# A Tool for the design of Fabric Structures

Keith MacBain and William R. Spillers

IJAC 4-2 09/06/06 9:41 am Page 82

# A Tool for the design of Fabric Structures

Keith MacBain and William R. Spillers

This paper describes the development of a tool, now available for use over the Internet, for the preliminary design of a class of fabric structures. This tool is based on the so-called *grid method* in which vertical equilibrium over a grid in the horizontal plane is used to determine shape for a cable net. It can subsequently be combined with other analysis routines to produce a detailed structural design.

## 1. INTRODUCTION

Fabric structures present the design professional with a quandary. They can be geometrically engaging for the designer but difficult for the architect or engineer who has to provide the required analysis. In fact, Horst Berger (Berger 1996), the fabric structures pioneer, has been known to comment that fabric structures could not be designed and built if computers were not available to help.

The practicing designer must deal with difficulties from several sides. First of all, the behavior of fabric structures is described by nonlinear equations (Levy and Spillers, 2003) and there is simply no way around this point. That would not be so bad if there were simple tools available to deal with these equations. The absence of such tools is reflected in the fact that fabric structures engineers now find themselves using nonlinear finite element programs to design fabric structures. And there is some proprietary software out there but it is both expensive and difficult to use. Certainly, the situation is nothing like that for typical structural analysis software which is readily available and cheap.

The point of this note is to offer a simple tool for the conceptual design of fabric structures. At the early design stages, the nonlinearities mentioned above do not come into play. The possible exception has to do with the geometry of edge cables, which we argue below is not significant. We are offering a simple tool for generating *shape* for fabric structures. It leaves the nonlinearities to the detailed design phase.

## 2. FABRIC STRUCTURES

The design problem of fabric structures (Huntington 2004) typically involves three steps: shape finding, analysis, and patterning. Preliminary design or conceptual design, on the other hand, is primarily concerned with shape and should not require the architect, for example, to have to deal with the technicalities of nonlinear structural analysis. That is the point of the algorithm described here.

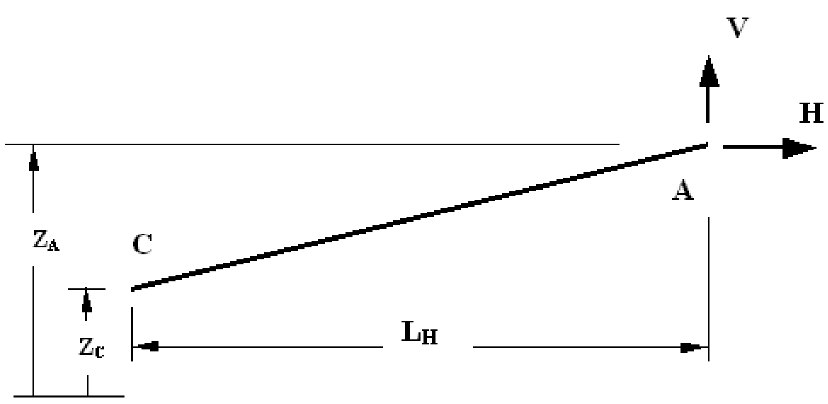
The work that follows is described in terms of cable nets. Whether the designer is actually working with a cable net or fabric, the cable model is very physical and, in these terms, seductive. But it should be noted that what is done here could also be done in terms of membrane finite elements.

The fabric structures industry is relatively small and difficult to get a handle on. But there seem to be three basic approaches to shape finding. Probably the most natural approach to shape is the use of deformed structures. Some years ago, this would have been done by loading a membrane and computing the small displacements, which would be subsequently scaled up to achieve the desired geometric effect. There were shortcomings to this method because, in many cases, the fabric is stiff and subsequently *patterned* to achieve shape while the deformed structure would suffer stress concentrations in the vicinity of the applied loads. (It should be noted that modern nonlinear analysis packages now allow stress adjustments to be made largely eliminating the shortcomings of earlier attempts to find shape through the use of deformed structures.)

The other two methods of shape finding deal more directly with the equations of equilibrium. In the *force density method*, it is assumed that the ratio of the cable force to the cable length (the force density) is known, leaving a linear set of equations to be solved for the shape of a structure. Since this ratio cannot be known *a priori*, shape finding using the force density method turns into an iterative process.

In the *grid method* (Siev and Eidelman 1964), the shape finding process starts with a grid of cables that is in equilibrium in the horizontal plane. (If there are edge cables, they must also be in equilibrium with this grid.) Solving this edge cable problem has always been an annoying initial step required by the grid method. There are, of course, elevations to be specified over this grid. The remaining node elevations (the shape) must be determined from the equations of equilibrium. These are linear equations, as will be seen below.

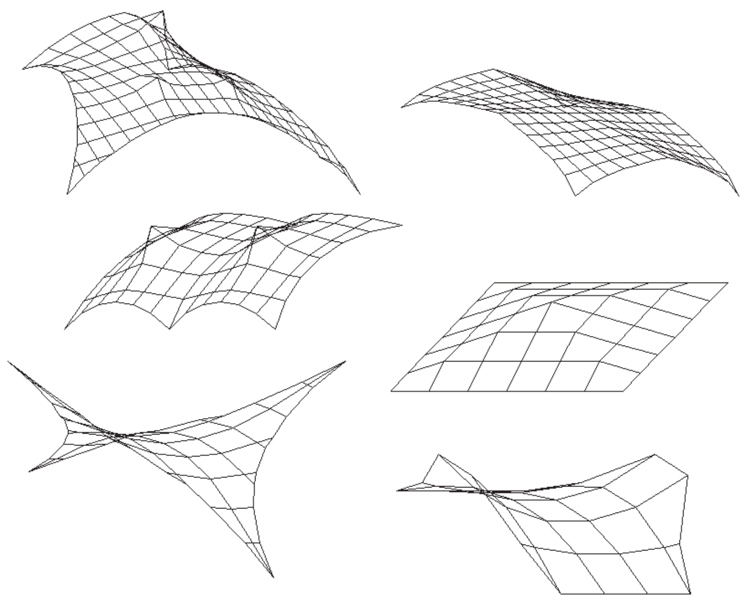
The algorithm described below is based on the grid method. The user starts by selecting a grid and then sketching in the edge cables by graphically specifying their end points and their center offsets. Next, the given elevations are introduced. Finally, the user views the resulting cable net in three dimensions. Some designs obtained using this algorithm are shown in Figure 1.




◀ Figure 1. Design examples

Our algorithm does not apply to all fabric structures geometry, as we assume that the rectangular grid would be reflected in the orientation of the fabric strips and consequently the load path. For example, it would not make sense to employ a rectangular grid where radial geometry is desired. (See Figure 2) We note that the algorithm used to generate this figure has been available on the Internet *(http://www-ec.njit.edu/civil/fabric/index.html)* for some time.

► Figure 2. A cable net with radial geometry



## 3. FEATURES OF THE ALGORITHM

With this algorithm the user is forced to work within the confines of a grid. Specified joints and the ends of edge cables must all be described in terms of node points on the grid. The user begins using the algorithm by defining the grid lines and their spacing in two dimensions.

The next step has the designer specifying the edge cables. In the past, that has required some side calculations to determine the intersection points between the edge cables and the grid itself. In order to avoid hand calculations, the algorithm assumes that the edge cables have the form of circular segments that are determined by their end points and their center offsets. The edge cable force is determined by the given "pressure" p which is the cable horizontal force component divided by the cable spacing. (This is analogous to the pressure in the wall of a pipe.) The circular geometry of the edge cables allows the points of intersection of the edge cables and the grid to be easily determined automatically.

Note at this point that two approximations have been made here with regard to the edge cables: a) real edge cables do not lie along circular segments, and b) the real forces in the edge cables are not those obtained from the "pipe" analogy used here. But it is our experience that when corrections are made to remove these approximations, they are not visible to the naked eye.

Once the edge cables have been described, the fixed elevations should be specified. When these elevations have been set, the information concerning the design has been completed, and the shape can be viewed by clicking on the appropriate icon. At this point the computer solves the linear equilibrium equations. These equations can be constructed by summing the contribution of each bar to the system matrix as described in the Appendix.

If more detail is required than simply the preliminary shape, the following steps can be carried out:

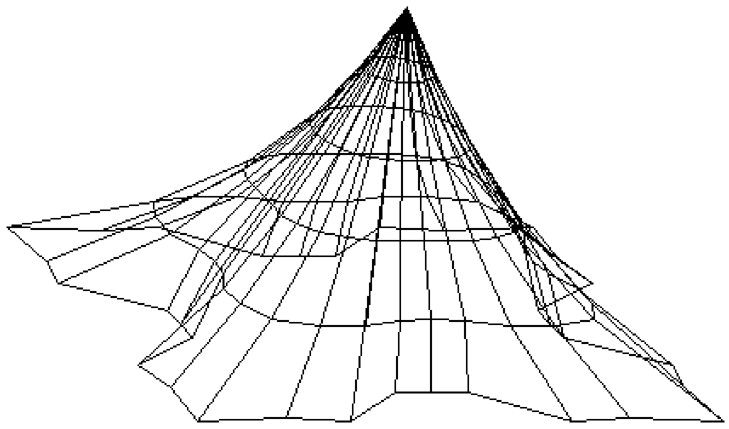
- The edge cables and the fixed nodes can be described as suggested above.
- Prior to solving for the shape, a nonlinear analysis can be run that allows the edge cables to move from their circular shape into their real equilibrium positions.
- Then the equilibrium equations can be solved for the shape.
- If uniform cable forces are desired, the nonlinear analysis can be carried out again, allowing the nodes to move to new equilibrium positions.

## 4. COMPARISON

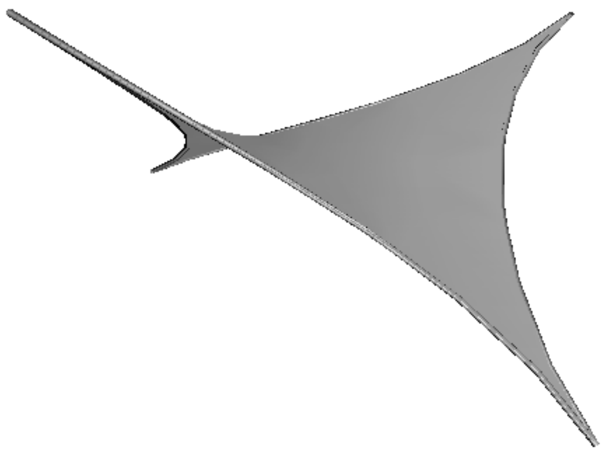
In the early stages of design where the goal is to quickly generate one or more forms for qualitative evaluation, the deformed-structures approach performs poorly when compared with the grid method. No matter if one is using membrane finite elements, a cable-net approach, or a hybrid of these two, the time to create and solve a structural model will be greater. Further, it requires the use of nonlinear analysis software (e.g., ADINA, ANSYS, ABAQUS) which is typically expensive and not of interest to the architect. Of course, structural analysis is inevitable for a final design and one cannot overlook the fact that if the developed structural model is selected for final design, the effort was well invested. The comparison between the grid method and the deformed-structures approach thus reduces to the following three considerations.

- What stage is the design in?
- Who is guiding and developing the design?
- Is there more than one design to be considered and is the comparison to be qualitative or quantitative?

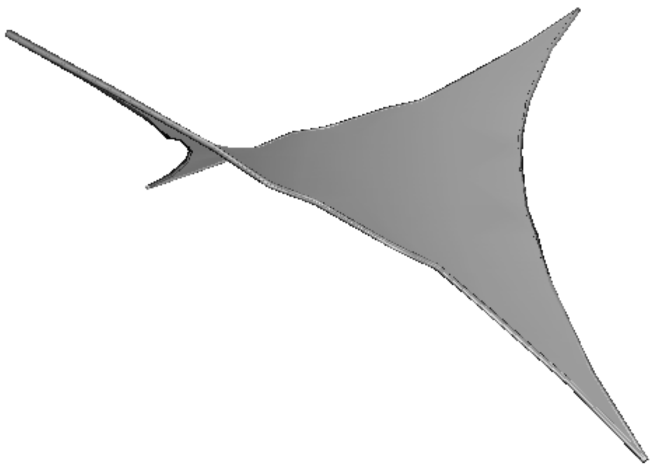
The force-density method can give the designer greater control over the form and is often a good choice for more refined models. This approach is particularly useful for the engineer when developing an analysis model in that it gives an initial equilibrium configuration most nearly approximating a set of desired element forces. Unfortunately, as mentioned earlier, this approach is often iterative and generally requires more user involvement than the grid method. Implementations such as Meliar's MPanel, for example, give the user more control over the surface by allowing one to specify target cable forces as well as the fabric stress ratio in the principal directions. However, if the implementation is not coupled with an analysis tool, the extra effort of specifying fabric stress and cable tensions is lost. In contrast, the grid method also allows one to specify vertical forces (in a manner such as Berger's FormFinder) but it is not possible to vary the horizontal force, which remains constant. Further, this is step can be entirely omitted as is shown here.



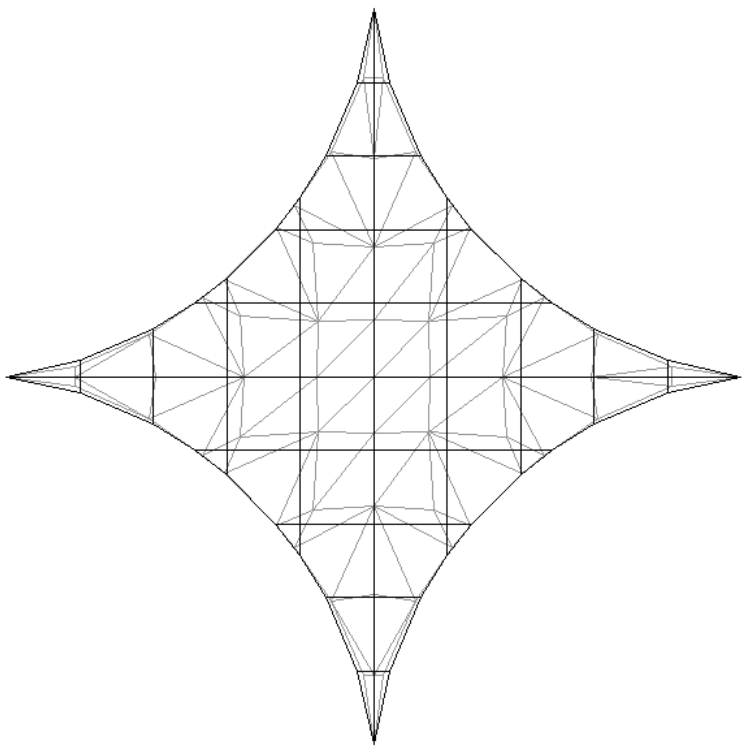
► Figure 3. Form using the force-density method



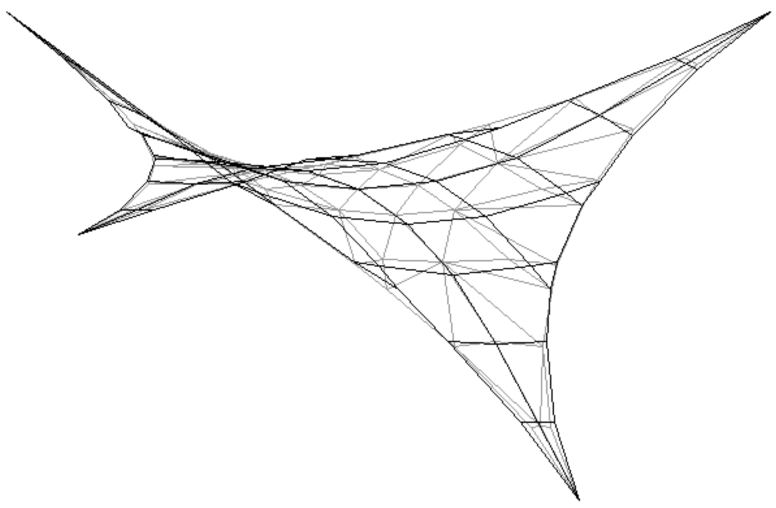
► Figure 4. Form using the grid method



► Figure 5. Plan view of two meshes superimposed, with the grid method in black, force-density method in grey.

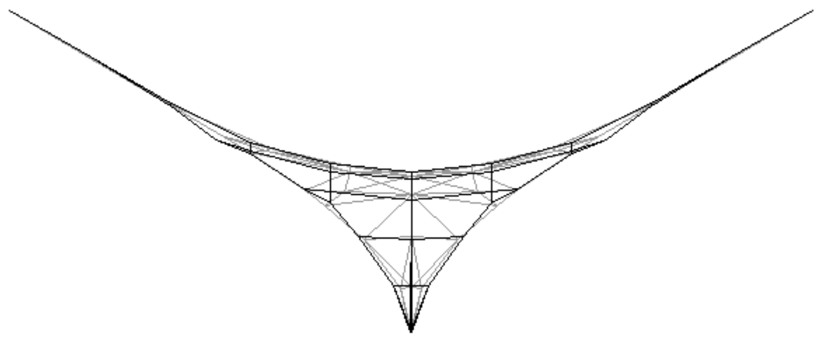


◀ Figure 6. View of two meshes superimposed

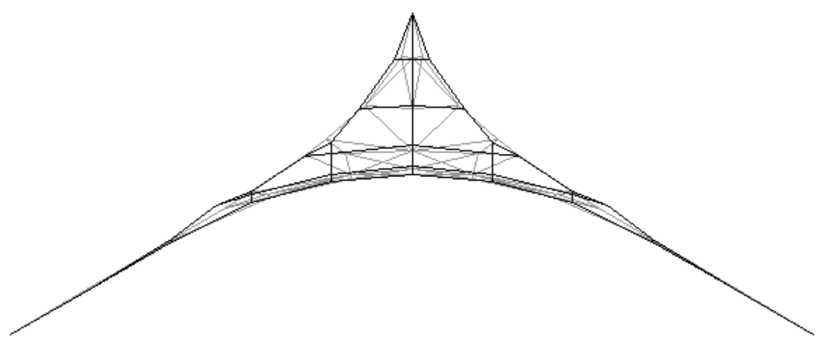


◀ Figure 7. Front elevation of two meshes superimposed

For a more qualitative comparison, a sample form was created using the force-density method and is rendered in Figure 3. This may be compared with the form generated by the grid method shown in Figure 4. The surfaces defined by the two meshes are practically identical, even though the actual meshes are not. The meshes from the two methods were superimposed and a plan view is seen in Figure 5, which shows that the nodes have taken different locations in plan. This is due to the fact that the force-density method does not constrain the nodes to a grid. The superimposed meshes are seen again in Figure 6. The elevation views of the two meshes in Figure 7 and Figure 8 further illustrate that the forms are nearly identical even though the location of the corresponding nodes within the surface is not the same. A more detailed view of the difference in plan is shown in Figure 9, where we note that, as mentioned earlier, the edge cables do not necessarily lie on circular arcs. Finally, we note that the form obtained by the force-density method required the selection of a target fabric stress of 20 pounds per inch and an edge-cable force of 2.5 kips, both of which were determined iteratively, and that this process took much longer than using the grid method.



► Figure 8. Side elevation of two meshes superimposed



► Figure 9. Detail of plan view of two meshes superimposed

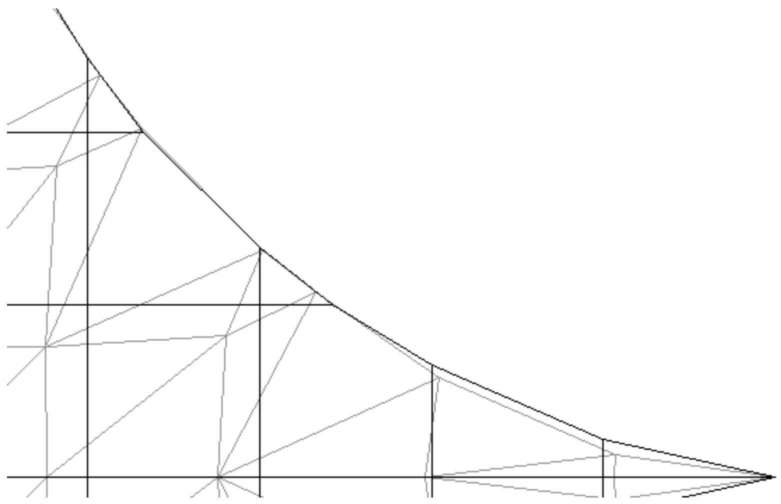
## 5. IMPLEMENTATION

The procedure described above is implemented in Java and can be run as an applet (in a web browser) or as an application *(http://www ec.njit.edu/civil/fabric/layout/main.html)*. The authors note the following benefits of selecting Java for this application.

- It is easy to code a graphical interface.
- It can run on several platforms (Mac, PC, Unix, ...).
- It can run in both Internet and intranet environments, making it easy for designers to exchange ideas.

Briefly, this implementation requires the user to first specify a plan layout, and then to assign elevations at particular locations thus determining the form. This results in three logical user modes: plan layout (i.e., placing and editing arcs), specifying the elevations of selected nodes, and finally viewing the resulting form. The final mode allows the user to navigate the 3D form in space and as the user changes modes, the program responds accordingly. This implementation also includes preset examples, which were selected to demonstrate some of the forms that can easily be created. Some details of the implementation are described here.

The plan layout is based on a rectangular grid, however the final form need not be rectangular. The user will typically draw edge cables on the initial grid to achieve the desired shape in plan. As mentioned earlier, edge cables are circular arcs and are defined here by their endpoints and center offsets. Our implementation terms this stage of the process as the arc-editing mode and an example of an initial grid with arcs in place is given in Figure 10.

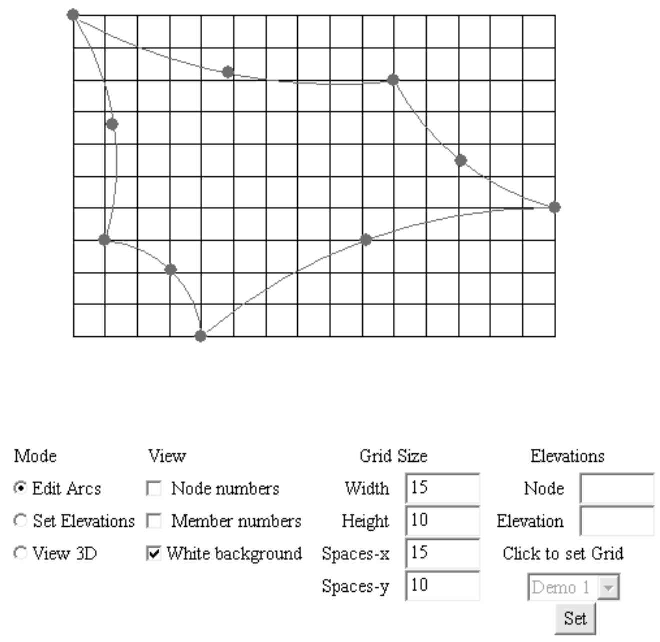


◀ Figure 10. Example plan layout

Once the plan layout is set (i.e., the arcs are in place), the mesh is modified by removing nodes internal to the arcs and adjusting the member connectivity accordingly. To remove portions of the initial mesh, a scan is done over all members. This scan is done with respect to each arc. Each member in the system belongs to one of the following three categories: for those members that have two nodes inside the circle defined by each edge cable, the entire member and both nodes are removed from the system; for members having both nodes outside the circle, nothing is done; finally, in the third case, members have one node inside the circle and one node outside. In this case, the node inside the circle is moved along the axis of the member so that the node interior to the circle is placed on the circumference. The number of this node is stored so that, when all members have been scanned, these nodes may be connected, typically to form an edge cable.

The new members placed along the arc are assigned a force according to a pipe analogy. Consider Figure 11, which shows a semi-circular arc and a portion of the remaining initial mesh after the arc-cutting routine. Given Fi the horizontal force component specified for each member in the rectangular grid (taken here as a unit value), the force Fc in the elements along the arc can be expressed as a function of Fi, the radius of the arc, and the grid spacing. In the algorithm, a fictitious *pressure* is computed as Fi divided by the cable spacing. From strength of materials this gives the edge cable force as this pressure times the radius of the edge cable. In our implementation, these computations are done when the user begins the next step, specifying elevations. Figure 12 below shows the example given earlier in Figure 10 after this step.

► Figure 11. Edge cable forces

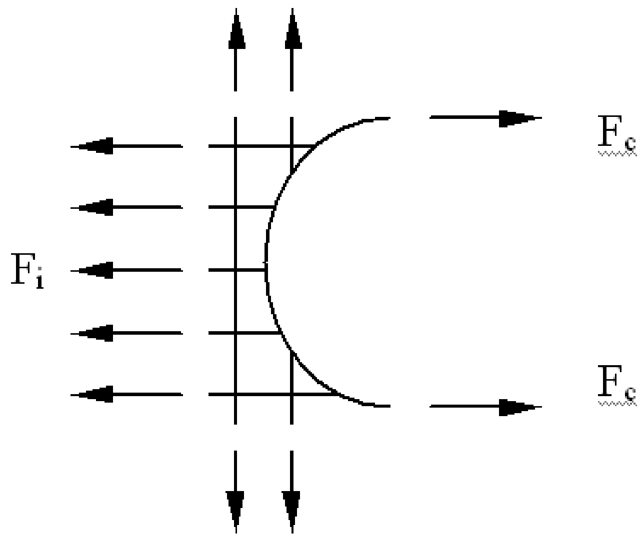



The next step is trivial: the user specifies the known elevations for particular nodes, typically supports. Finally, the system matrix is assembled (see the Appendix) and solved for the shape. Figure 13 below shows the example presented below in Figure 12 after specifying elevations and solving for the shape. At this point, the user will typically review the form and possibly return to the earlier steps to change some parameters in order to modify the form.

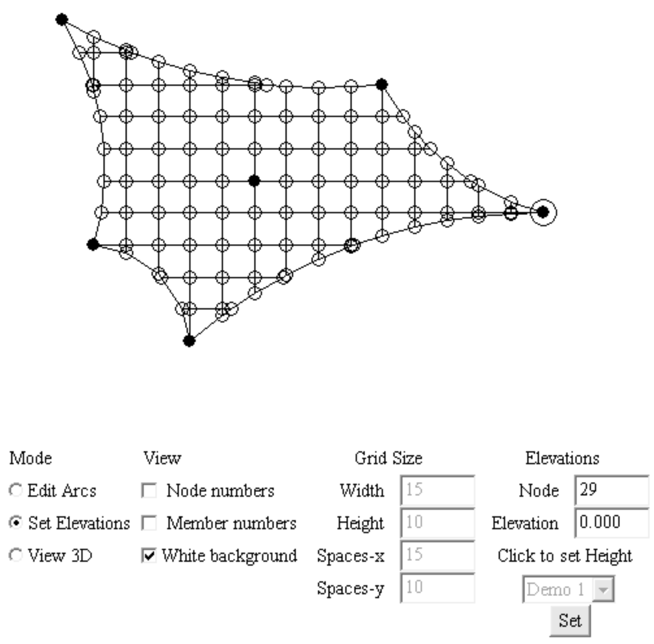
Future work for this implementation might include allowing multiple users to simultaneously view and modify a design and an option to store multiple designs for comparison.

## 6. CONCLUSIONS

This paper presents the details for a simplified method for form finding of fabric structures. A program developed in Java is provided which implements this method and some of the details of the implementation are discussed. The grid method is considered a valuable tool, particularly for the preliminary stages of design where ease-of-use and quick visual feedback can be more important than precise calculations. Designs can be created and modified in only a few minutes, giving both the architect and the engineer a quick means to visually explore and convey concepts as well as a starting point for more detailed structural analysis. Quick approximate design tools such as this allow architects to understand structural limitations and have more meaningful dialog with engineers.




◀ Figure 12. Example with edge cables placed



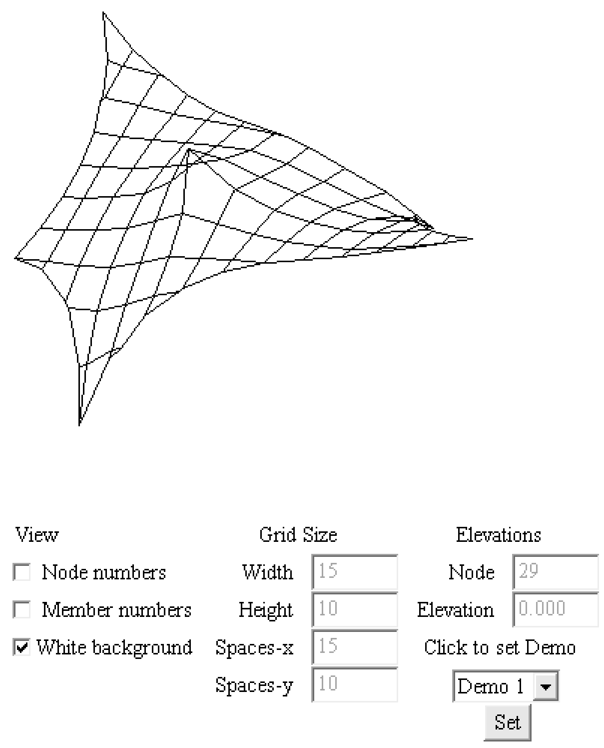

◀ Figure 13. Example in 3D

## References.


1. Berger, H., *Light Structures*, Structures of Light, Birkhauser, Basel, 1996.
2. Huntington, C. G., *The Tensioned Fabric Roof*, ASCE Press, Reston, VA, 2004.
3. Levy, R. and Spillers, W. R., *Analysis of Geometrically Nonlinear Structures,* Kluwer, Boston, 2003. (See also http://web.njit.edu/civil/fabric/notes)
4. Siev, A. and Eidelman, J., Stress Analysis of Prestressed Suspended Roofs, Paper 4008, in *Journal of the Structural Division, Proceedings of The American Society of Civil Engineers,* 1964, Vol 90, ST4, 103-121.
5. Otto, F., *Tensile Structures* (Volumes 1 and 2), MIT Press, Cambridge, Mass, 1967.
6. Irvine, M. H., *Cable Structures*, MIT Press, Cambridge, Mass, 1981.
7. Leonard, J. W., *Tension Structures*, McGraw-Hill, NY, NY, 1988.
8. Shaeffer, R. E., editor, *Tensioned Fabric Structures*, ASCE Press, NY, NY, 1996.


## Appendix:

► Figure 14. A typical cable segment




### The Vertical Equilibrium Equations.

The vertical equilibrium equations that determine shape can be easily generated as follows. Figure 14 indicates a typical cable segment that is in equilibrium in the horizontal plane under the horizontal force component H. (H does not change during the shape finding process.) The vertical force component V, in this figure can clearly be written as

$$V = H\ (ZA - ZC)\ /\ LH \qquad (1)$$

It follows that, if the vertical equilibrium equations are written as A Z = 0, the contribution of a typical member i to these equations can be written as

$$
(A)_i = \begin{array}{cc|l}
H/L_H & -H/L_H & \text{Row A} \\
-H/L_H & H/L_H & \text{Row C} \\
\hline
\text{Col. A} & \text{Col. C} &
\end{array}
$$

Here Z is a column matrix of the node elevations and that the equations so generated must be modified to include the effects of the nodes that are fixed in elevation.

Keith MacBain
Engineer, Geiger Associates
2 Executive Blvd
Suffern, NY 10901 USA

William R. Spillers, M.ASCE
Professor, Civil and Environmental Engineering
New Jersey Institute of Technology
Newark, NJ 07102 USA