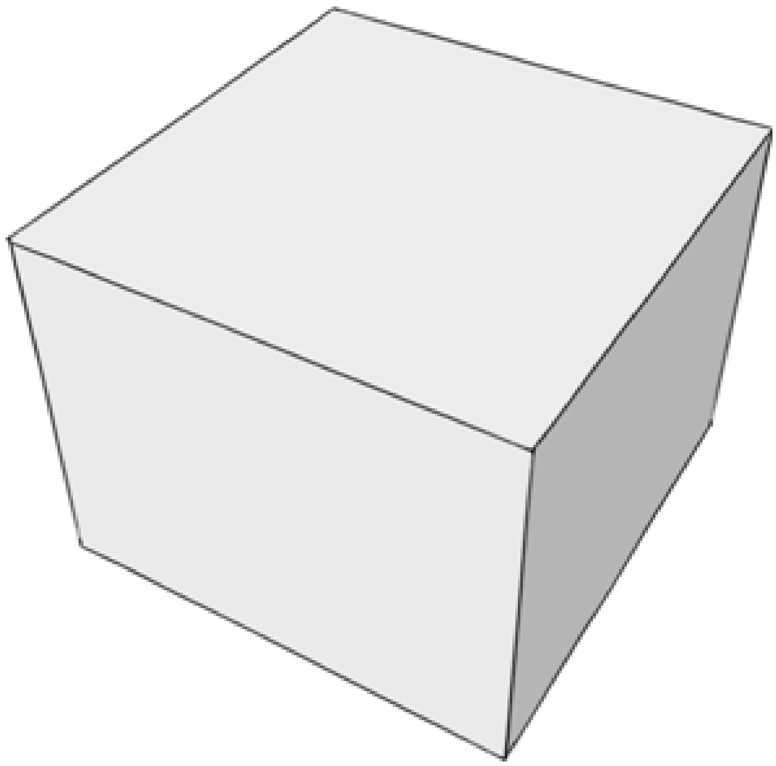

Figure 4: Unsolvable structure with quadrilateral faces

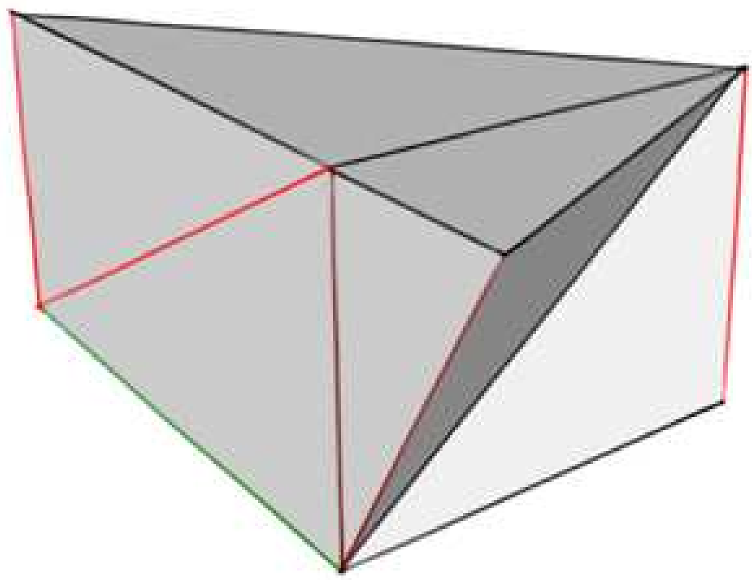

Figure 5: Improperly solved structure with members aligned to axes