

Plank Lines of Ribbed Timber Shell Structures

Karen Kensek, ^{3D}CLIPPER LAB, University of Southern California, USA

Judith Leuppi, ^{3D}CLIPPER LAB, University of Southern California, USA

Douglas Noble, ^{3D}CLIPPER LAB, University of Southern California, USA

Abstract

This paper discusses a method for determining the plank lines of ribbed timber shell structures. The information is necessary for the construction of the roof, but the information is usually not depicted accurately in three-dimensional modeling programs.

Keywords: geodesic line, finite element algorithm, ribbed shell structure, lightweight structure, timber construction.

1 Introduction

It is often a problem with digital modeling systems that the method of depiction of architecture does not correspond to the information needed to actually build the structure. This is decidedly the case with ribbed timber shell structures. The most common methods of portrayal consist of fairly arbitrary meshes that give a good indication of spatial characteristics, but fail to provide necessary information for the structural engineer (figure 1).

The typical construction of ribbed timber shell structures consist of wooden planks to form the ribs. These planks are approximately 3 x 16 cm, and several interlocked layers are used to form each rib (figure 2). These planks are screwed together; usually no glue is used. Each layer of planks is continuous in one direction. Filler boards are installed between these planks. The critical feature is

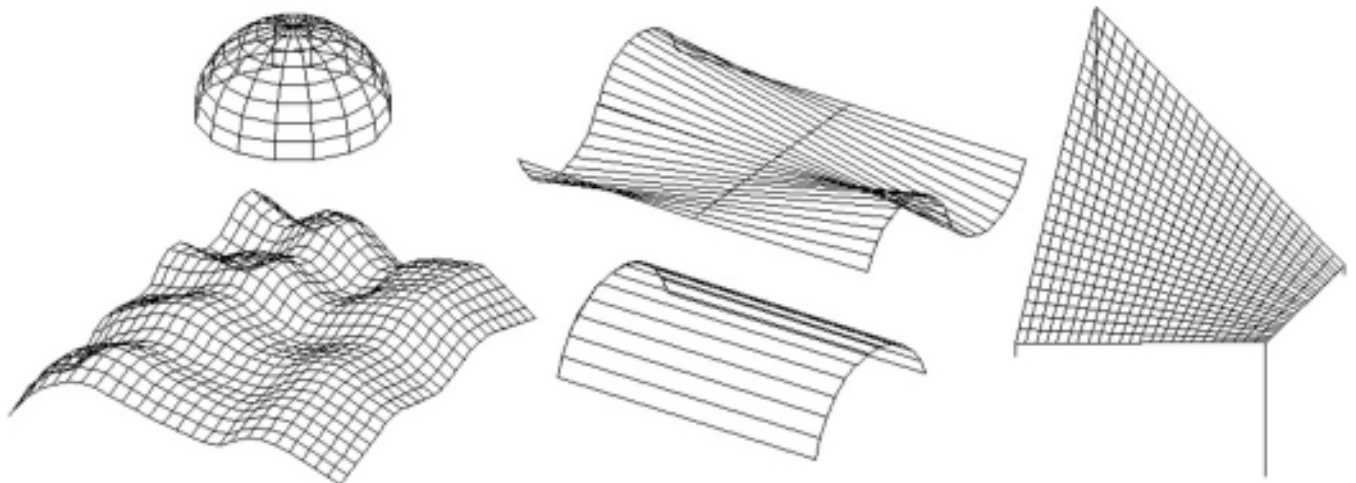


Figure 1. Images of AutoCAD meshes.

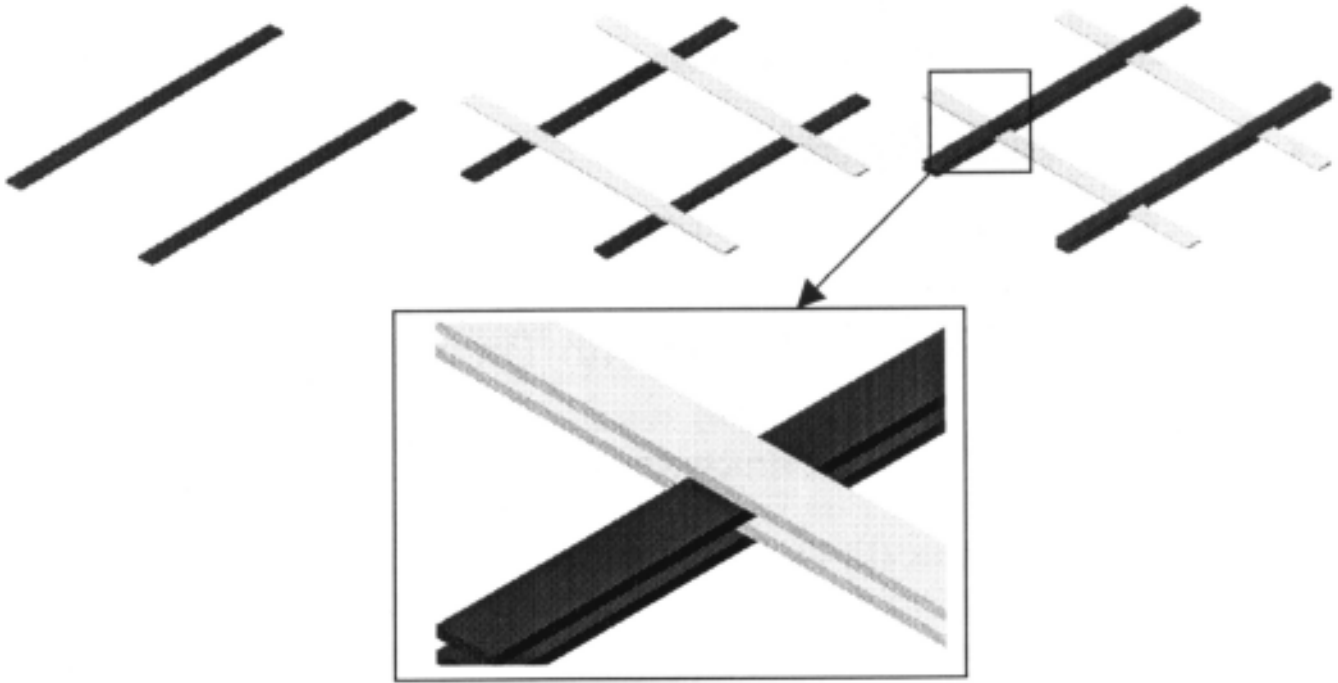


Figure 2. Interlocked construction method for wooden ribs (shown without filler blocks).

that the wooden planks can only be bent, within reason, in their weak axis of inertia, the long direction. If they are bent in the other direction, they break easily.

2 Example and Current Practice

An example of this type of construction is the exposition pavilion, the Polydôme, in Lausanne, Switzerland (1991) (figures 3 and 4). The roof is a partial sphere, approximately 25 x 25 meter square in plan, with a height of 7 meters. The radius of the sphere is 27.5 meters. The wooden planks follow the geodesic lines of the sphere.

Note that ribs in this spherical example follow the great circle lines, lines that pass over a surface and do not curve in the tangential plane. As mentioned earlier, this is crucial when using wooden planks. In the case of real structures, “great circle” or “geodesic line” may not be the correct terms to use because the wooden ribs have a real thickness. The term “plank line” can be defined as a line with an assumed width (such as a plank of wood) that passes over a surface and does not curve in the tangential plane and whose width is always tangential to the surface (figure 5).

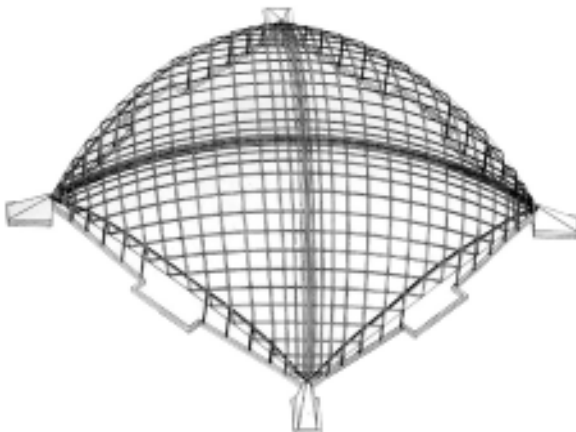


Figure 3. Drawing and photograph of the Polydôme (BCN Natterer; IBOIS).



Currently these plank lines are found in two major ways. If the roof has a simple geometric form like a sphere, then they can be found graphically with a three-dimensional drawing program. Sometimes this leads to roofs that are not structurally ideal, for example, they might be too flat, which causes buckling problems. For more complex shapes, a mathematician might be hired as a subcontractor. These shapes were often limited to those that could be described with geometrical equations. Each time the structure was changed during the design process, the mathematician would have to recalculate the lines. This could become very costly.

Figure 4. Construction photograph and detail of the Polydôme (IBOIS).

3 Proposed Solution

The solution proposed would take a three-dimensional design drawing of a ribbed shell structure, parse a dxf file to determine the necessary geometric entities, and run the calculations to discover

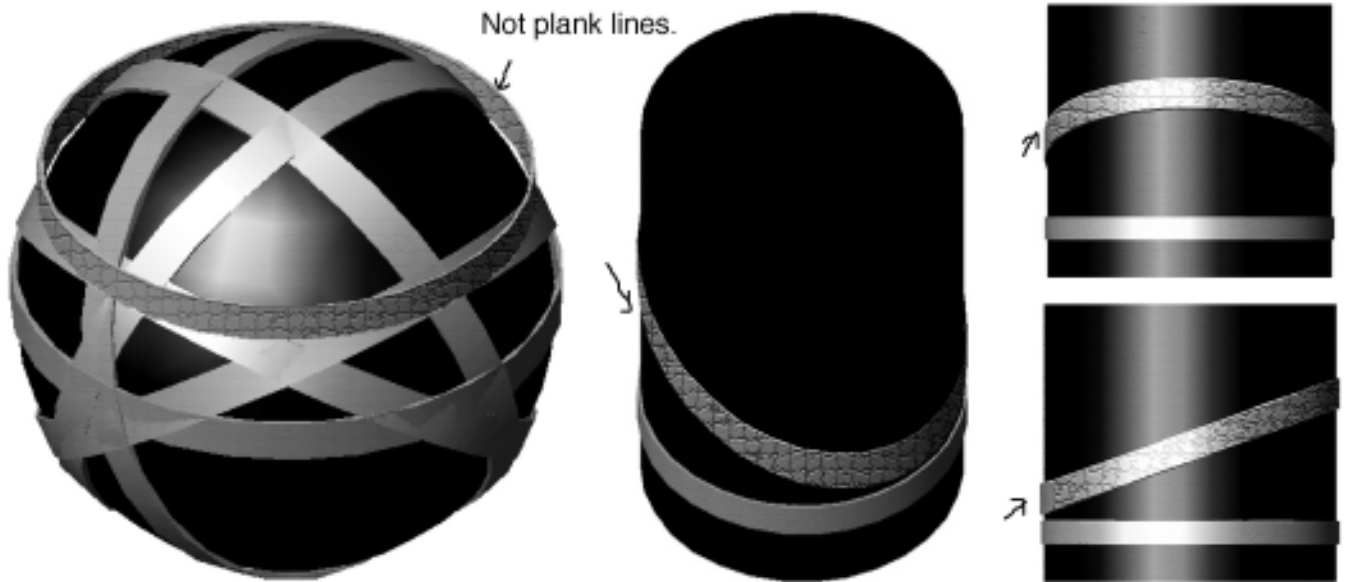


Figure 5. On the sphere, due to the width of the line, the red band is not a plank line; a cylinder cut with a 2D plane does not make a plank line.

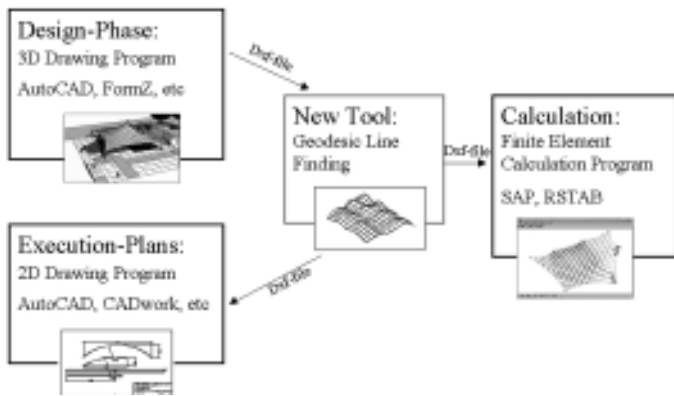


Figure 6. Relationship of proposed tool to design, CAD, and analytical programs

the positions of the plank lines (figure 6). It is important that the original design drawing follows a few guidelines; these include that the shape is defined by four edges, it has been meshed, and the mesh has been exploded into its component faces. C++ is being used for the development of this program. After the program's calculations are completed, the data is translated into a new dxf file and sent to drafting programs for inclusion into working drawings or sent to structural analysis programs.

The shapes can be divided into three classes: spherical and cylindrical, anticlastic, and free form. For each, the input will be the dxf file of a dense mesh (that will be used as the

finite element grid mesh in the program), the number of planks or distance between planks, and the starting angle. The output will be the locations of the plank lines as a dxf file. For the free form shapes, in some cases there may not be a solution.

4 Finite Element Approach

Although it is possible to use geometric equations to solve for these intersection points in some cases, an approach using finite element algorithms was chosen for four reasons. In theory, it is simpler because one algorithm could be used for every kind of shape. Second, the grid of the structure is already similar to the finite element grid, so finding the solution is comparable to what the initial design problem is. Third, the initial shape is not constrained to being defined by geometrically definable forms; this gives the designer more freedom in the initial architectural design. Fourth, by using a finite element approach, the problem is broken down in many small, easier to solve problems; recursively the larger problem is solved.

4.1 Principle

The principle of using the finite element approach is straightforward. The program receives the dxf file with the shape's geometric information and finite element mesh, and the user defines the starting points of the planks. The program will choose one of the planks x,y,z location (on the edge) as the initial condition. It then needs to calculate the end point of the sub-line within the small finite element grid. This end point becomes the starting point for the next, neighboring element (figure 7). Step by step the line finds its way (through calculation) over the shape to the other edge thus defining the plank line (figure 8). This process is repeated for the starting point of each plank along the edge of the shape. It is necessary to set up local coordinate systems to aid in this process's calculations and provide the rules that govern the calculation of the end point of the sub-line.

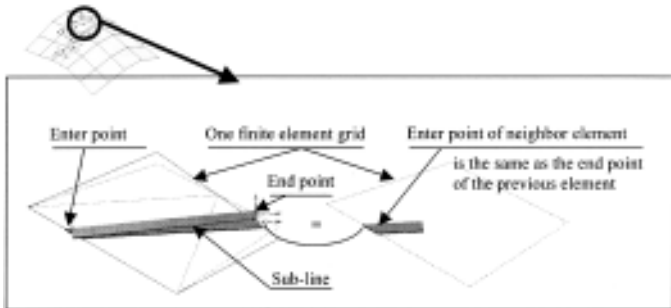


Figure 7. One finite element grid showing a sub-line, the part of the plank line within one finite element grid

4.2 Local Coordinate System

A local coordinate system is defined at the point where the line "enters" a finite element grid. The X-axis is the longitudinal axis of the plank at the enter point. The Y-axis is perpendicular to the X-axis and lays in the plane of the finite element. The Z-axis is determined by the right-hand-rule and is perpendicular to the plane of the finite element. Each finite element's plane is determined by two corner points and the middle point of the other two corners (figure 9). Once the end point of the sub-line in the current finite element is calculated, the new orientation of the local coordinate system is established. This new orientation of the local coordinate system is also the start point of the new finite element, and the calculations are repeated until the other edge of the shape is reached. This occurs for each plank line.

4.3 Properties of Plank Line

Following a set of rules governed by the properties of the plank line, the program can determine the end point of the sub-line in a specific finite element. Rotation around the local X-axis is allowed for a small degree (twist of the plank). Rotation around the local Y-axis is allowed within limits (curvature of the plank over the shape). Rotation around the local Z-axis is forbidden (rotation in tangential plane; the plank would break!)

4.4 Intersection of Plank Lines

Once the plank lines are calculated, their intersections need to be found. It is improbable that the lines will hit exactly. The program must determine the closest that the lines come together at an implied intersection and then make allowances for the offsets. The x,y,z coordinates of the intersections are the information that the structural engineer needs. Currently the program does not calculate the intersection points, but instead generates the plank lines and outputs them as a dxf file.

5 Program Validation

The program has been tested on different surfaces, from relatively free form shapes to hyperbolic-paraboloid, anticlastic, barrel vaults, and spherical shapes (figure 10). Visually checking the results against existing projects (such as the Polydôme, see figure 3), indicates that the program is working correctly. Further testing must be done by comparison of mathematically calculated plank system lines with the program's output for the same surface. The student author plans to meet with the planners of one such project to compare the results.

6 Program Optimization

Although this program will solve for the plank lines, its usefulness as a fully-developed tool is very limited. Future work still needs to address problems with the use of AutoCAD, problems in the program coding itself, and difficulties with the program's logical design and the user input. Briefly, AutoCAD has a few limitations that make the use of the program less than perfect. These include limitations on the length of text strings, minor problems in using the "option" feature in the output as dxf command, and conversion of files from AutoCAD 2000 to AutoCAD 14. Within the plank line program itself, dynamic arrays or linked lists should replace the static arrays that are currently used, and the user interface needs considerable reworking. This also applies to the user input section, and for added utility the program should check for maximum bending of the timber planks, locate the exact intersection points of the plank lines, propose meaningful solutions, and most importantly allow the designer more freedom in specifying starting and ending locations of plank lines.

The current version of the program can only be used by a designer who is experienced with ribbed shell structures. The necessary distance, the best orientation of the ribs, and the ideal form of the surface are based on the judgment of the person using the program. If bad input is given, the plank lines are calculated anyway. The program could be expanded with features for narrowing down the scope of the possible solutions to a handful of meaningful solutions and not allowing insignificant solutions. In addition, the program could be integrated with a structural analysis tools for calculations regarding structural performance (stability, stress distribution, and deflection).

7 Conclusion

This program will solve for a set of plank lines for ribbed timber shell structures. Although narrow in scope and work should be done to improve its overall usefulness, it does provide a real-life solution for architects and structural engineers who are struggling with this problem. Applied at different scales and for different materials, the program could also be used in other fields than for rib shell structures, for example, figure form work for double curved concrete structures; designing molds for glass, plastic, and lightweight cement; and for designing haute gammed furniture. Whenever a one directional element has to be curved and has a width larger than the height, the essential problem is the same.

Special thanks to Professors Schierle, Schiler, and Wagner, and Professor Natterer and his coworkers in the department of IBOIS, and offices BCN and IEZ, for using their pictures and tapping their knowledge.

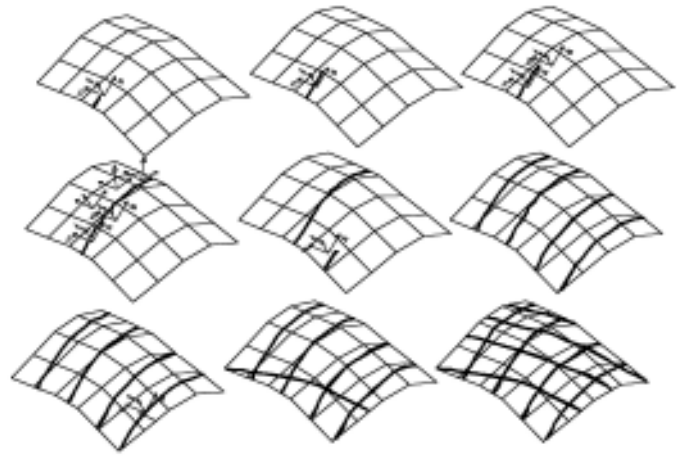


Figure 8. Schematic representation of the entire finite element sequence showing the incremental progress of sub-lines across the grid producing a plank line and then the repetition of the process for multiple plank lines

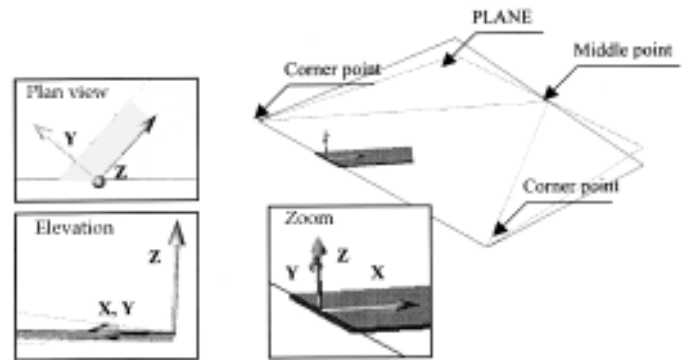


Figure 9. Definition of local coordinate system for sub-line.

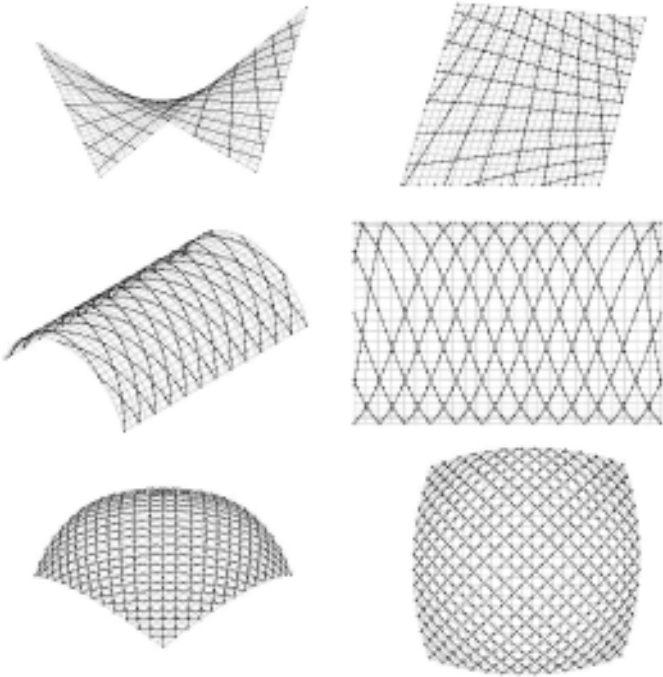


Figure 10. Sample output from the program for hyperbolic-paraboloid, barrel vault, and spherical shapes.