

THIN SHELL STRUCTURE DESIGN TOOL

By

R. Allan Pendergrast

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved:

Barbara M. Cutler, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

May 2010
(For Graduation May 2010)

CONTENTS

LIST OF TABLES	ii
LIST OF FIGURES	iii
ACKNOWLEDGMENT	iv
ABSTRACT	v
1. INTRODUCTION	1
1.1 Project Goals	1
1.2 THIN SHELL STRUCTURES	1
1.2.1 What are they?	1
1.2.2 Why do we care?	2
1.3 THIN SHELL OPTIMIZATION	2
1.3.1 Building Stability	2
1.3.2 Hanging Chains	3
1.3.3 Cloth	3
1.4 Related Works	3
1.4.1 Procedural Modeling	3
2. THIN SHELL DESIGN TOOL	5
2.1 Layout	5
2.2 Simulator	5
LITERATURE CITED	5

LIST OF TABLES

LIST OF FIGURES

1.1	Lines of thrust	2
2.1	A screenshot of the tool	6

ACKNOWLEDGMENT

Thanks to Barbara Cutler for all her help in creating, testing, and writing about this tool.

ABSTRACT

Thin-shell structures are becoming increasingly useful in construction and design of buildings. They allow the usage of less material to enclose larger spaces, are structurally efficient, and have a natural aesthetic beauty. However, they can be difficult to design, as the exact shape required for structural stability depends on the material used, the size of the shell, and other features. Fortunately, it is possible to simulate these structures quickly and accurately, allowing architects to concentrate more on their design and less on ensuring that their building is stable. The tool described in this thesis simulates thin-shell structures and aids architects in their design and optimization.

1. INTRODUCTION

1.1 Project Goals

This project was originally conceived as a structural analysis tool that would find highly over- or underloaded members in a structure and notify the architect that some force could be diverted towards the less-used member or that the underloaded member could be removed. This redesigning would help to optimize the structure, resulting in a structure that required less material or work to construct. Since then, the focus has shifted to the design and optimization of thin shell structures. These structures, covered in more detail in 1.2, are used in architecture to conserve materials, funds, or for simply aesthetic reasons. Unfortunately, thin-shell structures have been famously difficult to optimize and construct, sometime requiring a complete overhaul of the structure to fix stability problems. This tool attempts to alleviate this difficulty and make the design and optimization of such structures easier and faster for architects.

1.2 THIN SHELL STRUCTURES

1.2.1 What are they?

A thin shell structure is a structure which has a small thickness compared to its other dimensions. While this may seem to be an obvious definition, the design and construction of these structures can be complicated. Unlike normal beam and plate structures, thin shell structures are curved, which allows the force to travel through the thinner structural elements. See 1.3.1 for more information on the stability of thin shell structures. Thin shell structures are used often in construction when a large interior space is desired, since they do not require any interior columns if designed correctly. Some prominent thin shell structures include the TWA Flight Center Building at the JFK International Airport in New York, New York, the Kresge Auditorium on the MIT campus in Cambridge, Massachusetts, and the Montreal Biosphere in Montreal, Canada.

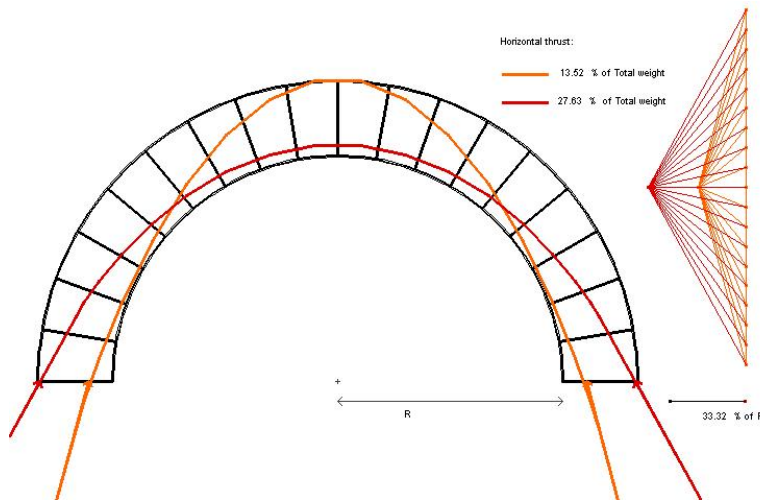


Figure 1.1: Lines of thrust

1.2.2 Why do we care?

Since the force travels directly through the structural elements in a thin shell structure, these structural elements can be thin. This thinness results in less material being used, which can save money for the companies involved. Furthermore, since the structure supports itself, it offers a large uninterrupted interior space, which can be used for any number of applications.

1.3 THIN SHELL OPTIMIZATION

1.3.1 Building Stability

In order for a building to stand up, the lines of thrust must pass through structural elements. As can be seen in Figure 1.1, traditional arches must be rather thick to contain the lines of thrust produced by their weight. However, a catenarian arch can be built much thinner for the same stability, as it contains the line of thrust exactly. To extrapolate to entire buildings, traditional construction methods require very thick elements to be used in order to keep the lines of thrust within a building's structural elements. However, if the shape of the building is instead matched to the shape of the lines of thrust, the structural elements can be much thinner, needing only to support the direct compressive force.

1.3.2 Hanging Chains

Hanging chains have been used by architects, most famously Antoni Gaudí, to design structures for stability and aesthetics. Since a hanging chain is supporting itself purely in tension and a catenary arch is supporting itself purely in compression, one can be used to describe the other. More complicated systems of hanging chains and weights can and have been used to design a number of structures, including the Sagrada Familia in Barcelona, Spain. Design by hanging chains offers a relatively quick and accurate turnaround time, but it is still tedious to adjust the chains to find the desired shape. Furthermore, inverting the model can be a tedious and confusing process. Fortunately, hanging chains are simple to simulate.

1.3.3 Cloth

A standard cloth simulation shares many similarities with the hanging chains model. A finite number of elements connected by links will, when submitted to gravity, form a catenary. In cloth simulation, a finite number of points exert forces on each other, falling into a tensionally stable configuration. Additionally, since the model is simulated, gravity can be reversed and the model shown as it would be built, making it easier for the architect to visualize the completed structure.

1.4 Related Works

1.4.1 Procedural Modeling

In "Procedural Modeling of Structurally-Sound Masonry Buildings" [1], Whiting, Ochsendorf, and Durand explore the possibilities of creating existing or novel structures procedurally. They began by creating a grammar which can be used to construct masonry buildings. Arches, buttresses, domes, and vaults are some of the structural elements which are then combined in their software. These grammar elements are assembled into a structure through a procedural algorithm which cuts windows in walls and assembles all the various masonry elements of the building. Once the initial configuration is generated, the software runs static analysis on the building. If it is feasibly stable, the program is done. If not, the program determines a measure of infeasibility, which is a measure of how far away from stable a structure

is. The static analysis only allows for compressive forces, as the tensile strength of masonry elements is close to zero. Friction is also modeled, allowing for some shear. Once the measure of infeasibility is calculated, a parameter search is conducted iteratively, searching the parameter space for a stable configuration. Depending on the application, this stable configuration will take into account a factor of safety. The more likely a structure is to have changing loads, the higher a factor of safety is needed. For example, a bridge needs a higher factor of safety than a cathedral. In the event that there is no feasible configuration for a structure, the least infeasible structure is returned and the user is required to add new structural elements.

In "Creating Models of Truss Structures with Optimization", Smith, Hodgins, Oppenheim, and Witkin propose a method of creating trusses procedurally. This work allows the user to define several anchor points and loads for a truss, then have the software automatically generate a truss. In this work, the risk of pieces falling apart is not an issue as it was in the previous paper. The primary failure method in this case is buckling, since all forces are axial. Therefore, the core of the algorithm is a multivariable optimization with constraints. The algorithm attempts iteratively to minimize weight while ensuring that none of the members will fail, either in tension or compression.

My work had initially intended to go in this direction, using static analysis of structures within Google Sketchup. However, Sketchup proved to be a poor environment for the program I wanted to write, so the project was moved to a standalone application and the focus shifted to thin-shell structures.

2. THIN SHELL DESIGN TOOL

2.1 Layout

As can be seen in figure 2.1, there are 3 main panels in the design tool. On the left is the viewing panel, where the structure is shown in 3D. This view can be rotated, zoomed, and panned to give the architect the view they need. There is no actual interaction with the structure in this panel; all of the interaction is done via the two right-hand panels. The top-right panel is the floorplan panel. This panel contains a point for each point of the structure that is in contact with the ground. These points can be moved around in order to modify the shape of the structure to fit the architect's vision. The bottom-right panel is the grid panel. This panel shows all the vertexes of the structure in 2D. This panel can be used to disable points, creating voids in the structure. It can also be used to associate different points in the shell with certain points in the floor plan. This is most useful when adding new points to the floor plan, allowing the architect to define a new point of support on the fly. Together, these windows allow the architect broad control over the design of their structure, while the tool works in the background to keep the shape of the structure optimal.

2.2 Simulator

The heart of the tool is the simulator. This simulator uses a second-order explicit integral solver to simulate the forces on the structure. These forces cause the structure to form catenaries, shapes that will support themselves. The floorplan, grid, and viewing window can all be interacted with in real time while the simulation is going on, allowing the architect to make changes on the fly. Depending on the magnitude of the changes made, the simulator will take anywhere from a second to a couple minutes to reach equilibrium. It should be noted that until the simulation does reach equilibrium there are no guarantees of the stability of the structure.

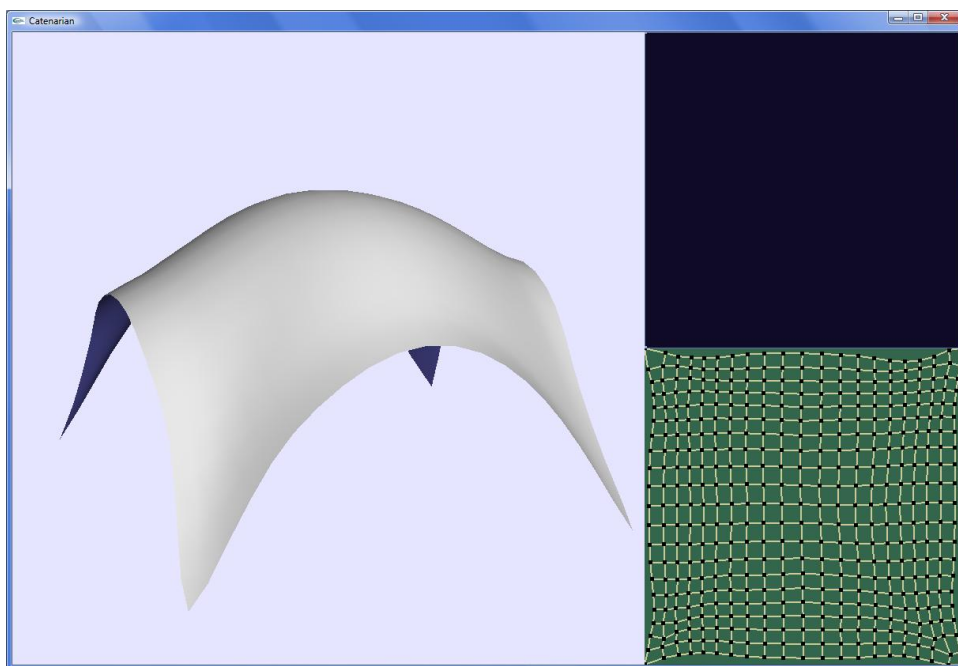


Figure 2.1: A screenshot of the tool

LITERATURE CITED

- [1] Emily Whiting, John Ochsendorf, and Frédo Durand. Procedural modeling of structurally-sound masonry buildings. *ACM Transactions on Graphics*, 28(5):112, 2009.