The Dissertation Committee for Jinman Kim Certifies that this is the approved version of the following dissertation:

PARAMETRIC DESIGN METHODOLOGY AND VISUALIZATION FOR SINGLE CURVATURE TENSEGRITY STRUCTURES

Committee:

Liapi, Katherine A., Supervisor

Haas, Carl T.

Tassoulas, John L.

Siegel, Jeffrey A.

Speck, Lawrence W.

PARAMETRIC DESIGN METHODOLOGY AND VISUALIZATION FOR SINGLE CURVATURE TENSEGRITY STRUCTURES

by

Jinman Kim, B.S., M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas at Austin

August, 2004

Dedication

To my lovely wife and parents,

Sukyoung Cho, Byung Kyu Kim, and Won-yi Kang

Acknowledgements

First and foremost, I would like to deeply thank my supervisor Dr. Katherine Liapi for her advice, guidance and inspiration to my research in completing this dissertation.

In addition, I am very grateful to all the commitee members, Dr. Carl T. Haas, Dr. John L. Tassoulas, Dr. Jeffrey A. Siegel, and Professor Lawrence W. Speck for their time and all the help.

I am grateful to Jong chul Song, my colleague, for his help and friendship. I am also wish to convey my gratitute and appreciation to my parents and my wife Suk Young Cho for her support.

Jinman Kim

The University of Texas at Austin

August 2004

PARAMETRIC DESIGN METHODOLOGY AND VISUALIZATION FOR SINGLE-CURVATURE TENSEGRITY STRUCTURES

Publication No._____

Jinman Kim, Ph.D.

The University of Texas at Austin, 2004

Supervisor: Katherine A. Liapi

Tensegrity structures are a special type of tensile structures consisting of cables (in tension) and bars (in compression) that can offer an alternative to conventional space covering structures. Geometric complexity inherent to these structures has posed a significant challenge in their geometric and structural design and limited their applications in buildings. This research is intended to develop a parametric design methodology for single-curvature tensegrity networks to address problems in their configuration and analysis. An important feature of the methodology is the development of an integrative visualization environment to assist in their form exploration and performance.

The methodology involves a) the development of algorithms to address the geometry of vaulted configurations that generate models of initial geometry b) integrating design algorithms to structural analysis and development of models of pre-stressed geometry, and c) importing the pre-stressed geometry model into a CAD environment.

Specifically, 3D coordinates of a preliminary tensegrity structure are generated by the design algorithms, automatically processed by an existing analysis code, and visualized in CAD environment by the graphical interface. Resulting 3D solid models of the structure can then be used by architects and engineers to validate the design performance of preliminary configurations under consideration. The morphological variation considered in this study is that of vaulted configuration composed of tensegrity units of square-base with bar to cable connection.

Table of Contents

List of Tables

List of Figures

CHAPTER 1: INTRODUCTION

Tensegrity, a contraction word of "tensional integrity" (Fuller, 1962), refers to a special type of tensile structures that can offer an alternative to traditional space frame structures. A tensegrity structure is also defined as "a set of discontinuous compressive components interacting with a set of continuous tensile components that define a stable volume in space" (Pugh, 1976). Because tensegrity structures consist of continuous tension members, cables, and discontinuous compression members, bars, that form a system in which forces are in pure tension and compression where no bending or twisting occurs, their shape can be optimized with a minimal number of compressive bars; this implies that tensegrity structures, in principle, can be lightweight but strong due to the characteristic material properties of cables.

Tensegrity structures are also defined as free-standing self-tensioning structures (Emmerich, 1988; Hanaor, 1993) which are characterized by large displacements. In the last two decades several methods for the analysis of tensegrity structures have been developed. Yet, despite recent advancements in the analysis and technology of tensegrity structures, topological and geometric complexity, the lack of a standardized engineering method for their design, and difficulty in visualization remain problems that have been only partially addressed and which hinder their full scale application in the building industry

An important feature of tensegrity structures is that they are potentially deployable (Hanaor, 1993); this property has also suggested their application to the deployable building industry. Indeed methods for deploying tensegrity structures for building design have been addressed by several pioneer researchers in the field. A small

scale model of an inflatable tensegrity structure was already developed in the early nineties (Hanaor 92), while several deployment studies have already been conducted at the Centre de Tensegrite of the Monpelier University (Motro, 1993). At the University of Texas at Austin, a novel method for the rapid assembly of tensegrity structures composed of deployable tensegrity modules, that is based on the invention of a prototype tensegrity module, has been developed and experimentally tested with half and full scale models. (Liapi, 2002b) (Figures 1-1 and 1-2). A basic feature of the method that can find applications in both permanent and ephemeral structures is that it allows for various geometrical configurations, including but not limited, to spherical and vaulted shapes, which can be obtained easily by adjusting the manner in which the deployable units are attached to each other.

Figure 1-1: Full scale experimental model composed of 16 units (designed and built by K. A. Liapi)

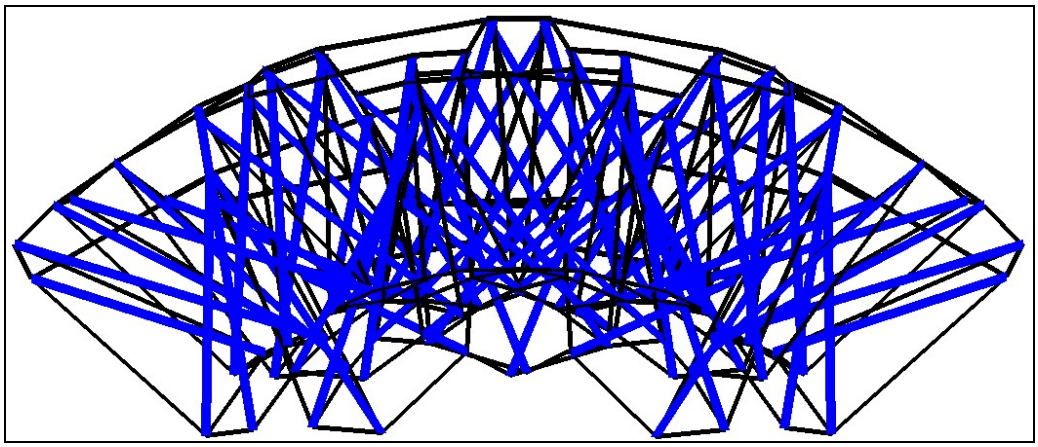


Figure 1-2: Computer visualized model of the previous structure (virtual model developed by F. S. Kazi)

Earlier research work at the University of Texas at Austin has also been focused on the development of methods for the generation of the initial geometry of tensegrity structures that are based on specific patterns of unit connection by following constructive geometry processes (Liapi, 2001a). The objective was to allow architects and engineers to study the effect of critical geometric parameters in the shape of these structures. Yet since any change in any one or combination of parameters affects the geometry of the whole, a tedious geometric construction procedure will have to be repeated for each change. However, although models of initial geometry generated in this manner provide a tool for form exploration during the preliminary design phase, the shape the structure acquires after pre-stress and loading are applied is also needed so that architects and engineers can finalize decisions about both the form and the loads the structure can carry. This further complicates the problem of form generation in tensegrity structures, since, in addition to the geometric parameters, the material properties of its members and the applied pre-stress need to be considered.

Hence, in order for tensegrity structures to be feasible and used in building design, a method that overcomes the above limitations needs to be developed. A parametric approach, that will take into account all geometric relationships and principles that apply to the geometric design of tensegrity structures and which will integrate an analysis software to determine the pre-stressed configuration would overcome most of the limitations designers face in the application of tensegrity in buildings. The most important advantage of implementing a parametric design method would be that new geometries will be generated automatically by changing only the numerical values of certain parameters rather than having to redesign the entire structure. At the same time architects and engineers would greatly benefit from the development of a design methodology that will offer to them the ability to visualize tensegrity models of initial and pre-stressed geometry.

1.1 RESEARCH OBJECTIVES

The primary objective of this research is to develop a methodology for the design and visualization of single-curvature tensegrity structures. To support the main objective, the research effort is intended:

- To develop algorithms that can generate parametric models of initial geometry for single-curvature configurations of tensegrity structures
- To develop methods for importing models of initial geometry into structural analysis
- To develop an integrative visualization environment that displays models of initial geometry and pre-stressed or loaded configurations, as well as stress flows in the structure

- To develop methods for importing models of tensegrity structures in a commercially available Computer Aided Design environment

1.2 RESEARCH SCOPE

This research is focused on geometric aspects of tensegrity structures including prestress geometry, loaded configuration, and visualization. Developing the interface between geometric design and analysis algorithms is part of this research, though a thorough study of the performance of tensegrity structures is beyond the scope of this study. Design and visualization methods developed in this research pertain to single-curvature tensegrity structures that consist of square-based units and which maintain bar independence throughout the structure. The analysis algorithms chosen for this study directly address the geometric features and methods of pre-stressing modular tensegrity structures. Only two instances of single-curvature networks are considered: a) vaulted and b) helicoidal. Numerical results obtained from the developed design methodology are not intended to be validated with experiments.

1.3 OUTLINE OF DISSERTATION

This dissertation contains seven chapters. Following this introduction, Chapter 2 explains general features of tensegrity structures including background and geometric configuration. The evolution of the tensegrity concept and the literature review of single and double-layer tensegrity structures are presented in this part. This Chapter also provides an overview of form-finding methods and commercially available software. A general review of deployable tensegrity structures is presented as well the features of the methodology to be followed in the geometric design and analysis of tensegrity structures.

In Chapter 3, the proposed methodology for the design of vaulted structures is presented along with a discussion on limitations of their geometric configuration. Two design algorithms, a) “unit-based parametric design algorithm” and b) “structure-based parametric design algorithm” are developed in order to address most design/construction of scenarios. Design parameters for each design algorithm are also discussed.

In Chapter 4, the design methodology for helicoid structures is developed and details of their geometric configuration and limitations, overlap conditions, and graphical representation are provided.

Chapter 5 focuses on the development of an integrative environment that allows for importing and displaying data from different applications (geometric design and structural analysis). Displacements and flow of stresses are visualized by the integrative environments.

Chapter 6 presents a method to visualize tensegrity structures in a CAD environment. A Microsoft Visual Basic for Applications (VBA) code which imports models of tensegrity structures in the MicroStation environment is developed and its implementation is presented.

In Chapter 7, conclusions of this research and recommendations for future works are presented.

CHAPTER 2: FEATURES OF TENSEGRITY STRUCTURES

2.1 BACKGROUND AND GEOMETRIC CONFIGURATION OF TENSEGRITY STRUCTURES

In the history of tensegrity, three people, Richard Buckminster Fuller, Kenneth Snelson, and David Georges Emmerich have significantly contributed to the invention of this new concept, the authorship of which has been claimed by all three. The artist Snelson built the first tensegrity structure, an X-shape sculptural object in 1948 (Figure 2-1) while still a student. Although Snelson built numerous tensegrity sculptures, Fuller was the first to look at tensegrity structures from an engineering point of view and he was the one who coined the word tensegrity, which is a contraction of tensional integrity. Fuller patented his idea in 1962 in the USA. Almost at the same, Emmerich in France, filed his own patent application in 1963, and described tensegrity systems as a “self-stressing system”. Snelson’s patent was granted in 1965 with the title “Continuous tension, discontinuous compression structures” (Snelson, 1965).

Fuller described tensegrity systems as "islands of compression in an ocean of tension" (Fuller, 1965). This description is vague and does not clearly define the limits of what could be called tensegrity. The following more precise definition was given by Pugh (Pugh 1976): “A tensegrity system is established when a set of discontinuous compressive components interacts with a set of continuous tensile components to define a stable volume in space” while Hanaor defined tensegrity systems as “internally pre-stressed, free-standing pin-jointed networks, in which the cables or tendons are tensioned against a system of bars or struts” (Hanaor, 1993).

Figure 2-1: Snelson's X-shape, 1948

Another broad definition was given by Pellegrino who described as tensegrity "any structure realized from cables and struts, to which a state of prestress is imposed that imparts tension to all cables" (Pellegrino, 2003), while at the same time he narrowed the definition by adding that "the state of prestress serves the purpose of stabilizing the structure, thus providing first-order stiffness to its infinitesimal mechanisms".

2.1.1 Single-layer Networks

In general, tensegrity structures can be classified according to the number of cable layers in their geometry. Single and double-layer structures are the two main classes. Single-layer structures were developed by Fuller and Vilnay. A tensegrity dome structure, introduced by Fuller in the context of his 1962 patent (Figure 2-2), consists of a high-order of complex polyhedral forms in which the tensile and compressive members do not touch each other. This dome is a highly material-efficient technology which, according to Vilnay, can be used for outer-pace applications. Theoretically, this tensegrity dome can cover a large span. However, in practical applications, in order to achieve a useful span, and to avoid members from coming in contact with each other, a low curvature is necessary. An improved single-layer tensegrity structure (Figure 2-3) developed by Vilnay manages to overcome the problem with members coming into contact with each other. Vilnay's structure was based on regular planar cable nets that produce curved surfaces after inserting bars of unequal length in between the cells of the cable nets. By following Vilnay's method some large span structures can be achieved easily by increasing bar length, thus requiring large cross section to prevent buckling (Vilnay, 1981). Vilnay's method cannot be used for low curvature or planar structures due to the fast increasing self-weight of the network. In this method compressive and tensile forces are strictly segregated among the members. Figure 2-3 shows Vilnay's single-layer tensegrity structures (Vilnay, 1981).

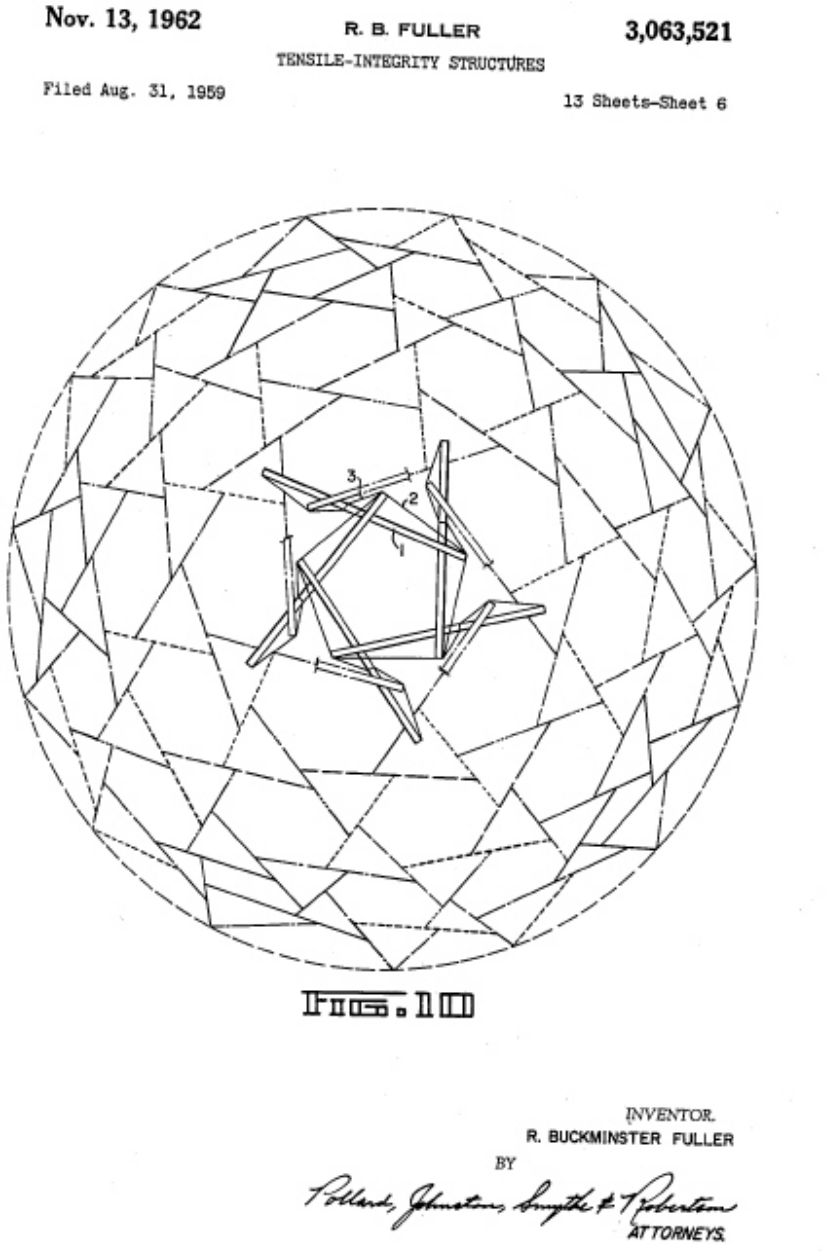


Figure 2-2: Fuller's tensegrity dome (U.S. Patent, 3063521, 1962)

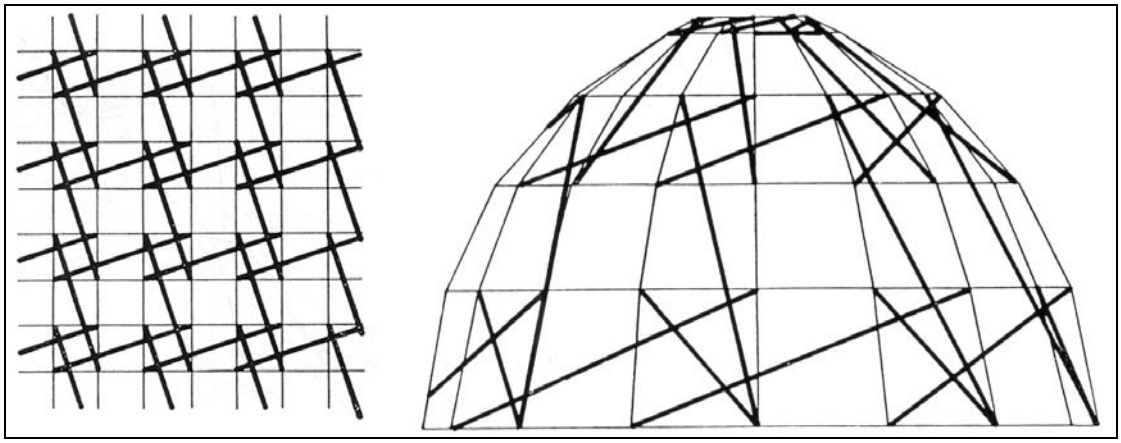


Figure 2-3: Vilnay's single-layer tensegrity dome

Pugh classified single-layer structures based on the connection pattern between cables and bars (Pugh, 1976). Four different patterns have been identified: diamond, circuit, geodesic, and zigzag as shown in Figure 2-4 and 2-5 (Pugh 1976).

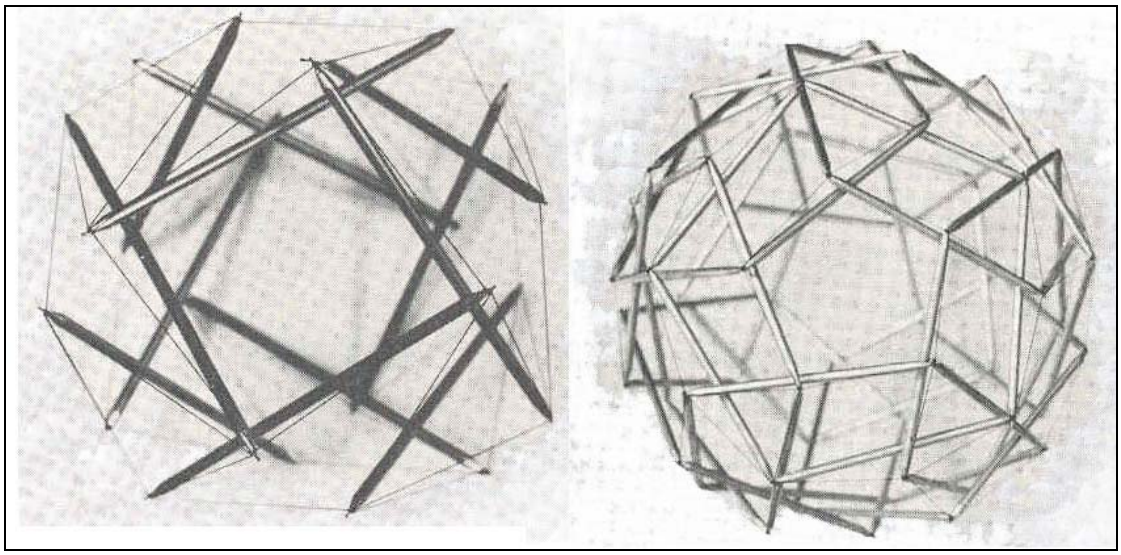


Figure 2-4: Left: Diamond-pattern with a twelve-strut three-layer, Right: Circuit-pattern with a small thombicosidodecahedron

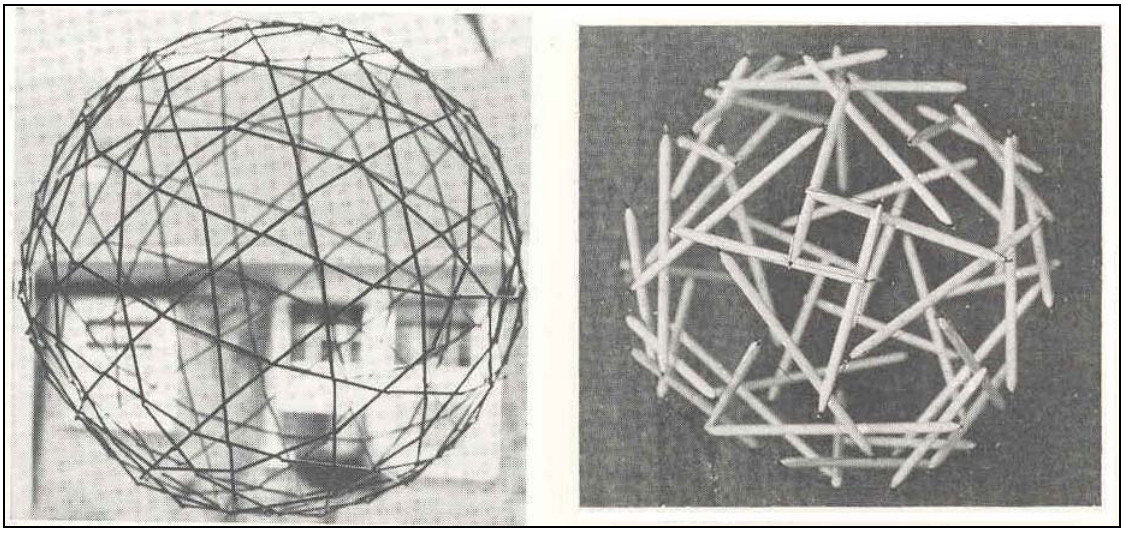


Figure 2-5: Left: Geodesic pattern with a six-frequency icosahedron based on the geodesic polyhedra, Right: zigzag-pattern with a six-frequency octahedron

The definition of the "diamond" pattern (or otherwise called "rhombic system") by Pugh is as follows: "Each strut of a "rhombus system" constitutes the longest diagonal of a rhombus of cables, folded according to this axis" (Figure 2-6).

"Circuit" patterns consist of four edges at the ends of two struts are joined as shown in Figure 2-7. Circuit patterns can be also derived from the diamond pattern by increasing the number of the elements they connect. The circuit patterns are less flexible than the diamond patterns.

About the "zigzag" pattern Pugh observes: "between the two extremities of each strut there exists a totality of 3 non aligned cables" which seem to form a "Z" as shown in Figure 2-8. Many interesting configurations can be achieved by applying these single-layer tensegrity patterns: however the construction of single-layer tensegrity domes requires very tedious work due to null stiffness on members.

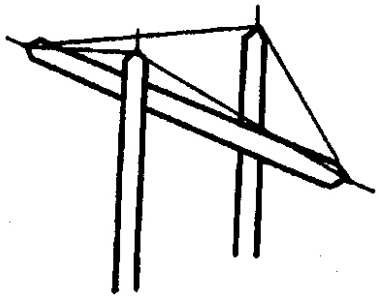


Figure 2-6: Principle of diamond pattern

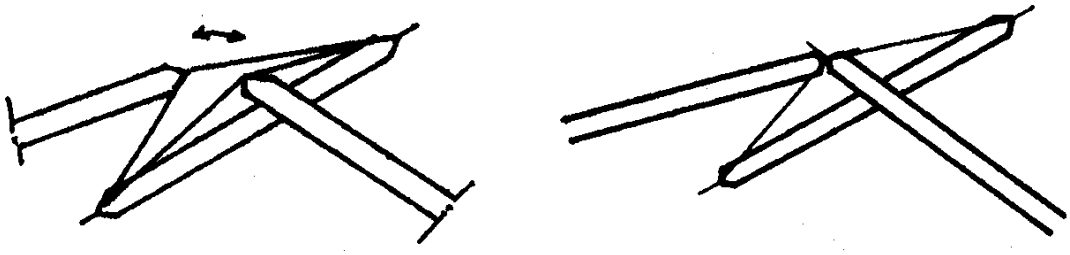


Figure 2-7: Principle of circuit pattern

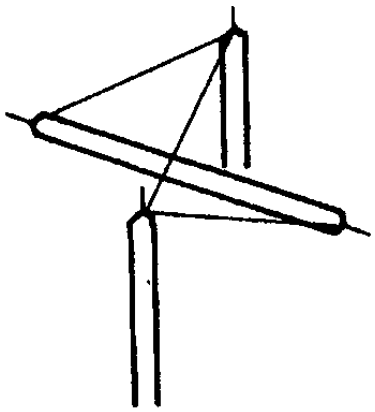


Figure 2-8: Principle of zigzag pattern

2.1.2 Double-layer Networks

Double-layer tensegrity structures occur from the assembly of self-stabilized tensegrity units. Regular tensegrity units have all bars confined between two parallel cable bases. There are two types of tensegrity units: a) a tensegrity prism called T-prism and b) a truncated pyramid called T-pyramid. T-prisms were patented by Emmerich in 1963. Figure 2-9 shows plan and isometric views of triangular, square, and pentagonal T-prism units.

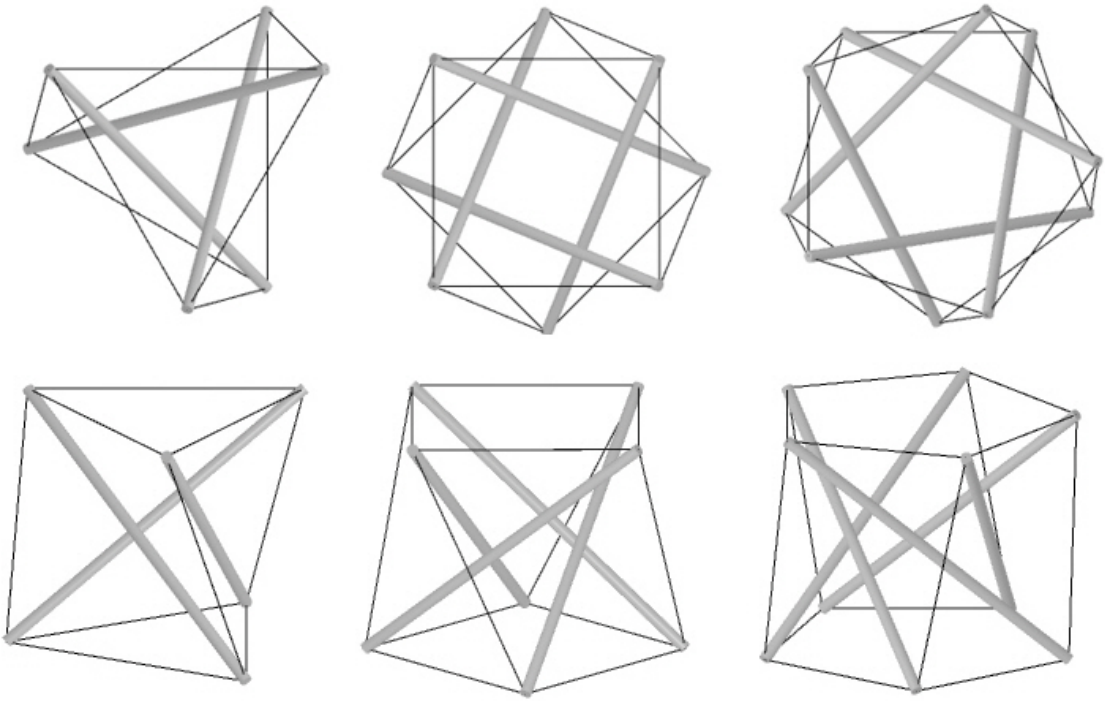


Figure 2.9: Examples of tensegrity prisms

The polygonal bases of both types are parallel to each other and rotated at a certain unique angle which is constant and depends on the shape of their polygonal base.

The twist angle between the two polygonal bases can be calculated by the equation below developed by Kenner (Kenner, 1976).

Twist angle $\alpha = 90^\circ - \frac{180^\circ}{n}$ where n is number of sides in the upper or lower polygons

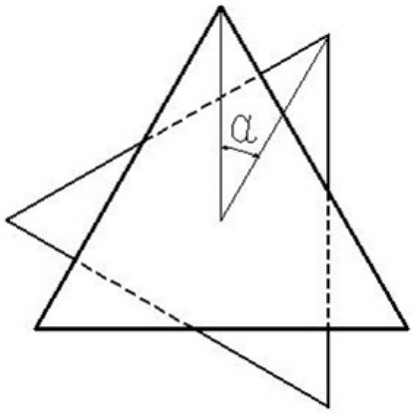


Figure 2.10: Twist angle of a triangular base unit

Figure 2-11 shows the cylindrical coordinates that were used to calculate the twist angle where bar is indicated with *s*, side tendon with *t*, and end tendon with *e*. The twist angle α was calculated geometrically (Kenner, 1976) and it was shown that the twist angle between the polygonal bases of a T-unit is half the size of the angle of the base of the polygon; accordingly this angle is 30° for a triangular base, 45° for a square-base, and 54° for a pentagonal base.

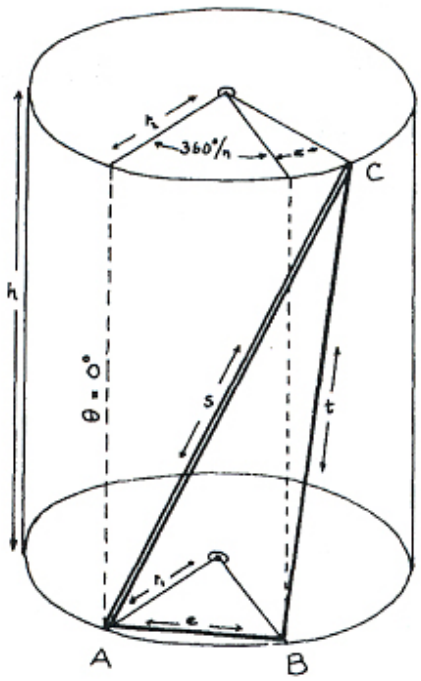


Figure 2.11: Cylindrical coordinates of twist angle

Stern (Stern, 1999) conducted a static analysis of the internal forces and developed a design equation that calculates the lengths of the struts and elastic ties on n-strut tensegrity systems. This provides relationships between the internal forces and identifies geometric patterns which are correlated with Kenner's twist angle. Detailed equations are developed in Stern's thesis (Stern, 1999).

Since the size of the upper and lower base of a unit is the same in T-prisms but different in T-pyramids, flat structures can be generated by combining T-prisms while T-pyramids are more appropriate to form curved structures as shown in Figure 2-12 (Hanaor, 1992).

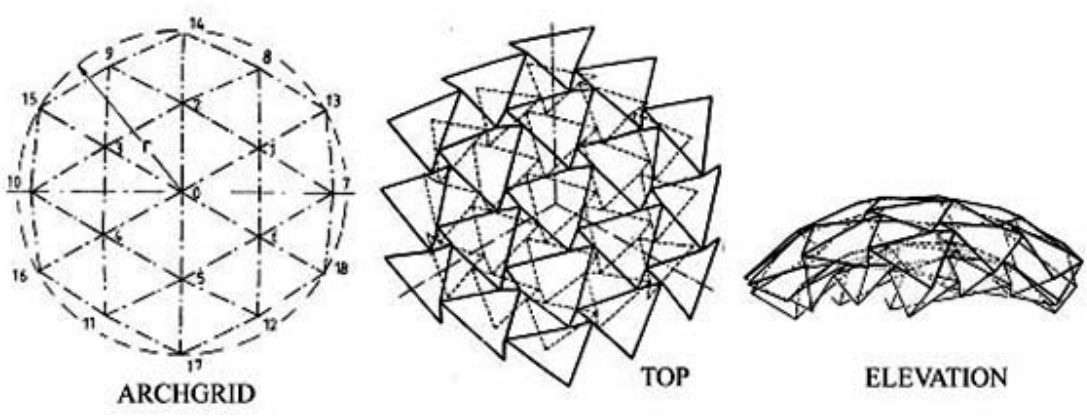


Figure 2-12. Hanaor's double-layer tensegrity dome composed of truncated pyramids

Tensegrity structures can be joined in several ways to generate double-layer tensegrity grids (DLTGs). Wang categorized tensegrity structures into two configurations: “non-contiguous strut configuration” and “contiguous strut configuration” (Wang, 1998). Three connecting methods (Ia, Ib, II) have been proposed by Hanaor (Hanaor, 1987) to form non-contiguous strut configuration (Figure 2-13).

Connecting methods Ia and Ib are vertex-to-edge connections of tensegrity systems in which the triangular bases are connected consecutively (Figure 2-13:Ia, Ib). The vertices in method Ia and Ib are connected to an edge of the adjacent unit. The method Ia is feasible when the composing units have polygonal bases with an odd number of sides. The upper layer in method Ib is parallel to the lower layer of adjacent units and feasible when the bases of the composing units are polygons with an even number of sides. The method II (Figure 2-13:II) requires an edge-to-edge connection.

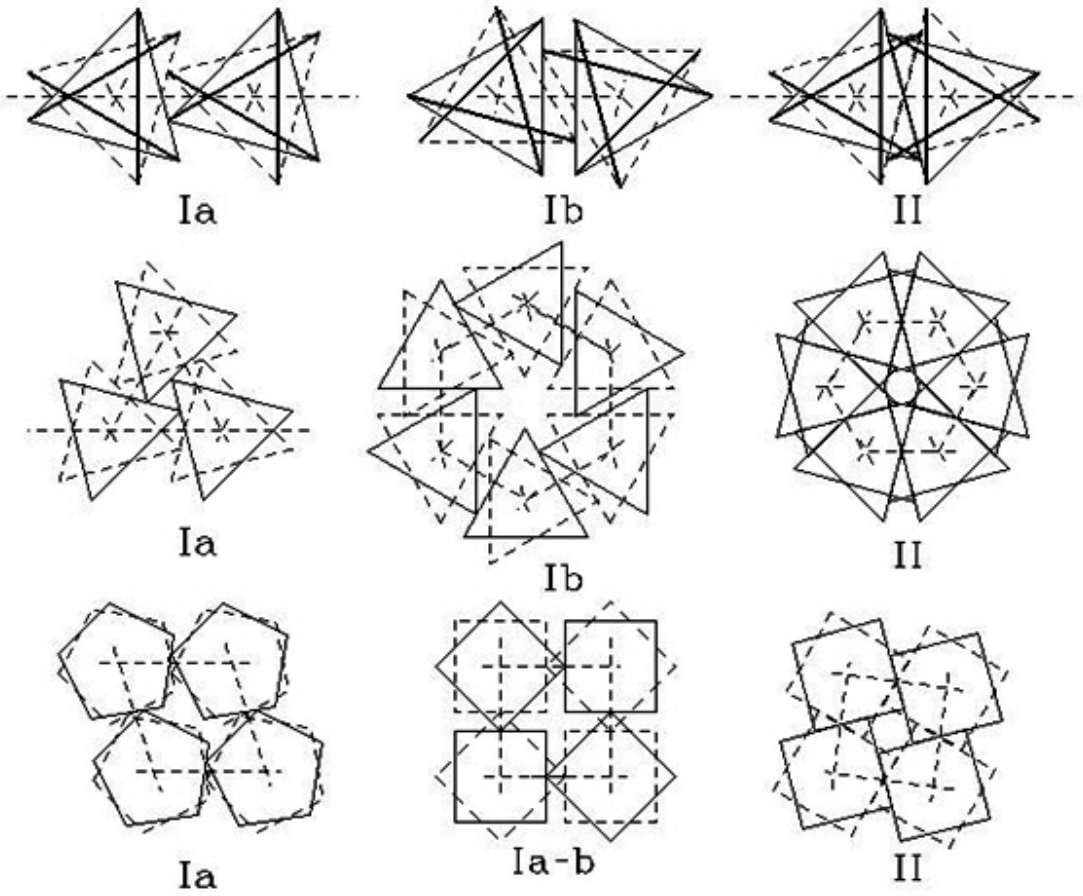


Figure 2-13: Non-contiguous strut configuration by Hanaor

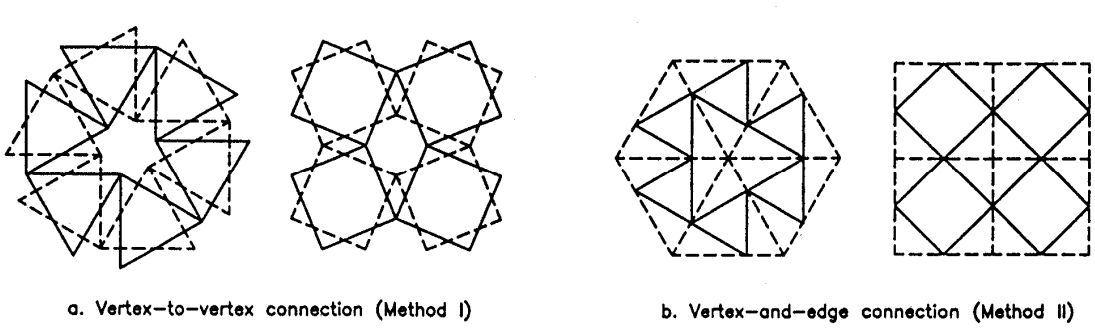


Figure 2-14: Contiguous strut configurations by Hanaor

In contiguous strut configurations, two types of connection methods can be used (Figure 2-14): a) vertex-to-vertex and b) vertex-and-edge connection. The concept of the contiguous configurations has been developed by Motro (Motro, 1990). Instead of connecting units together by vertex-and-edge connections, Motro showed the possibility of connecting units together by vertex-to-vertex connections. He suggested the use of square-based-prisms based on the half cuboctahedron geometry (Figure 2-15). This method offers simplicity in the construction of the grid and the visualization of the structure while the cuboctahedron was found to distribute stresses evenly under load (Motro, 1987 and 1990).

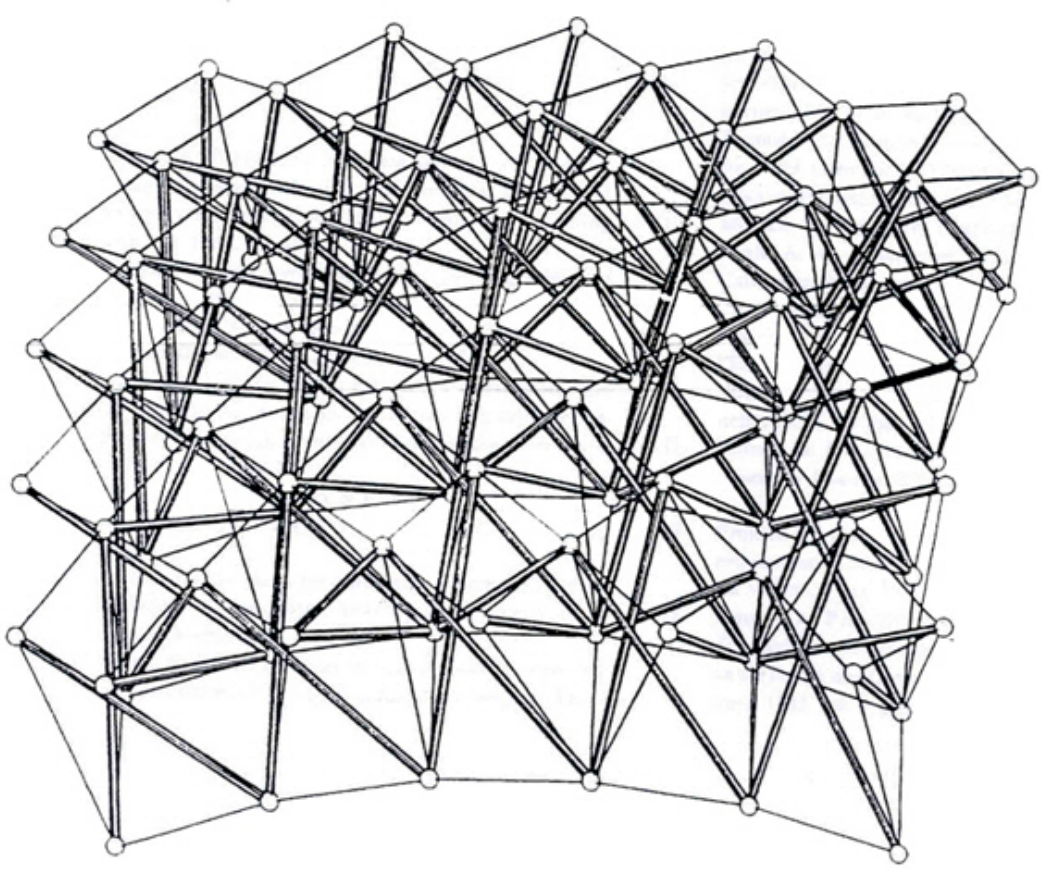


Figure 2-15: Double-layer tensegrity grid with bar-to-bar connection by Motro

At the University of Texas at Austin, Liapi, based on Hanaor's concept of unit-connection, proposes curved configurations from identical square-based tensegrity units. The morphological variation of a square rather than a triangular base proposed by Liapi permits the development of both single and double-curvature structures (Figures 2-16, 2-17).

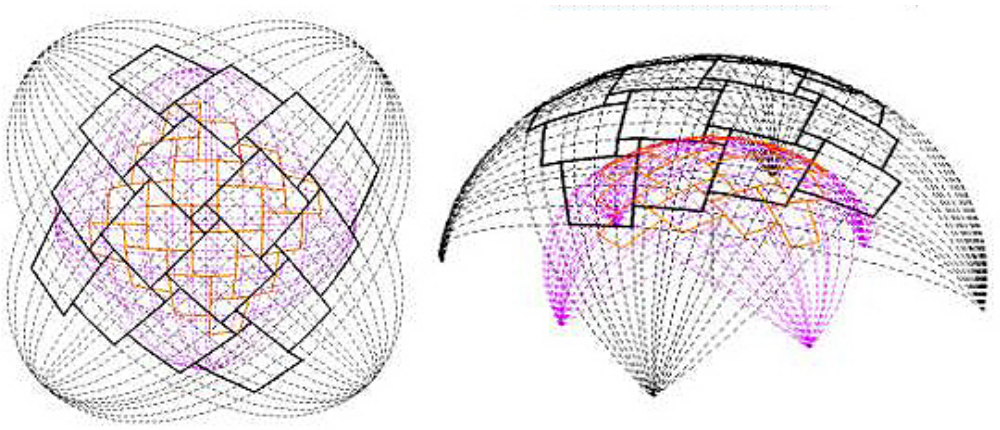


Figure 2-16: Digital models of a tensegrity spherical structure composed of 16 tensegrity units

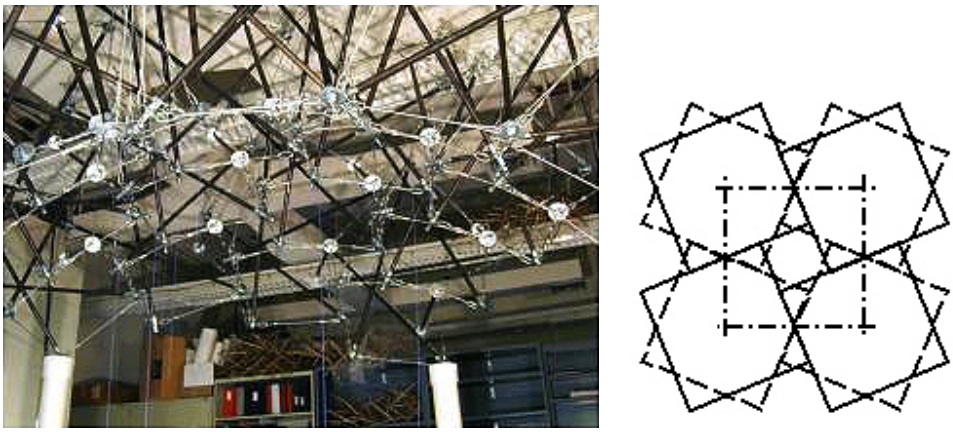


Figure 2-17: Physical model of tensegrity structure and connection pattern

2.2 DEPLOYABLE TENSEGRITY STRUCTURES

Deployable tensegrity structures have been studied during the last 20 years. Deployment mechanisms are classified based on the method applied for modifying the lengths of certain members. Elongation or shortening of the struts, “strut mode”, or of the cables, “cable mode” or adjusting both struts and cables together, “mixed mode” are the most common methods.

In the cable method, the lengths of struts are kept fixed while the lengths of the appropriately chosen cables, “control cables”, are modified. In general, the collapsed geometry could be determined by the choice of “control cables” which are limited to only a few cables that run through the structure. In general, after collapsing the structure, all struts and cables need to be carefully aligned, otherwise the collapsed materials tend to get entangled each other due to the loose cables.

In the strut mode, the lengths of all cables are kept fixed while the lengths of the struts are changed. Hanaor used the strut mode for the development of a deployable structure in which the lengths of the struts were controlled by a hydraulic or pneumatic power (Hanaor, 1993). Hanaor also pointed out that in principle, the concept of deployable structure appears to be simple but for actual application several technical problems should be overcome.

A recently developed deployable double-layer tensegrity system at the University of Texas at Austin, which can be easily deployed, retracted and redeployed in another location addresses some of the technical problems identified in earlier researcher works. The main features of this system are: a) modular on site assembly, b) deployability at the unit level, c) re-usability of modules for the development of various geometric configurations and d) efficient assembly/erection method without the need for heavy

equipment (Liapi, 2002b). This study applies to the system developed at the University of Texas at Austin, but can be expanded to other applications.

2.3 FORM-FINDING METHODS FOR TENSEGRITY STRUCTURES

Form-finding is a required and critical process in the design of tensegrity structures by which their geometric shape in space is determined. Early investigations of tensegrity form-finding were conducted by Snelson (Snelson, 1965) and Emmerich (Emmerich, 1988), who developed heuristic approaches for deriving new configurations. Such methods often yield inaccurate results since they do not take into account the effect of pre-stress in determining the final configuration.

Form-finding methods for tensegrity structures that take into account their mechanical behavior, such as self-equilibrium, static or dynamic behavior are reviewed and broken into two major categories as identified by Pellegrino: statical approaches and kinematical approaches (Pellegrino, 2003). Statical methods include an analytical method, the "force density" method, and an energy-minimization method. Kinematical methods include an analytical approach, a non-linear optimization, and a pseudo-dynamic iteration. Form-finding methods are also divided into two major categories called "form controlled methods" and "force controlled methods" by Motro (Motro 2003). A detailed explanation of "form controlled methods" are discussed in the book "*Tensegrity: Structural systems for the future*" (Motro, 2003). In this chapter, major numerical form-finding methods are reviewed in detail.

2.3.1 Numerical Static Methods

The main feature of these approaches is the development of techniques for setting up the equilibrium equation between node displacements and member forces. Four

methods are identified (Pellegrino, 2003): an analytical method, force density method, energy method, and reduced coordinates method.

• Analytical method

The analytical method sets up the equilibrium equation without external forces at each node point. The equilibrium equation, then, can be used to find the value of the twist angle to meet the equilibrium state. This approach is simple because it is based on the regular geometric configurations to identify the geometry of simple tensegrity units but there are limitations when non-regular geometries or higher order units are to be calculated. Kenner (Kenner, 1976) used this method to find the configuration of a six-strut octahedron

• Force density method

Force density method is one of the statical methods already used for the form-finding of tensile structures. This method is based on the force density coefficient defined as q_{ij}, which is the normal stress T_{ij} divided by length l_{ij}. Motro used this method for form-finding of the tensegrity systems (Vassart and Motro, 1999).

$$q_{ij} = \frac{T_{ij}}{l_{ij}}$$

A simple mathematical trick is used in this method to transform a non-linear equilibrium equation into linear equations. The linearlized equilibrium equation at node *i* along the x-direction can be expressed as follows:

$$\sum_j \frac{T_{ij}}{l_{ij}}\left(x_i - x_j\right) = f_{ix}$$

where $\frac{T_{ij}}{l_{ij}}$ is the force density coefficient: q_{ij}, f_{ix} is the external load at node *i* along x- direction, then the above equation can be written as $\sum_j q_{ij}\left(x_i - x_j\right) = f_{ix}$

This nodal equilibrium equation can be written in matrix form as:

$$[C_S^T][Q][C_S]X_S = F_X$$

where $[C]$ is connection matrix, $[Q]$ is a diagonal force density matrix, X_S and F_X are column vectors of x-coordinates and external loads, respectively.

The equilibrium matrix equation can be solved for nodal coordinates.

• Energy method

The *energy method* was first devised by Connelly (Connelly,1993). The energy method follows two necessary steps: 1) satisfying nodal equilibrium, and 2) analyzing the stability of the structure by minimum potential energy creation. The nodal equilibrium condition is satisfied with the equation

$$\sum_j w_{ij} \cdot (P_i - P_j) = 0$$

where P denotes a configuration of n ordered points in d -dimensional space as

$$P = [P_1, P_2, ..., P_n]^T$$

Satisfying equilibrium alone is not enough to satisfy the stability of the tensegrity structure. In order to satisfy the stability of the structure, the basic principle of minimum potential energy must also be satisfied by the following energy equation

$$E(P) = \frac{1}{2}\sum_{ij} w_{ij} \left\| P_j - P_i \right\|^2$$

This condition shows that when an element is displaced (positive for cables, negative for bars), potential energy is built to the square of the displacement. Stability is achieved when the energy built up is minimized after the elements are displaced from their initial length. Connelly, by following this method, has successfully analyzed and catalogued many tensegrity structures that were previously unknown.

• Reduced coordinate method

Basically, this method uses the principle of virtual work and was introduced by Cornel Sultan for application in deployable space antennas (Sultan 1999). The virtual displacements of cable *j* can be expressed as

$$\delta l_j = \sum_{i=1}^{N} \frac{\partial l_j}{\partial q_i} \delta q_i$$

where δq_i is the virtual displacement of independent generalized coordinates and is a number that indicates the degrees of freedom of the system.

This equation can be expressed in matrix form for all cables as

$$\delta \mathbf{l} = \mathbf{A}^T \delta \mathbf{g}$$

where $\mathbf{A}$ is $((N \times M)$ matrix and defined as

$$A_{ij} = \frac{\partial l_j}{\partial q_i}$$

The virtual work for cable *j* is expressed as

$$V_j = T_j \cdot \delta l_j = T_j \sum_{i=1}^{N} \frac{\partial l_j}{\partial q_i} \delta q_i = T_j \sum_{i=1}^{N} A_{ij} \delta q_i$$

where T_j is internal force on member *j*.

By definition, the total virtual work should be zero for any virtual displacement δq_i, then, following the reduced equation is yielded

$A(q) \cdot T = 0$ and $T_j > 0$ for $j = 1, \ldots, E$ where *E* is the number of tendons

Because T_j is positive, $A(q)$ should be non-zero for a nontrivial solution. If $N = E$, the determination of $A(q)$ is zero, and if $N > E$, determination of $A^T(q)A(q)$ is also zero. Therefore, the condition $N < E$ guarantees $A(q)$ to be nonzero.

In general it is not easy to solve the pre-stressability problem and Sultan presents some cases which are solved analytically, or tried to reduce them to a problem in which the pre-stressability problem can be easier to solve.

2.3.2 Numerical Kinematical Methods

The main idea of these methods is to keep the lengths of the cables constant while increasing the lengths of the struts to reach their maximum, or to keep the strut length constant while the cable lengths are decreased until a minimum is reached. This method is closer to the way in which tensegrity structures are constructed. Analytical method and dynamic relaxation method are explained in detail.

• Analytical method

In regular simple structures, the twist angle α between upper and bottom polygon depends on the number of sides of the polygons and the struts that are connected to the corresponding vertices. Accordingly the geometry of the simple tensegrity, the vertices P_1 to P_5 (Figure 2-18) can be defined as:

$$P_1 = [R,0,0]$$

$$P_2 = [R\cos\theta, R\sin\theta, H]$$

$$P_3 = [R\cos\left(\theta + \frac{2\pi j}{v}\right), R\sin\left(\theta + \frac{2\pi j}{v}\right), H]$$

$$P_4 = [R\cos\left(\frac{2\pi}{v}\right), -R\sin\left(\frac{2\pi}{v}\right), 0]$$

$$P_{54} = [R\cos\left(\frac{2\pi}{v}\right), R\sin\left(\frac{2\pi}{v}\right), 0]$$

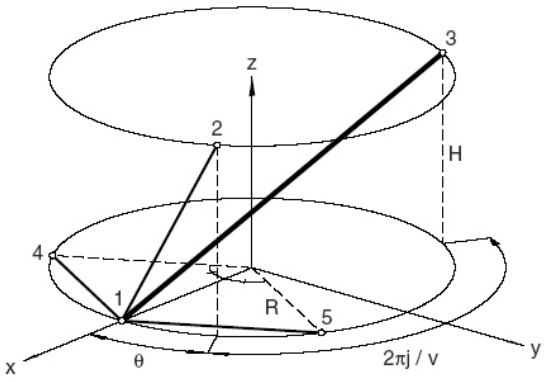


Figure 2-18: Geometric configuration of all members meeting at node 1

The following relationship can be found by squaring the lengths of the lateral cable, l_c, and strut, l_s.

$$l_c^2 = 2R^2(1-\cos\theta)+H^2$$

$$l_s^2 = 2R^2\left[1-\cos\left(\theta+\frac{2\pi j}{v}\right)\right]+H^2$$

By substituting the equation of lateral cable l_c into the equation of strut l_s, the two equations can be written as:

$$l_b^2 = 4R^2\sin\left(\alpha+\frac{\pi\cdot j}{n}\right)\sin\frac{\pi\cdot j}{n}+l_c^2$$

where R is the radius of the polygon, and j is an integer less than n. For a given cable l_c the strut length l_s is maximized for

$$\alpha = \pi\left(\frac{1}{2}-\frac{j}{n}\right)$$

This method is straightforward for regular symmetric polygons but the system of equations for non-symmetric structures can be infeasible because a large number of variables are required and their relationships are more complicated.

• Dynamic relaxation method

This method was used for the analysis of membrane and cable networks and was also applied to the form-finding of tensegrity structures by Belkacem (Belkacem, 1987). Dynamic equations are used to describe the structure and include the external load and damp as follows:

$$M\ddot{d}+C\ddot{d}+Kd=f$$

where M is the mass matrix, C is the viscous damping matrix, K is the stiffness matrix, and d is the vector of displacement, f is the vector of externally applied load, respectively. The mass matrix M and damping matrix C should be

diagonal matrices and, initially, the vector of velocity $\dot{d}$ and displacement vector d are set to zero. For the form-finding process, the iteration will be continued until $M\ddot{d} + C\dot{d}$ converges to zero, which means that equilibrium is achieved.

The dynamic relaxation method is good for simple and small number of regular structures; when the node numbers are increased or the structure is non-symmetric, this method is not effective.

2.4. COMMERCIALLY AVAILABLE SOFTWARE FOR GEOMETRIC CONFIGURATION, ANALYSIS AND VISUALIZATION

A significant problem in tensegrity structures is to define a method to generate the geometry of the structure. A review of commercially available software that resolves and displays the 3D geometry of complex structures has been conducted in order to determine their applicability in the configuration of tensegrity structures.

2.4.1 FORMIAN

"Formian algebra" developed by Hoshiar and Nooshin can be applied to find complex form in 3D structures. In particular Formian may be employed to generate information about various aspects of a structural system including element connectivity, nodal coordinates, loading particulars, joint numbers and support arrangements. The information generated may be used for the graphic visualization of the structural system or may even be submitted as input data to an analysis package (Nooshin,1991). A software package was developed for this purpose at the Space Structures Research Centre of the Department of Civil Engineering of the University of Surrey at Guildford UK.

Although the design of double-layer tensegrity systems is parametric, only a partial solution can be achieved by the use of Formian which is only limited to defining

the geometry of one single unit. The main reason behind this limitation of Formian algebra is that it is based on the node to node connection.

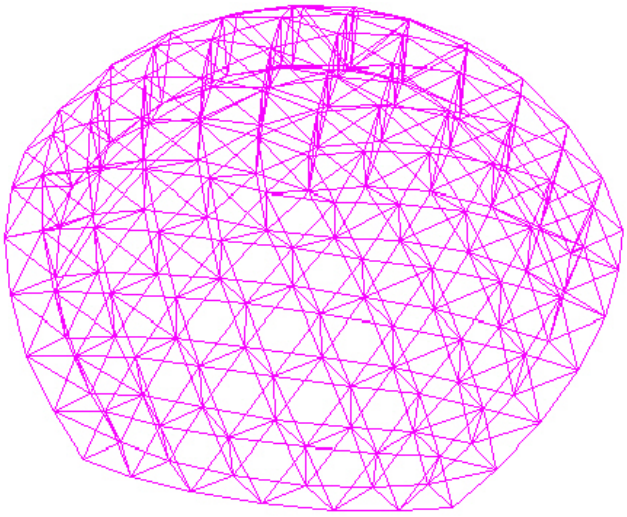


Figure 2-19: Configurations generated with Formian

2.4.2 TekCAD

TekCAD is relatively new software developed by TekStar International that can be used for the generation of the geometry of space structures. TekCAD is intended to provide an advanced computer aided mathematical system that is interoperable with industry-compatible formats in the architectural design area. TekCAD includes built-in mathematical statements and formula evaluation based on analytical functions to allow precise positioning and manipulation as well as generation of geometric forms.

To generate and manipulate various geometric forms, TekCAD uses simple hubs and struts that are analogous to vertices and edges or regular polyhedra. In addition, TekCAD has a powerful user friendly interface that allows a structure to be seen in multiple views. The working model can be manipulated fast and easily by operating a

rich set of functions and features. Sophisticated 3D structures can be created by directly generating new forms or modifying readily available forms. These structures can be also imported into or exported from industry-standard CAD packages via the DXF (Drawing eXchange Format), a public standard introduced by AutoDesk.

TekCAD may be well suited to the traditional 3D trusses that are represented by bar to bar connections, but it does not support bar to cable connections that are critical to represent tensegrity structures (www.tekcad.com).

2.4.3 Engineering Design Software that Integrate a 3D Graphical Display

Several commercially available for the analysis and visualization of space structures exist. Among them, SAP2000 presents advanced features and capabilities. SAP2000 is a Windows based integrated graphical package, developed by Professor E. Wilson and his associates in CSI (Computer and Structure Inc.) that allows for quick model creation using templates. SAP2000 is a nonlinear structural analysis program that includes 3D static and dynamic analysis and is based on the finite element method to calculate forces and reactions in beams and trusses and displacements. A wide range of general civil structures, including bridges, dams, tanks and buildings can be either created and modified from the pre-modeled structures or exported from DXF drawings generated from CAD software. The output of the analysis can be shown as graphical and analytical results displayed by selecting individual members or joints.

Element types supported by SAP2000 are Frame/Truss, Shell/Plate, Solid and Nonlinear Link. Each element supports a wide range of load types by controlling either static loading options or dynamic loading options. Static loading options allow for gravity, pressure, thermal and prestress conditions in addition to nodal loading with

specified forces or displacements. Dynamic loading can be in the form of multiple base response spectrums, or multiple time varying loads and base excitations.

Applying prestress to the members is a critical point in the analysis of tensegrity structures. SAP2000 and similar software supports prestress as a loading condition for the analysis, however, procedures to assign prestress require many steps for each member and are very time consuming processes.

A modular tensegrity structure consists of a small number of compression bars and many pre-stressed cables. For example, a square-based tensegrity module has 4 compression bars and 12 cables that require 12 iteration procedures to apply pre-stress on each member. Furthermore, the geometric form of tensegrity structures built by modular units is so complex that it is not easy to select the correct member by the computer mouse. Varying the applied pre-stress is of critical importance for the investigation of tensegrity structures since the stability of the structure and its stiffness depend on the applied pre-stress. There is therefore a need for a software application that will allow for applying pre-stress on selected cables. There is also a need for a graphical interface to display deformed configurations and the magnitude of applied forces.

2.5 METHODOLOGICAL APPROACH

2.5.1 Geometric Design: A Parametric Approach

As mentioned earlier, among critical issues in the design of tensegrity structures, geometric complexity accounts for significant difficulties in their study and for their limited application in building design. The review of currently available form-finding methods that take into account the mechanical behavior of tensegrity structures has indicated that these methods, though they yield accurate results can be hardly of use in

determining new or complex tensegrity geometry often involved in irregular or two-way arrangements of units that is typically the case with tensegrity roof structures.

To address this problem and to generate models of initial geometry of tensegrity structures for application in building structures composed of units of triangular shape, Hanaor has used a method of geodesic subdivision of the dome which has helped in providing models of initial geometry (Hanaor, 1992). However, very little documentation of the geometric processes that he followed can be found in bibliography. For deriving the tensegrity geometry that occurs from the assembly of tensegrity units of square-base, and to overcome the limitations of form-finding methods that integrate dynamic and statical properties, Liapi has developed a geometric approach that makes possible the investigation of the spatial configuration of tensegrity structures of spherical shape (Liapi, 2001a). Specifically, fundamental geometric principles that govern the generation of form in tensegrity structures are demonstrated. A method for the generation of spherical tensegrity structures, based on the integration of Euclidean Geometry procedures and CAD tools has been developed. This approach takes into account all of the interrelated parameters in the design of a tensegrity structure, such as proportions of units, overlap of upper and lower unit bases, etc, and proposes a step by step graphical procedure that follows constructive geometry methods. Liapi's approach is not devised for automatic generation of virtual models, so every time a change in one or more parameters occurs, the designer has to repeat this tedious process.

Parametric geometric design is a method to represent the geometry in terms of geometric design parameters. Regarding the geometry of tensegrity structures, an appropriate parametric geometric design would allow for changing one or more design parameters which would lead to the generation of either the entire structure or unit dimensions to meet updated specifications. Design parameters, therefore, need to be

expressed as mathematical expressions that take into account all interrelated parameters. The geometric configuration of modular curved tensegrity structures involves a high level of complexity, and the investigation of their morphology almost compels the development of several algorithms. The most critical advantage of implementing a parametric design method is that new geometries can be generated automatically by changing only the numerical values of certain parameters rather than having to redesign the entire structure.

Charalambides developed parametric computer method that provides numerical values of the initial geometry of tensegrity structure. This method generates parametric virtual models within a CAD environment. However, Charalambides' approach only provides numerical values for unit dimensions and overlap lengths, and does not provide 3D coordinates of the entire structure (Charalambides, 2004).

The parametric design is expected to allow for experimentation with many design variables. Specifically two geometric algorithms are expected to recreate the initial geometric design conception/process performed by an architect/engineer and to cover most scenarios that may be involved into the design of a tensegrity structure.

2.5.2 Method of Pre-stressed Design: Use of NONSA0

The review of form-finding methods and commercially available software has shown that none of these methods or software can be directly applied to tensegrity system considered in this study. It has also shown that the specific features of the system developed at the University of Texas at Austin, that is the proposed method of assembly and the method by which pre-stress is applied cannot be addressed by any of the available software. For this purpose, "NONSA0", a Nonlinear Structural Analysis computer program, developed by Tassoulas at the University of Texas at Austin, has been adapted

by its author to address tensegrity structures (Tassoulas, 2002 and 2003). The software takes also into account the method by which tensegrity structures are pre-stressed during assembly on site. NONSA0 is based on virtual work and direct stiffness principle using Newton equilibrium iterations. In this application, the estimated initial geometry is adjusted through NONSA0 until a pre-stressed form is computed by including the initial elongation of the bars and self-weight of members. NONSA0 can then calculate displacement and internal forces, as well as support reactions of the tensegrity structure subjected to external loads.

CHAPTER 3: VAULTED TENSEGRITY STRUCTURES

The morphological variation of double-layer tensegrity structures that is most likely to find application in building technology is the one that curves in one direction. Such structures, as already discussed in Chapter 2, can be generated from the assembly of triangular or square-based tensegrity units. The method of unit connection between units is that of "bar to cable" attachment. This study will only consider morphologies that occur from the assembly of square-based units and can support the deployable technology under development at the University of Texas at Austin (Liapi, 2002b). According to this, units can be connected by a partial overlap of their upper and lower base cables. By changing the amount of overlap on each or both bases the curvature of the structure in one or both directions can be changed. Certain overlap conditions can also lead to configurations of zero curvature (flat). As mentioned again in the previous section one of the features of this new technology is that the same units can be used for the generation of structures of various curvatures. The development of the parametric code will allow for the essay exploration of various configurations.

In general, structures with curvature can be of single or double-curvature depending on whether the structure curves in two or one direction only. Figures 3-2 and 3-3 show a 4-unit structure of single and double-curvature. In addition, a curved structure is further classified into uniformly or non-uniformly curved structure. A uniformly curved structure presents a constant curvature along one or two cross-sections. For example, vaulted (or cylindrical) and helicoid structures are structures of single-curvature, whereas the cylindrical can be also characterized as structure of uniform curvature while the helicoid falls within the non-uniform curvature category.

In tensegrity structures that occur from cable-bar assembly of square-base units, as mentioned in the previous section, determining the exact geometric form of the structure requires a very tedious and complex process, which depends on the synergetic effect of several interrelated geometric parameters. Any change in one of more of the geometric parameters affects the relationship of all other interrelated parameters and eventually affects the curvature of the structure (Liapi, 2001b).

The geometric principles and rules that apply to all regularly curved configurations have been developed in earlier research work, as well as a method for the generation of spherical configuration (Liapi, 2001a). In this chapter a new parametric design methodology that applies to uniformly curved configuration and which allows for the automatic generation of the geometry of such configurations is presented. The method requires the development of algorithms that integrate geometric principles that apply to curved networks from earlier research work as well as new principles and relationships developed during this study.

3.1 BASIC FEATURES OF THE GEOMETRIC PROCESS

A general geometric principle for configuring spherical and vaulted structures, as defined by Liapi, is that: given the unit size and upper-base overlap, the lower-base overlap, and the center of curvature can be found by rotating two adjacent units in opposite directions and around an axis that lies on the upper base's plane and which intersects the upper-base overlap (Figure 3-1) (Liapi, 2001).

A general rule that can be derived from the above is that, for the development of spherical configuration, the same amount of unit-rotation on both structure axes is needed while for the generation of cylindrical configurations a rotation on one axis only is needed, respectively adjacent units with overlapping upper and lower cables along the

cylinders axis should lie on a flat plane (Figures, 3-2 and 3-3). A detailed description of the geometric relationships in a vaulted structure and the features of the algorithms that were developed are discussed in the following section 3.2.

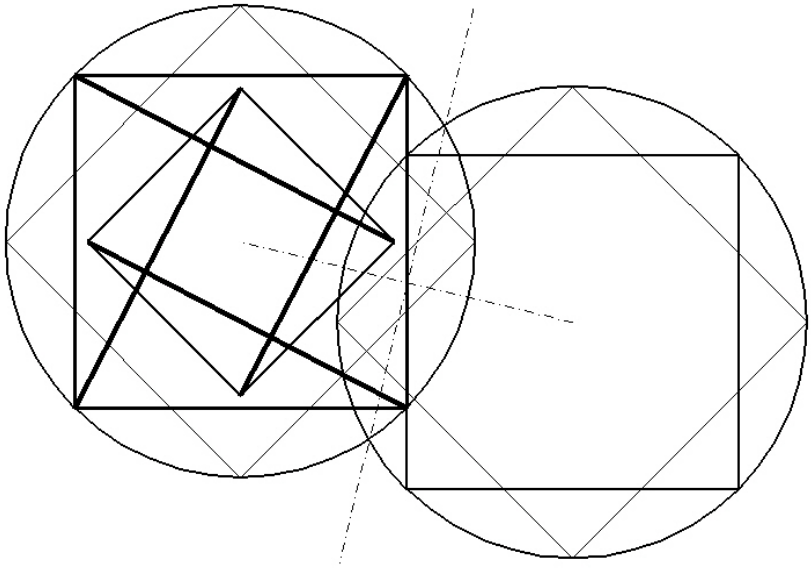
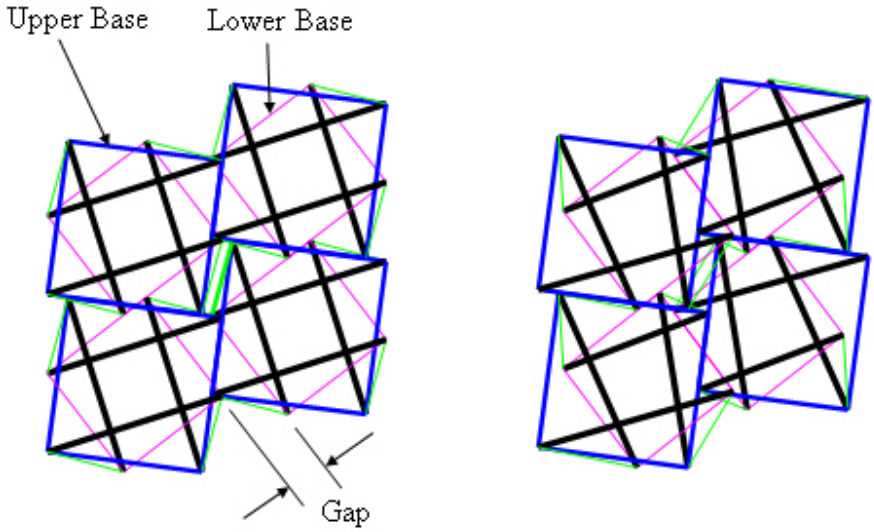
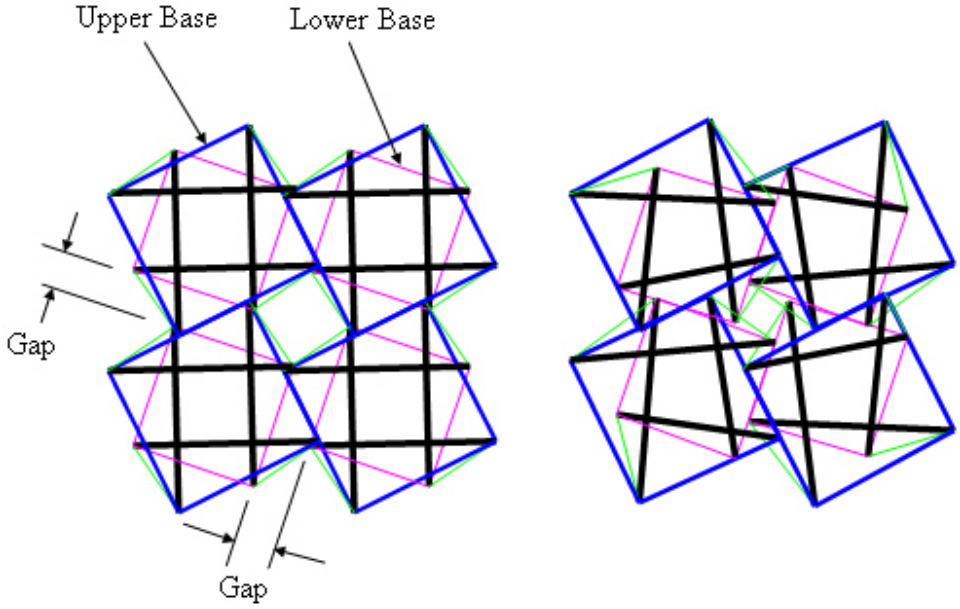


Figure 3-1: Overlap conditions for adjacent units that apply to all curved tensegrity structures



(a) (b)

Figure 3-2: 4-unit configuration of a single-curvature structure in a top view; (a) before unit rotation and (b) after rotation



(a) (b)

Figure 3-3: 4-unit configuration of a double-curvature structure in a top view; (a) before unit rotation and (b) after rotation

3.2 GEOMETRIC CONFIGURATION OF VAULTED TENSEGRITY STRUCTURES

Vaulted tensegrity structures discussed here are networks of single and uniform curvature. A condition that applies to single-curvature networks is that once the upper bases meet, the lower bases of adjacent units along the cylinder's axis should also meet so that no rotation along that axis will occur. Due to this characteristic, the center points of tensegrity units are shifted with respect to the line that connects the centers of the two units (Figure 3-2). After shifting, the line connecting two center points is no longer parallel to the plane of the circumference as shown in Figure 3-4.

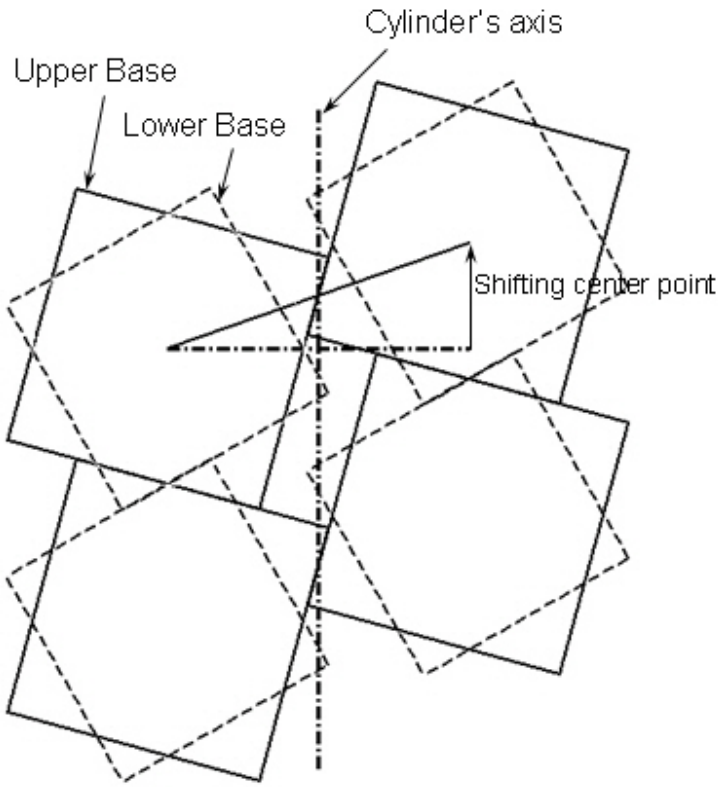


Figure 3-4: Top view of a 4-unit structure after shifting center points

To illustrate the process of creating a vaulted structure, a flat configuration prior to rotation, shown in Figure 3-2 (a), is considered first. The upper bases of adjacent tensegrity units need to overlap in both the direction of the cylinder's axis and along the circumference. The lower bases are arranged with an overlap in the direction of the cylinder's axis but with a distance between the bases in the direction of the circumference's axis. This distance from here on will be named "gap". The size of the gap determines the curvature of the structure. Note that during the process, the amount of overlap between the bases of adjacent units along the cylinder's axis does not change. The amounts of upper and lower overlap depend only on the upper and lower bases size.

Figure 3-5 shows a six-unit cluster of a vaulted tensegrity structure with the overlaps as indicated (only upper and lower cables are shown). It can be seen from the scheme that for a given unit size, upper and lower-base overlaps of adjacent units along the cylinder's axis are fixed. On the other hand, for a given upper-base overlap, two adjacent units along the circumference are rotated around an axis that passes through the mid point of the overlap so that their lower bases can meet. As one unit rotates relative to the other, the curvature of the resulting vault is determined solely by the amount of upper bases' overlap along the cylinder's axis.

As illustrated in Figure 3-7, two structures differing only in the overlap along the cylinder's axis end up with different curvatures. The structures shown in Figure 3-7 have a 35% and 50% overlap respectively, but both have the same upper base length, 2.5m, the same lower base length, 2.3m, and the unit height of the tensegrity in both case is 1.2m. Figure 3-8 shows the relative position of a 4-unit of Figure 3-5 after rotation, and the plan of a vaulted structure assembled from a 24-unit (Liapi, Kim 2003).

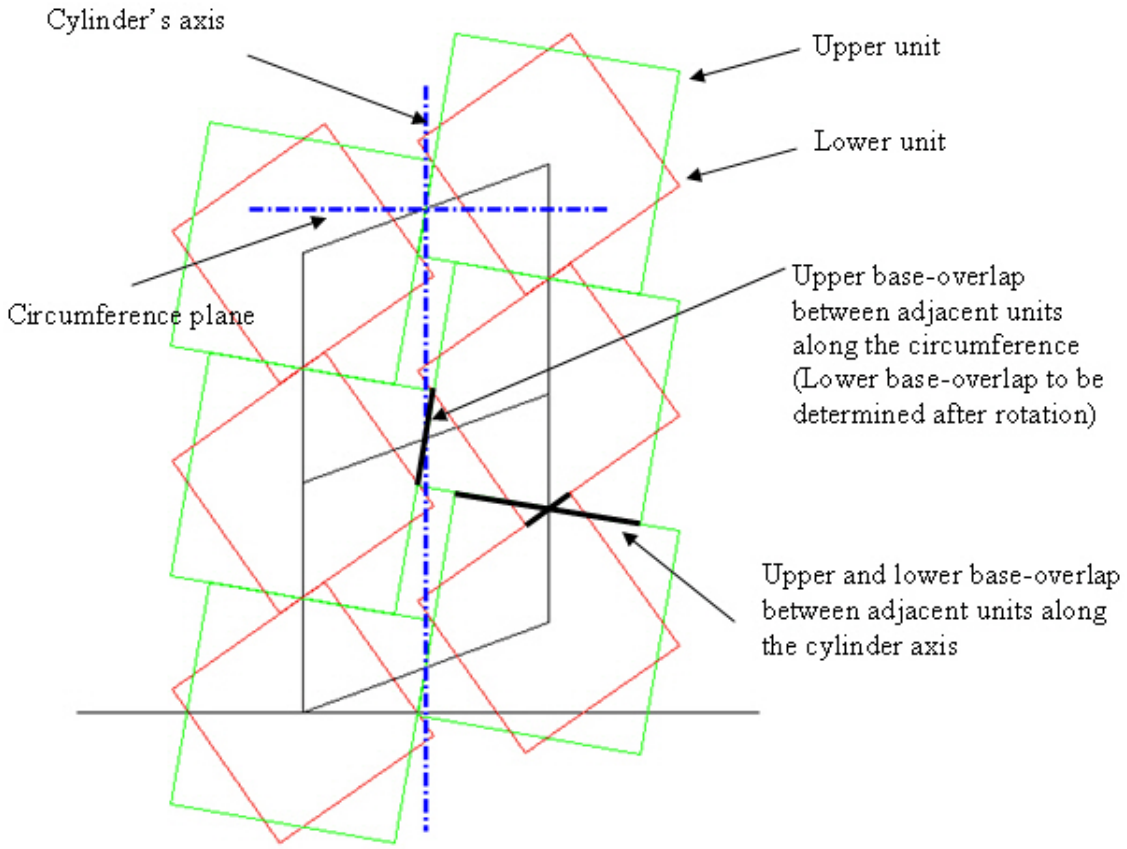


Figure 3-5: Overlap conditions for vaulted tensegrity structures

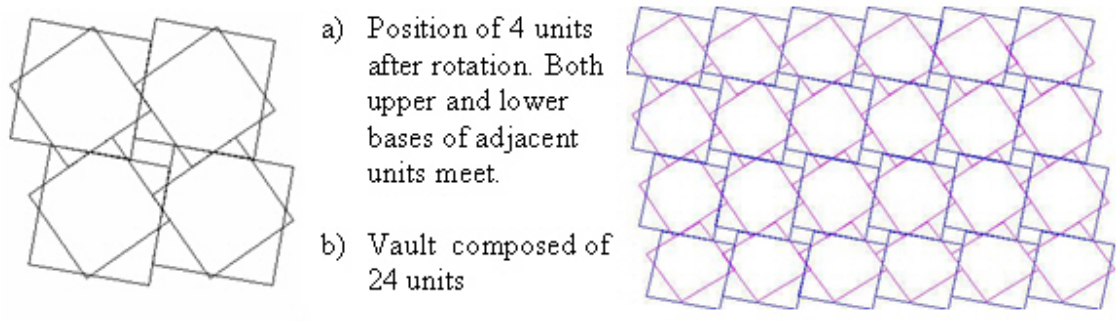
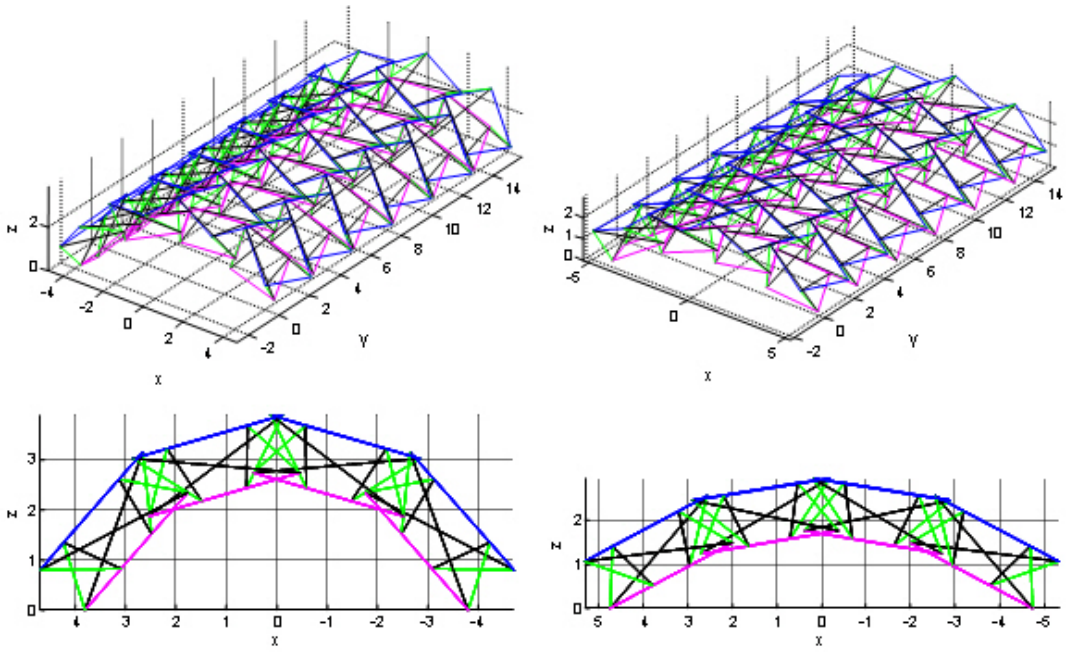


Figure 3-6: Plan view of a 4-unit assembly and a 24-unit vaulted structures



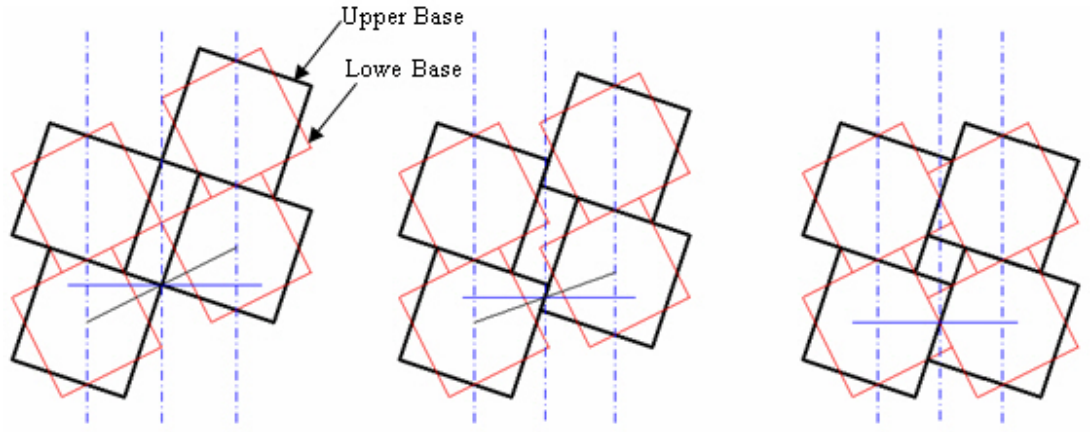
(a) 35% upper-base overlap. (b) 50% upper-base overlap.

Figure 3-7: Curvatures generated from different upper-base overlap values

3.3 GEOMETRIC LIMITATIONS

The above overlap relationship between units places limitations on the proportional relationship between upper and lower bases. This means that with a given size of upper base, only a certain size of the lower base of each unit is allowed to assure an overlap between adjacent units, as shown in Figure 3-9. If a vaulted tensegrity structure is to be generated but the lower base length is smaller than the minimum length, then the lower-base overlap along the cylinder axis does not meet, means that a single-curvature cannot be created. If the lower base length is greater than the maximum length, the overlap of upper base cannot happen and a "bar-to-cable" connection is not possible. Figure 3-8 shows the theoretical geometric configuration for the maximum curvature and the zero curvature structure. As shown in Figure 3-8 (a), the maximum

curvature can occur when the upper bases meet at a point of zero overlap, though it depends not only on the proportion of upper and lower bases but also on unit height. Based on this observation, it derives that by increasing the upper to lower base ratio, the curvature of the structures increases.



(a) Maximum curvature (b) Zero curvature

Figure 3-8: Geometric configuration for maximum and zero curvatures

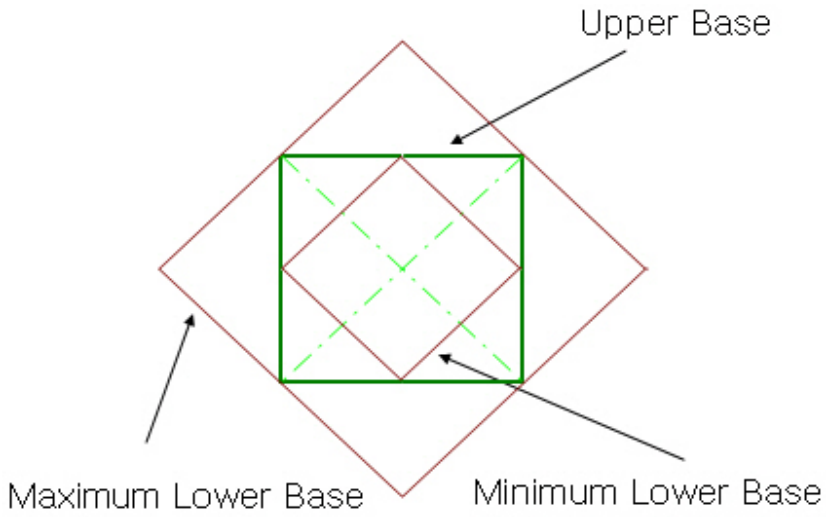


Figure 3-9: Limitation of lower base based on the given upper base

Given the upper base length, the minimum and maximum of the lower base length can be calculated by observing this overlap relationship of single-curvature structures. Let l_U and l_L denote the upper length and the lower length, respectively. Then the minimum and maximum lengths of lower bases are given by:

- Minimum length of lower base = $\frac{l_U}{\sqrt{2}}$
- Maximum length of lower base = $\sqrt{2} \cdot l_U$

3.4 DESIGN PARAMETERS

In the parametric design method, new geometries can be generated or modified by changing certain design parameters instead of redesigning the entire structure. Therefore, selecting appropriate design parameters is crucial in the parametric approach and mathematical relationships among the selected parameters should be investigated. All possible design parameters for vaulted tensegrity structures are considered in this chapter, each of which is categorized as input or output parameter.

Two primary design algorithms are considered from a unit-based as well as a structure-based standpoint. Figures 3-10 and 3-11 show diagrams that illustrate the concepts of unit and structure-based design algorithms.

For unit-based design, input parameters include the length of upper and lower bases, unit height, overlap of upper base, and the number of units along the circumference and cylinder's axes. Output parameters generally pertain to the overall structure composed of the units, and include: 1) span, height and opening-angle of the structure, 2) length and angle of strut, 3) lateral cable length, and 4) overlap of the lower

base. The input and output design parameters for the unit-based design are summarized in Table 3-1.

For the unit-based design, the parameters related to the overall structure are considered input parameters, including the span and height of the overall structure, the number of units along the circumference and the cylinder's axis and output parameters are used to represent dimensions of a tensegrity unit, including upper and lower base length, unit height, bar length, and lateral cable length.

These two algorithms cover most scenarios that may be involved into the design of a vaulted tensegrity structure. The input and output design parameters for the structure-based design are summarized in Table 3-2, and design parameters used in both algorithms are illustrated in Figures 3-12, 3-13, and 3-14.

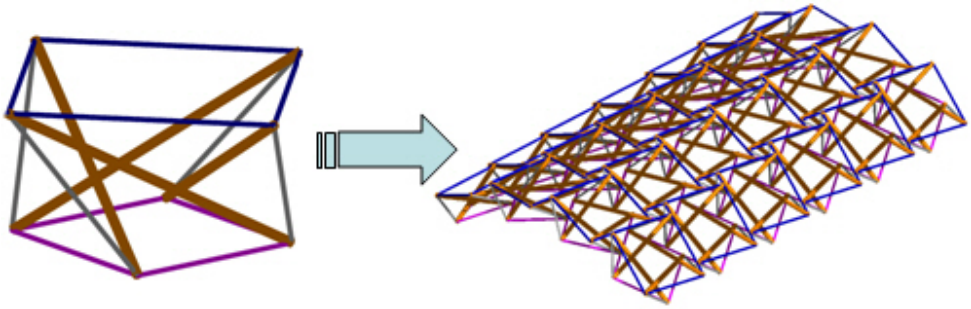


Figure 3-10: Unit-based design concept

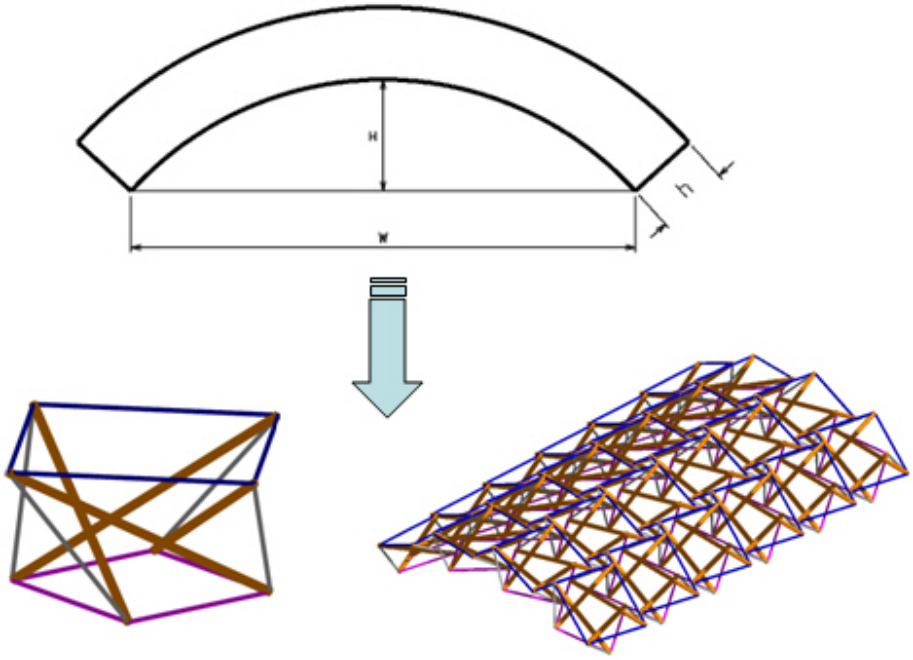


Figure 3-11: Structure-based design concept

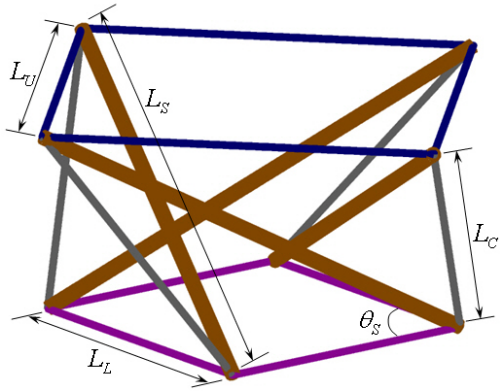


Figure3-12: Unit parameters

L_U : Upper base length

L_L : Lower base length

L_C : Lateral cable length

L_S : Bar length

θ_S : Angle of bars

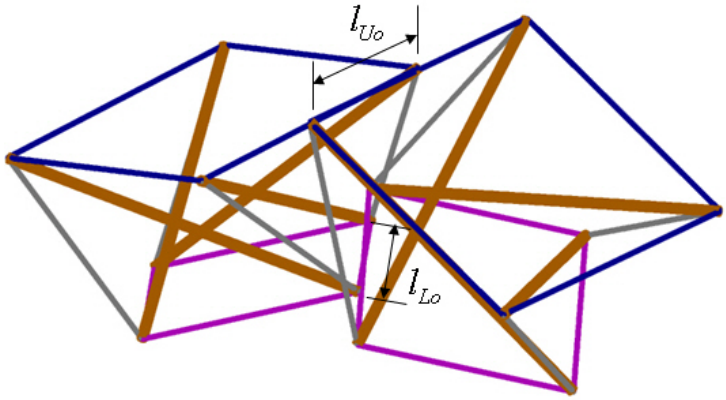


Figure3-13: Overlap parameters

l_{Uo} : Overlap of the upper base

l_{Lo} : Overlap of the lower base

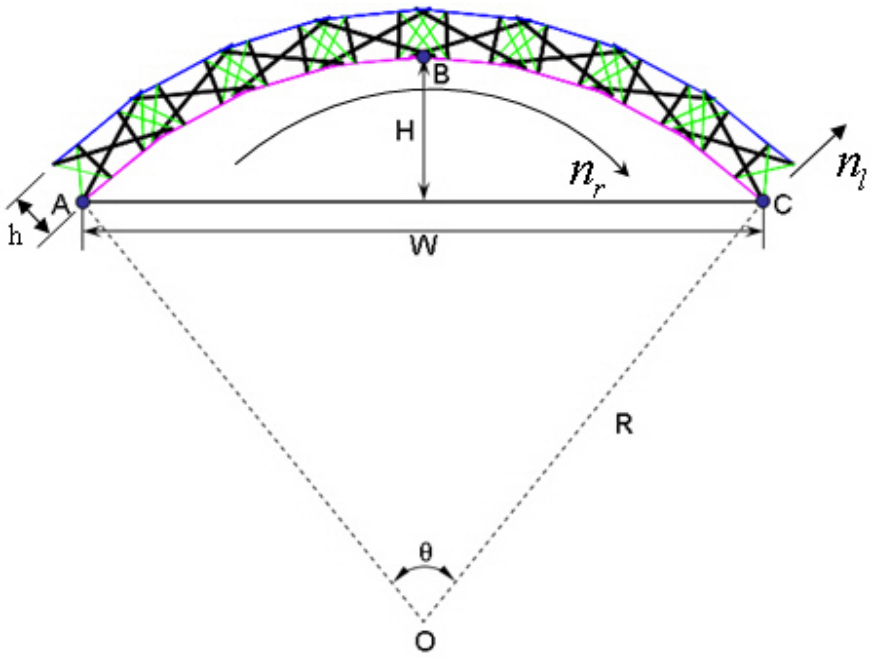


Figure3-14: Design parameters for the structure

H : Structure height

h : Unit height

W : Structure span

θ : Structure opening-angle

R : Structure radius

n_r : Number of units along the circumference

n_l : Number of units along the cylinder axis

Table 3-1: Input and output design parameters for unit-based design

Input Parameters for unit-based design	Output parameters
- Upper base length - Lower base length - Unit height - Overlap of upper base - Number of units along circumference - Number of units along cylinder axis	- Structure span - Structure height - Structure opening-angle - Structure radius - Overlap of lower base - Lateral cable length - Rotation angle between adjacent units - Angle of bars - Bar length

Table 3-2: Input and output design parameters for structure-based design

Input Parameters for the structure-based design	Output parameters
- Structure span - Structure height - Unit height - Overlap of upper base - Number of units along circumference - Number of units along cylinder axis	- Upper base length - Lower base length - Overlap of the lower base - Structure radius - Structure opening-angle - Lateral cable length - Rotation angle between adjacent units - Angle of bars - Bar length

3.5 CALCULATING THE INTERSECTION POINT IN 3D SPACE

In developing this algorithm, several processes required the intersection point of two lines to be determined in 3D space instead of 2D plane. These processes were mathematically expressed through a general analytical geometry procedure that determines the intersection point in 3D space by the vector properties, as follows:

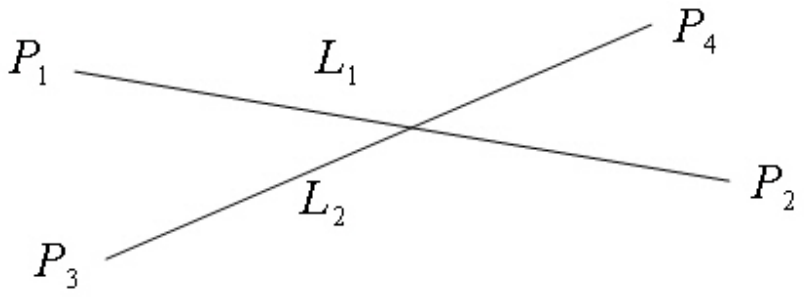


Figure 3-15: Two lines intersecting in space

Let's assume we have two lines L_1 and L_2 in space and each line is connected from point P_1 to P_2 and P_3 to P_4, respectively (Figure 3-15). These lines can be expressed in a vector form:

$$L_1 = P_1 + u(P_2 - P_1) \text{ and } L_2 = P_3 + w(P_4 - P_3)$$

By defining $D_1 = P_2 - P_1$ and $D_2 = P_4 - P_3$, and then, substituting $P_2 - P_1$ for D_1 and $P_4 - P_3$ for D_2 these two line equations can be written as:

$$L_1 = P_1 + uD_1 \text{ and } L_2 = P_3 + wD_2$$

Let's define another perpendicular vector $(P_3 \times D_2)$ and implement the dot product with these two lines. The results of the dot product are zero due to the properties of dot product and the following expression can be derived:

$$(P_3 \times D_2) \bullet (P_1 + uD_1) = (P_3 \times D_2) \bullet (P_3 + wD_2) = 0$$

Accordingly, the values of scalars *u* and *w* can be derived as follows:

$$u = -\frac{(P_3 \times D_2) \bullet P_1}{(P_3 \times D_2) \bullet D_1}, \quad w = -\frac{(P_1 \times D_1) \bullet P_3}{(P_1 \times D_1) \bullet D_2}$$

The intersection points are then calculated by substituting the value *u* or *w* for the *u* or *w* of the line equation L_1 or L_2.

3.6 UNIT-BASED PARAMETRIC DESIGN ALGORITHM

In general, the unit-based design method can be used to design a vaulted tensegrity structure when the dimensions of the unit are predetermined. The overall procedure for the unit-based design algorithm and geometric model development process are shown in Figure 3-16. This algorithm allows the user to generate the initial geometry of a vaulted tensegrity structure from (a) the dimensions of composing units, and (b) unit-overlap conditions, that is: by the upper and lower-base overlap length or percentage between adjacent units.

This algorithm was developed as an analytical geometry procedure that investigates the geometric relationship between design parameters and that expresses this process mathematically using common trigonometric functions and rotation matrices. This unit-based design provides the flexibility that the designer can reuse tensegrity modules in other applications where the final geometry can be re-modeled by simply modifying the design parameters.

To initialize this process of the unit based design, the length of the lower base must fall into the feasible region in which the lower base can be overlapped. This constraint is evaluated by the following equations, already presented in the section on the "Geometric configuration of vaulted tensegrity structures" discussed previously.

- Minimum length of lower base = $\frac{l_U}{\sqrt{2}}$
- Maximum length of lower base = $\sqrt{2} \cdot l_U$

where l_U is the length of upper base.

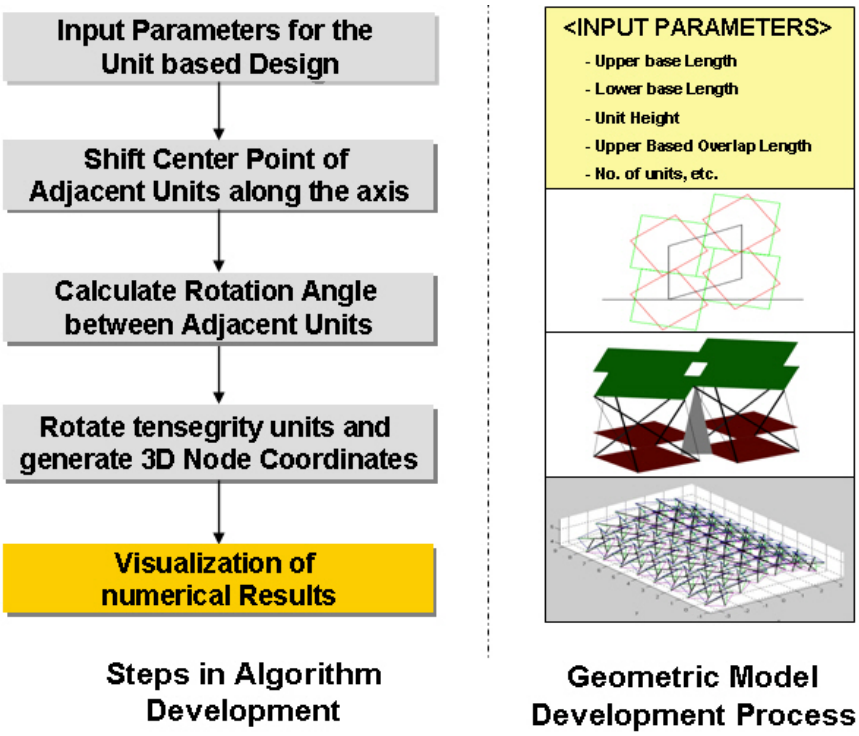


Figure 3-16: Steps in unit-based design algorithms and geometric model development processes

3.6.1 Determining Unit-overlap along the Axis of the Cylinder

When the constraint is satisfied, the initial geometry of tensegrity structures can be generated from the unit dimensions and overlap lengths. The scheme in Figure 3-17 shows the geometric relationships that apply to adjacent units ("Unit A": along the

cylinder's axis and "Unit B": along the circumference). Initially, the temporary origin is set up in the center of "Unit O", and x and y coordinate axes are drawn in a parallel direction to the sides of the upper base so that the coordinates of all nodes can be easily determined. This origin will be moved after rotating the tensegrity units to make the cylinder's axis parallel to the global y-direction, and the position of this global origin will be determined later.

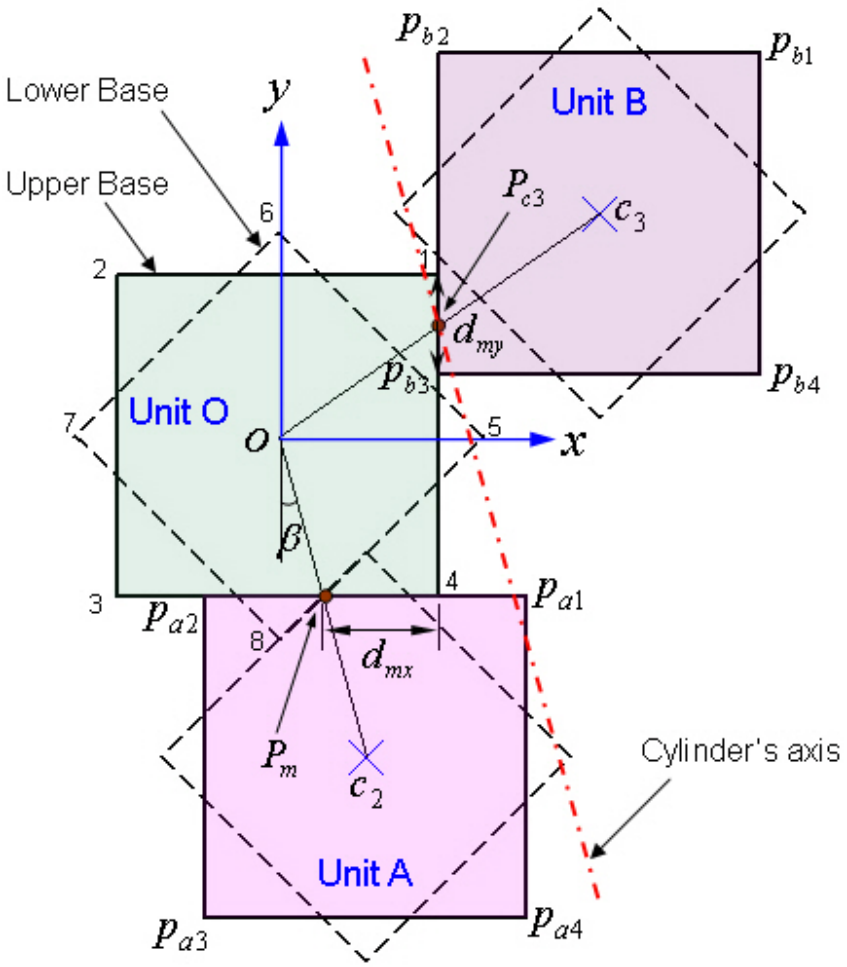


Figure 3-17: Geometric relationships that apply to adjacent units

In order to develop single-curvature structures, the upper and lower bases of adjacent units along the cylinder's axis ("Unit O" and "Unit A") should meet at their lower bases. There exist two overlap lengths for the upper base; one is along the cylinder's axis and the other is along the circumference direction. These overlap lengths are totally independent, and the overlap length along the circumference direction is fixed and can be calculated based on the dimensions of the unit by the following procedure:

The point P_m is first determined to calculate the overlap length ($2 \cdot d_{mx}$) of the upper bases of units along the circumference. The scheme in Figure 3-17 indicates that the point P_m is at intersection of two lines, one connecting vertices 3 and 4 of the upper bases and the other connecting vertices 5 and 8 of the lower bases, P_m can be calculated using the intersection equation introduced earlier.

Finally, the length d_{mx} which is half of the upper-base overlap along the circumference, can be calculated from two points, P_m and vertex 4.

The next step is to find the coordinates of nodes P_{a2} and P_{a4} for the generation of the adjacent tensegrity units along the cylinder's axis. Since the length of overlap d_{mx} is known, the coordinates, P_{a2} and P_{a4} are calculated from the geometrical relationship.

$$P_{a2} = \left(\frac{l_U}{2} - 2d_{mx}, -\frac{l_U}{2}, h \right)$$

$$P_{a4} = \left(\frac{3l_U}{2} - 2d_{mx}, -\frac{3l_U}{2}, h \right)$$

$$c_2 = \frac{P_{a2} + P_{a4}}{2}$$

where l_U is the length of the upper base and h is unit height of the tensegrity

Based on these two coordinates, all 3D coordinates of adjacent units along the axis of the cylinder can be calculated, including the center point c_2 of "Unit A". The line $\overline{\mathrm{Oc}_2}$ is parallel to the cylinder's axis and the calculated angle β will be used to rotate these units to align with the global y-coordinate. The lower-base overlap length between "Unit O" and "Unit A" is also computed using the same procedure.

3.6.2 Determining Unit-overlap along the Circumference

Calculating all coordinates of "Unit B" is the same as with "Unit A", except that the upper-base overlap length d_{my} along the cylinder's axis is not fixed but varied by the given input value. The upper-base overlap length can be input by two options: by the actual length of overlap or by the percentage of overlap. First, the points of vertices p_{b2} and p_{b4} are needed to calculate including the center point c_3. These coordinates are:

$$P_{b2} = \left(\frac{l_U}{2}, \frac{3l_U}{2} - l_{Uo}, h \right)$$

$$P_{b4} = \left(\frac{3l_U}{2}, \frac{l_U}{2} - l_{Uo}, h \right)$$

$$c_3 = \frac{P_{b2} + P_{b4}}{2}$$

where l_{Uo} is the length of the upper-base overlap

After all 3D coordinates of "Unit B" are calculated, "Unit O", "Unit A", and "Unit B" are rotated clockwise by angle β to be aligned with global coordinates. A 3D rotation matrix is used for this process. The remaining units along the circumference are repeated from the coordinates of "Unit B" with offset D_{ax} and D_{ay} which are calculated from the distance between two center points of "Unit O" and "Unit B". Figure 3-18 shows the new geometric configuration after rotating by angle β and the

temporary origin moved to the global origin *O* as indicated in the scheme because all of the other units will be generated along this global coordinates. The remaining units along the cylinder's axis are also created by repeating the coordinates of "Unit O" with offset distance between two center points of "Unit O" and "Unit A".

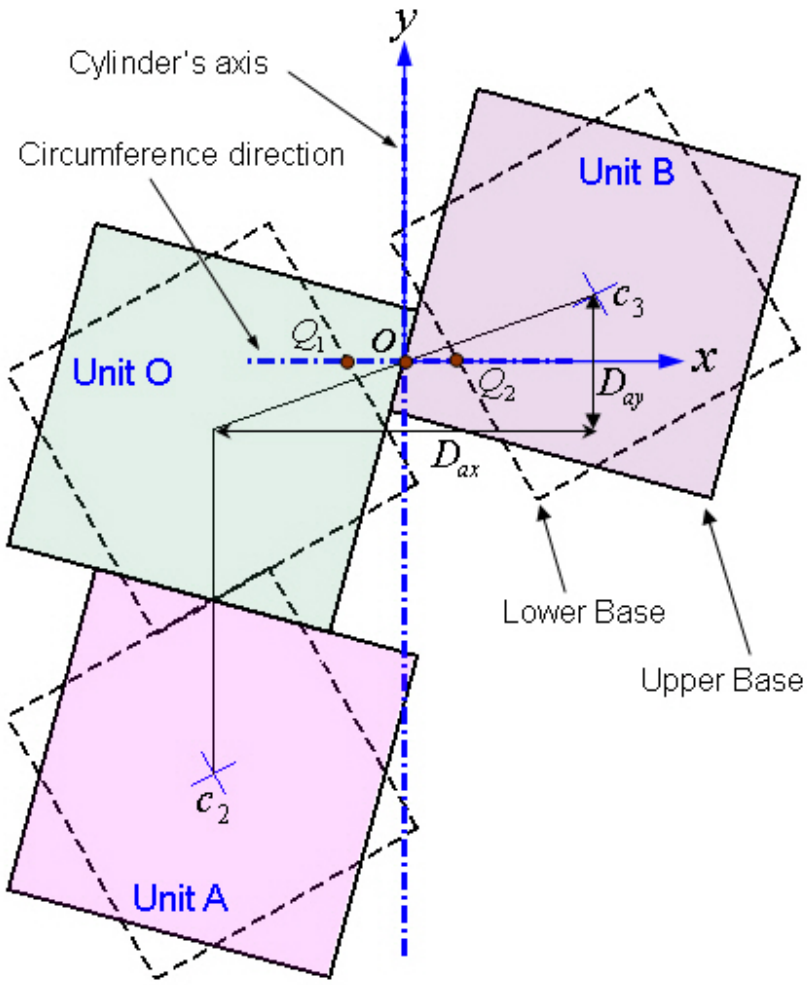


Figure 3-18: Tensegrity configuration after rotating units by an angle β

3.6.3 Determining Angle of Unit Rotation

The angle of rotation ϕ between vectors $\vec{u}$ and $\vec{v}$ of adjacent units along the circumference is calculated from the dot product of vectors $\vec{u}$ and $\vec{v}$ that originate from O and end at intersection points Q_1 and Q_2 (Figure 3-19). The intersection points Q_1 and Q_2 lie on the plane of lower base and are perpendicular to the cylinder's axis as indicated on the scheme of Figure 3-20.

$$\phi = \cos^{-1}\left(\frac{u_x v_x + u_z v_z}{|u||v|}\right), \text{ where } \vec{u} = (u_x, u_z) \text{ and } \vec{v} = (v_x, v_z)$$

Initially, all tensegrity units are placed on a flat plane and a 3D rotation matrix with a rotation angle ϕ is applied to each row of units parallel to the cylinder's axis. The rotated 3D coordinates of all nodes are computed and saved for later use in visualization and analysis.

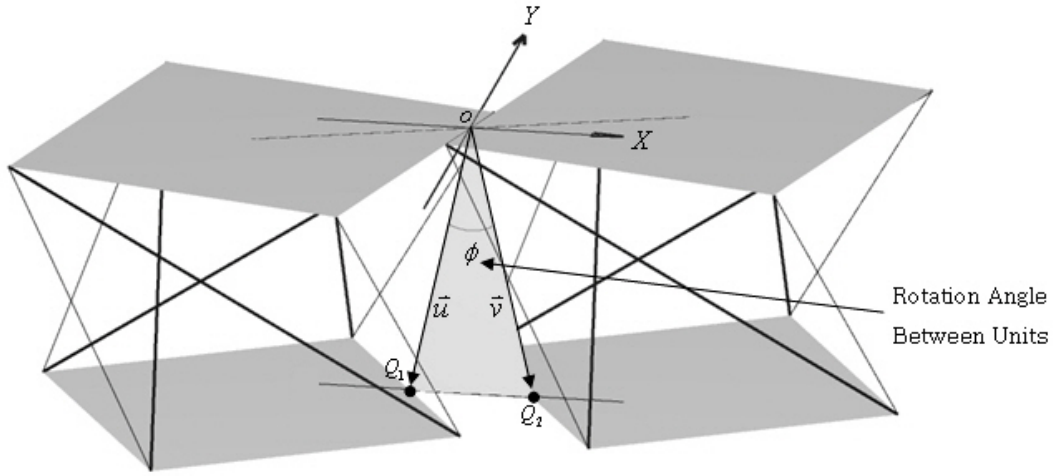


Figure 3-19: Angle of unit rotation

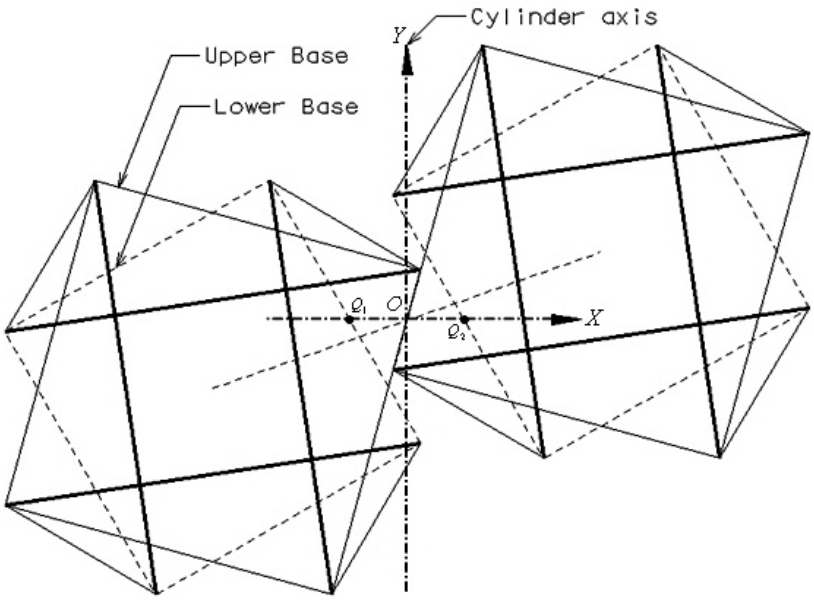


Figure 3-20: Intersection points Q_1 and Q_2 between adjacent units

3.6.4 Graphical Representation of Unit-based Design

Programming codes implementing the above algorithm have been developed in a MATLAB software environment, named *vaultdwg.m*. The codes take into account all interrelated parameters and constraints, and allow the user to modify such inputs as: a) unit dimensions (upper and lower bases and its height), b) value of the upper-base overlap, and c) the number of units along the cylindrical and circumference direction. Figure 3-21 shows an example input for *vaultdwg.m* in MATLAB.

Given the values of input parameters, the unit-based design algorithm yields the output: (a) remaining unit dimensions (bar lengths, angle of the bar, and the length of lateral cables), (b) information about the structure (radius of the structure, structure height, structure span, and structure opening-angle), (c) angle of rotation between

adjacent units and (d) overlap of lower or upper bases, respectively. Figure 3-22 shows an example of an output generated from the *vaultdwg.m* codes.

The parametric codes allow for an initial form exploration of tensegrity structures by displaying geometric information in a 3D graphical environment. Figure 3-23 illustrates 4x6 models of vaulted tensegrity structures with the following input data; the upper base length 2.5, the lower base length 2.1, the unit height of tensegrity 1.2, overlap percentage 50%. Figure 3-24 shows 5x6 models created from the following input data; the upper base length 2.3, the lower base length 2.1, the unit height of tensegrity 1.2, overlap percentage 50%.

```
%  ============================================================================
%  =   vaultdwg.m   By Jinman Kim on 02/10/04                                 =
%  =   To generate vaulted tensegrity networks                                =
%  =   using the Unit Based Design Algorithm                                  =
%  =                                                                          =
%  =   The Input data for this algorithm are below                            =
%  =   T_L  = The Upper base length                                          =
%  =   B_L  = The Lower base length                                          =
%  =   ht  = Height of the Tensegrity Unit                                    =
%  =   ovt = Overlap length of the Upper base        ( for the CASE 1)        =
%  =   ovt_P = Overlap percentage of the Upper base  ( for the CASE 2)        =
%  =   num_unit_l = The number of units along the cylindrical axis            =
%  =   num_unit_r = The number of units along the circumference axis          =
%  ============================================================================
clear all;      % Clear all previous data
%#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=
T_L = 2.5;      % The Upper base length
B_L = 2.2;      % The Lower base length
ht = 1.2;       % Height of the Tensegrity Unit

% Overlap length of the Upper base along the cylindrical axis (for CASE 1)
ovt = 1.6;
% Overlap percent of the Upper base along the cylindrical axis (for CASE 2)
ovt_P = 50;

num_unit_l = 6;     % The number of units along the cylindrical axis
num_unit_r = 4;     % The number of units along the circumference axis

% #=#=#=#=#   Choose  CASE Flag No.  #=#=#=#=#
flg = 2;
% 1 = Case 1: Proceed with the actual overlap length (ovt)
% 2 = Case 2: Proceed with the overlap percentage (ovt_P)

% #=#=#=#=#   Options to Control Output Display    #=#=#=#=#
Disp_Node_No_TOP = 0;  % Option to display Node No. of upper base, (1=on, 0=ff)
Disp_Node_No_BOT = 0;  % Option to display Node No. of lower base, (1=on, 0=ff)
Disp_BAR = 1;          % Display Strut on or off, ( 1=on, 0=off)
Disp_Lateral_cable = 1;% Display lateral cable on or off, ( 1=on, 0=off)
grid_on=1;             % Turn on or off the grid (1=on, 0=off)
xy_label_on=1;         % Turn on or off the x, y and z label (1=on, 0=off)
%#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=
```

Figure 3-21: Example of input parameters of *vaultdwg.m*

```
############## ===   OUTPUT   === ################
   ------<< For the Entire Structure >>---------
* Structure Span = 8.4459
* Structure Hight = 2.1125
* Height of the Tensegrity Unit = 1.2
* No. of units along the cylindrical axis = 6
* No. of unit along the circumference axis = 4

* Structure Radius = 5.2772
* Structure opening angle = 106.3032(degree)
* Rotation Angle between units = 24.8312(degree)
* Angle of a BAR = 21.34(degree)
* Length of a BAR = 3.2976

   -------- << For the Upper Base >> -----------
* Length of the Upper Base = 2.5
* Upper based overlap along the cylindrical axis = 1.25
* Upper based overlap % along the cylindrical axis = 50 %

* Upper based overlap along the circumference axis = 1.8887
* Upper based overlap % along the circumference axis  = 75.5492 %

   --------- << For the Lower Base >> ---------
* Length of the Lower Base = 2.2
* Lower based overlap along the cylindrical axis = 1.5804
* Lower based overlap % along the cylindrical axis = 71.8349 %

* Lower based overlap along the circumference axis = 0.86447
* Lower based overlap % along the circumference axis = 39.2939 %

================================================
```

Figure 3-22: Output parameters generated from vaultdwg.m

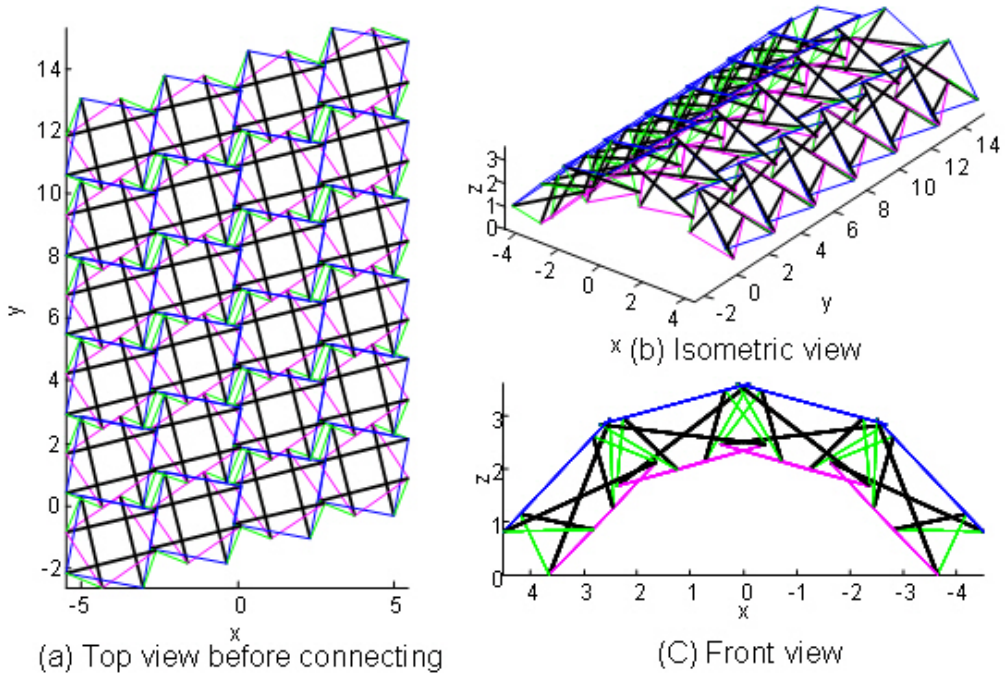


Figure 3-23: 4x6 model of a vaulted tensegrity structure

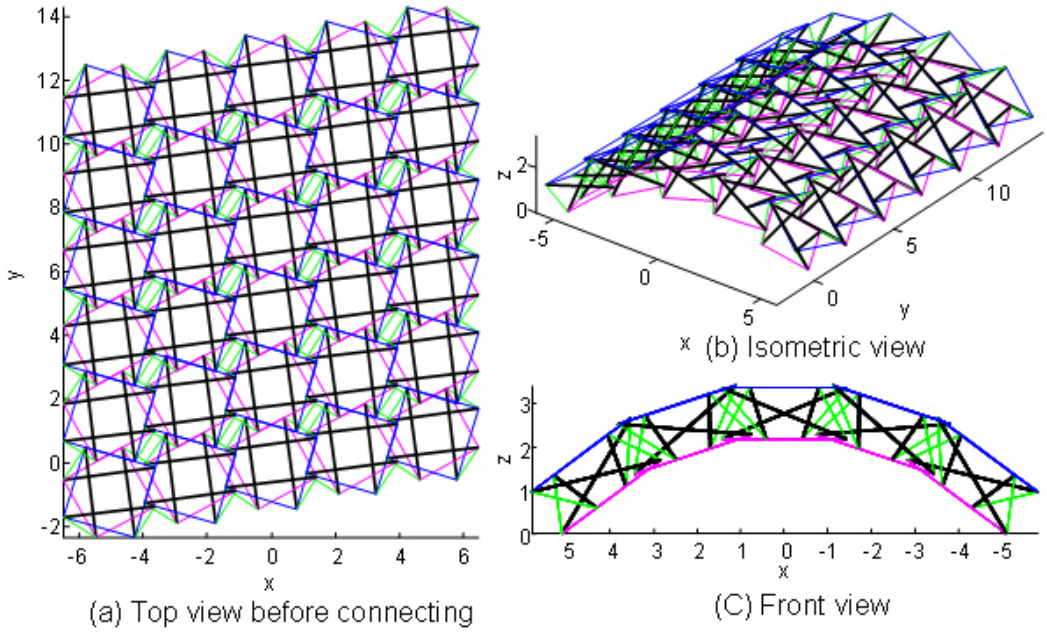


Figure 3-24: 5x6 model of a vaulted tensegrity structure

3.7 STRUCTURE-BASED PARAMETRIC DESIGN ALGORITHM

The structure-based design algorithm determines optimal unit dimensions for generating geometry of tensegrity structures when the dimensions of the entire structure are given, such as structure span and height. Consequently, this design process called "reverse design" appears to be much more complex compared to unit-based design and in fact, involves more steps, additional constraints, and iteration processes. Figure 3-25 illustrates the overall process of the structure-based design. Final dimensions of the units are not unique because a number of feasible solutions can be found that satisfy the given structure dimensions. The unique solution may be achieved only after restricting the set of feasible solutions by adding design constraints.

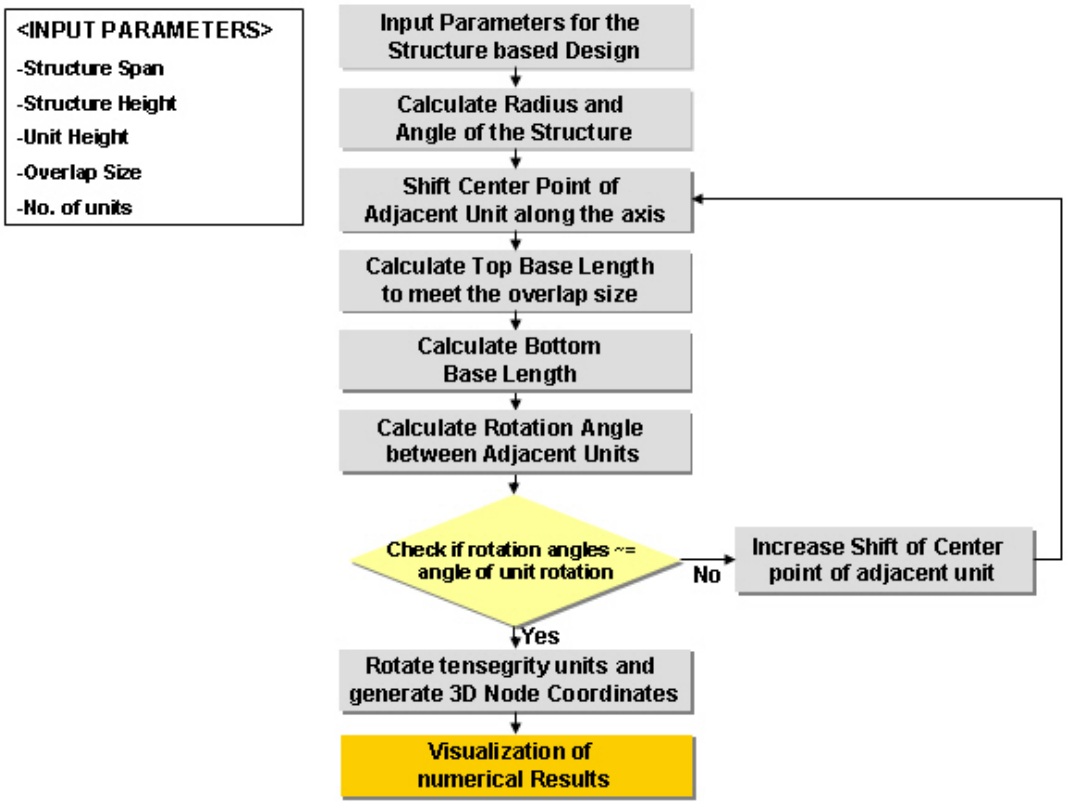


Figure 3-25: Flow chart for structure-based design algorithm

3.7.1 Curvature and Opening-angle of the Structure

As an initial step for the geometric process, the curvature of the structure, expressed in a combination of the radius R and the opening-angle θ, are calculated from the three points *A*, *B*, and *C* shown in Figure 3-26.

These three points can be defined with given ***W*** and ***H*** values as:

$$\begin{cases} A=(x_1,z_1)=\left(-\dfrac{W}{2},0\right) \\ B=(x_2,z_2)=(0,H) \\ C=(x_3,z_3)=\left(\dfrac{W}{2},0\right) \end{cases} \tag{3.9}$$

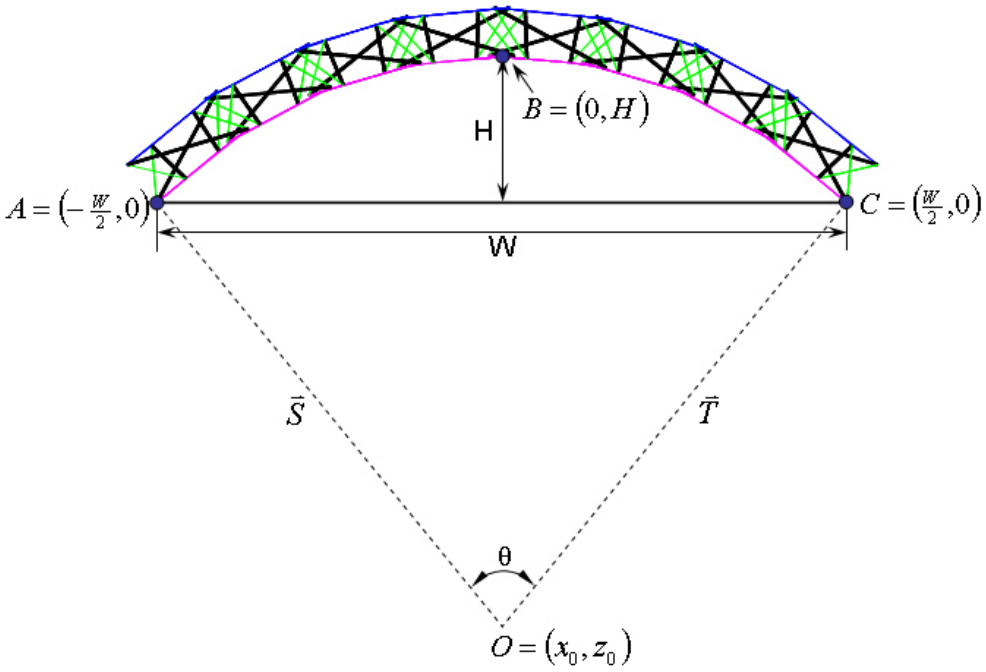


Figure 3-26: Structure parameters for calculating structure radius and opening-angle

Assuming the center point *O* is defined as (x_0, z_0), a circle equation with these three points can be expressed as:

$$2x_i x_0 + 2z_i z_0 - G = x_i^2 + z_i^2 \quad \text{for } i = 1, 2, 3 \tag{3.10}$$

$$\text{where} \quad G = x_0^2 + z_0^2 - R^2 \tag{3.11}$$

Note that the left hand side of the equation (3.10) involves three parameters of unknown value, i.e., x_0, z_0, and G, and the right hand side is in terms of parameters with known values. Substituting known three points *A*, *B*, and *C* into (3.10) gives:

$$\begin{bmatrix} 2x_1 & 2z_1 & -1 \\ 2x_2 & 2z_2 & -1 \\ 2x_3 & 2z_3 & -1 \end{bmatrix} \begin{Bmatrix} x_0 \\ z_0 \\ G \end{Bmatrix} = \begin{Bmatrix} x_1^2 + z_1^2 \\ x_2^2 + z_2^2 \\ x_3^2 + z_3^2 \end{Bmatrix} \tag{3.12}$$

This equation can be rewritten in a matrix form:

$$[\mathbf{K}]\{\mathbf{U}\} = \{\mathbf{V}\} \tag{3.13}$$

where,

$$[\mathbf{K}] = \begin{bmatrix} 2x_1 & 2z_1 & -1 \\ 2x_2 & 2z_2 & -1 \\ 2x_3 & 2z_3 & -1 \end{bmatrix}, \quad \{\mathbf{U}\} = \begin{Bmatrix} x_0 \\ z_0 \\ G \end{Bmatrix}, \quad \{\mathbf{V}\} = \begin{Bmatrix} x_1^2 + z_1^2 \\ x_2^2 + z_2^2 \\ x_3^2 + z_3^2 \end{Bmatrix}$$

The linear system of equations (3.13) can be solved for the unknown vector $\{\mathbf{U}\}$ by taking the inverse matrix of $[\mathbf{K}]$:

$$\{\mathbf{U}\} = [\mathbf{K}]^{-1}\{\mathbf{V}\} \tag{3.14}$$

Now that the value of G is known by (3.14), the structure radius can be determined from the equation (3.11):

$$R = \sqrt{x_0^2 + z_0^2 - G} \tag{3.15}$$

On the other hand, the opening-angle θ of the entire structure is derived from the dot product of two vectors $\vec{S}$ and $\vec{T}$:

$$\vec{S} \bullet \vec{T} = \left|\vec{S}\right|\left|\vec{T}\right| \cos\theta \tag{3.16}$$

where $\vec{S} = (S_x, S_z) = (x_1 - x_0, z_1 - z_0)$ and $\vec{T} = (T_x, T_z) = (x_3 - x_0, z_3 - z_0)$

Solving (3.16) for θ, the structure opening-angle is explicitly written as:

$$\theta = \cos^{-1}\left(\frac{S_x T_x + S_z T_z}{\left|\vec{S}\right|\left|\vec{T}\right|} \right) \tag{3.17}$$

3.7.2 Determing the Relationship between Rotation and Opening Angles

In the structure-based design, the dimensions of the unit do not serve as input data. Hence, the equation (3.8) for calculating the rotation angle between units cannot be used until unit dimensions are determined. However, this rotation angle can be predetermined from the inputs of structure span, height, and the number of unit along the circumference direction. This rotation angle is denoted by δ (Figure 3-27)

$$\delta = \frac{\theta}{n} \tag{3.18}$$

where θ is the structure opening-angle and n is the number of units along the circumference direction.

This predetermined angle δ is compared with the rotation angle ϕ calculated by the equation (3.8).

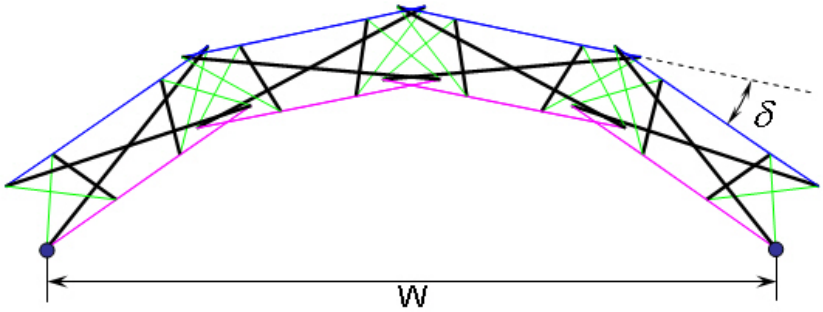


Figure 3-27: Determining rotation angle δ

For the next step, the upper and lower base dimensions are computed by increasing the center point of unit gradually while satisfying the given overlap constraint as shown in Figure 3-4. During this iteration process, the unit dimensions are varied with an amount of the center shifting and the rotation angle ϕ between adjacent units is calculated at each step. This rotation angle ϕ calculated at each iteration will be checked against the predetermined rotation angle δ. If the rotation angle ϕ is smaller than δ, the iteration process will continue with a small increment of center shift until the rotation angle ϕ between units becomes equal to the predetermined angle δ.

3.7.3 Overlap Input Methods

In the structure-based design algorithm, three different methods for defining the unit overlap among adjacent units are considered. According to the first method the dimensions of each unit are determined using a fixed value of overlap length as a basic design constraint. In the second case, the overlap of the upper-base is expressed as a percentage instead of an actual numerical value. In the last case the amount of the upper and lower-base overlap is determined by using the fixed upper base length as the primary constraint. Mathematical relationships and corresponding algorithms for each of these methods are described in detail.

• Case 1: Given upper-base overlap length

In this case, the length of the upper-base overlap between adjacent units, expressed in an actual numerical value, is used as a given constraint. The upper and lower bases can be determined by the following analytical process.

Figure 3-28 shows an initial configuration of two adjacent units along the circumference and detailed parameters. Let d_1 be a half the upper-base overlap length and $d_1 + d_2$ be a half the upper base length. Since the upper-base overlap length is given as a numerical value, the value d_2 is all that is required to determine the upper base length.

The geometry in Figure 3-28 leads to the following equation:

$$d_2^2 + (d_1 + d_2)^2 = r^2 \tag{3.19}$$

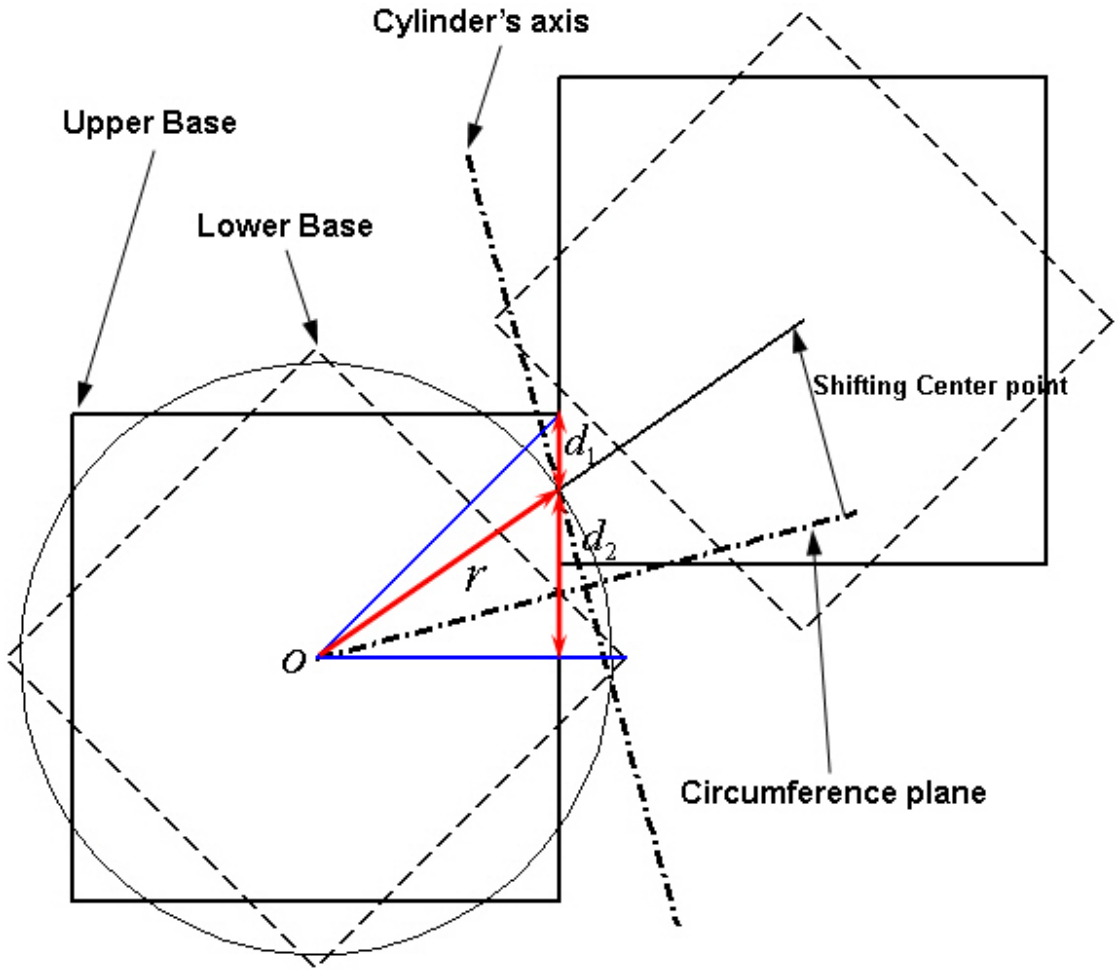


Figure 3-28: Unit-overlap geometry and parameters

From (3.19) the unknown d_2 can be expressed as a function of d_1 and r:

$$d_2 = \frac{-2d_1 + \sqrt{4d_1^2 - 8\left(d_1^2 - r^2\right)}}{4} \tag{3.20}$$

Note as the value r changes when the center point is shifted upward, so does the value d_2.

• ***Case 2: Given upper-base overlap percentage***

In this case, the upper-base overlap is given in terms of percentage instead of actual length. As such, other design parameters should be evaluated as a function of the given overlap percentage, making it more complicated to find a relationship among them. Detailed parameters required for this process are shown in Figure 3-29.

The first step in this process begins with a triangle $o\,q_1\,q_2$. Applying a sine rule and trigonometric function to this triangle, the relationship between two parameters r and d_1 can be written as:

$$r = \frac{d_1}{\sqrt{2}\sin\left(45^\circ - \alpha\right)} \tag{3.21}$$

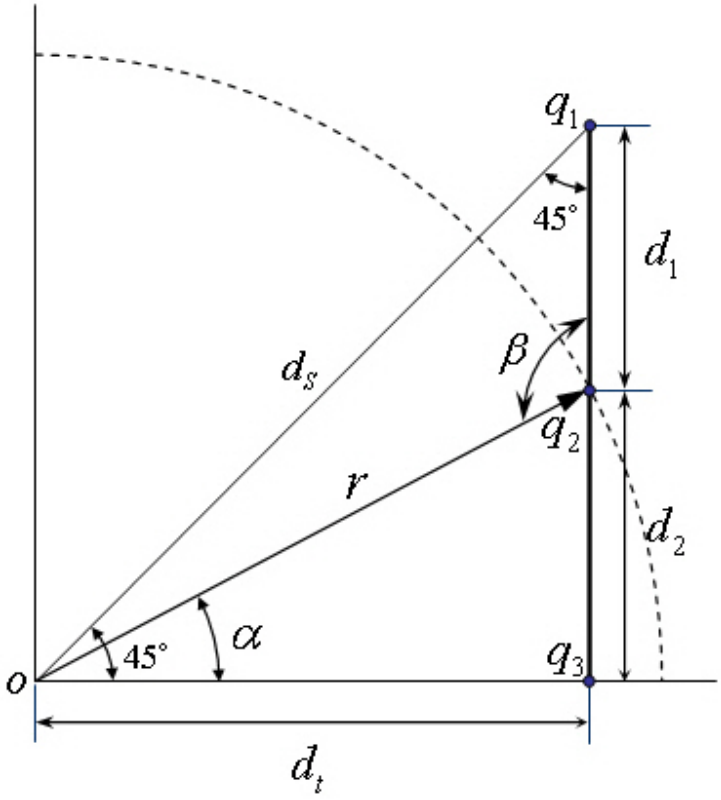


Figure 3-29: Parameters for the structure-based design(Case II)

In addition, the following geometric relationships can be observed in the scheme of Figure 3-29.

- $\beta = 90^\circ + \alpha$
- d_1 is a half the upper-base overlap length
- $d_t = d_1 + d_2$, where d_t is a half the upper based length

Each design parameter is affected by the upper-base overlap and the influence of overlap % is listed in Table 3-3.

Table 3-3: Constraints of the design parameters influenced by overlap %

Overlap %	d_1	d_2	α	d_t	l_U
0 %	$d_1 = 0$	$d_2 = \frac{\sqrt{2}}{2} r$	$\alpha = 45^\circ$	$d_t = \frac{\sqrt{2}}{2} r$	$\sqrt{2}\, r$
50%	$d_1 = r$	$d_2 = 0$	$\alpha = 0^\circ$	$d_t = r$	$2\, r$
100%	$d_1 = \sqrt{2} r$	$d_2 = 0$	$\alpha = -45^\circ$	$d_t = \frac{\sqrt{2}}{2} r$	$\sqrt{2}\, r$

Since a constraint of the upper-base overlap is given by a percentage, the parameter d_1 can be described by the given percentage. Let m denote the percentage, then d_1 can be written as:

$$d_1 = \frac{m}{100} d_t \tag{3.22}$$

Substituting $d_1 + d_2$ for d_t in (3.22) and solving it for d_2 is also expressed by the given percent such as:

$$d_2 = \frac{(100-m)}{m} d_1 \tag{3.23}$$

Applying the Pythagorean theorem to the triangle oq_2q_3 in Figure 50 leads the following equation as:

$$d_2^2 + d_t^2 = r^2 \tag{3.24}$$

Substituting (3.21), (3.22), and (3.23) into (3.24) gives:

$$\left(\frac{100-m}{m}\right)^2 d_1^2 + \left(\frac{100}{m}\right)^2 d_1^2 = \frac{1}{2}\left(\frac{1}{\sin(45^\circ - \alpha)}\right)^2 d_1^2 \tag{3.25}$$

Dividing both sides of (3.25) by d_1^2 and using the addition formula for sine $\sin(45^\circ - \alpha) = \sin 45^\circ (\cos\alpha - \sin\alpha)$, we obtain the following result:

$$\frac{1}{\cos^2\alpha + \sin^2\alpha - 2\cos\alpha\sin\alpha} = \frac{m^2 - 200m + 20000}{m^2} \tag{3.26}$$

Finally, (3.26) is solved for α after applying addition formula $\sin(2\alpha) = 2\cos\alpha\sin\alpha$ and the identity $\cos^2\alpha + \sin^2\alpha = 1$:

$$\alpha = \frac{1}{2}\sin^{-1}\left(1 - \frac{m^2}{m^2 - 200m + 20000}\right) \tag{3.27}$$

Now that we have found a solution for the angle α expressed in terms of the given percentage, substituting (3.27) into (3.21) gives the half the upper-base overlap length:

$$d_1 = \sqrt{2}\sin\left(45^\circ - \alpha\right)r \tag{3.28}$$

Then d_2 can be found using the same formula as in (3.20):

$$d_2 = \frac{-2d_1 + \sqrt{4d_1^2 - 8\left(d_1^2 - r^2\right)}}{4}$$

Note that in this case, the value r affecting the solution of d_2 is also varied as the center point is shifted upward. The upper based length now can be determined from these d_1 and d_2.

• Case 3: Given upper base length

This method is different from the previous two cases which are primarily concerned with calculating the upper base length from the given upper-base overlap length and percentage, respectively. In this third case, the numerical value of the upper base length is given as input, and the upper-base overlap length should be determined so that it satisfies the given structure constraints like structure span and height. Since the known design parameters are d_t (the half of an upper based length) and r (measurable when the center points are shifting), the remaining parameters should be expressed in terms of d_t.

Applying the sine rule to the triangle $o\,q_1\,q_2$ in Figure 3-29, the distance d_S between points o and q_1 is given by:

$$d_S = \frac{r\sin\beta}{\sin 45^\circ}, \quad \text{where} \quad \beta = 90^\circ + \alpha \tag{3.29}$$

It can be shown that the length d_S remains constant during this iterative process because the given upper base length is fixed. Another equation for d_S is obtained by applying the Pythagorean theorem to the triangle $o\,q_1\,q_3$ in Figure 3-29:

$$d_S = \sqrt{2}\,d_t \tag{3.30}$$

By combining these two equations (3.29) and (3.30) with the identity of $\sin(90^\circ + \alpha) = \cos\alpha$, the following equation is obtained:

$$\sqrt{2}\,r\cos\alpha = \sqrt{2}\,d_t \tag{3.31}$$

Therefore, the angle can be written in terms of d_t as:

$$\alpha = \cos^{-1}\frac{d_t}{r} \tag{3.32}$$

Finally, we can derive the solution of the d_2 from the triangle $o\,q_2\,q_3$ as:

$$d_2 = r\sin\alpha \tag{3.33}$$

Note that, in the third case, there exists a minimum length of the upper base because the distance between two units along the circumference direction is predetermined by the given information of a structure and the number of the units along that axis. Therefore, the given upper base length should be greater than this distance denoted by D_{ax} in Figure 3-18. If the given upper base length is smaller than D_{ax}, the upper-base overlap cannot be created. In addition, the maximum of radius r is also constrained by the fixed length of the given upper base, as shown in Figure 3-30.

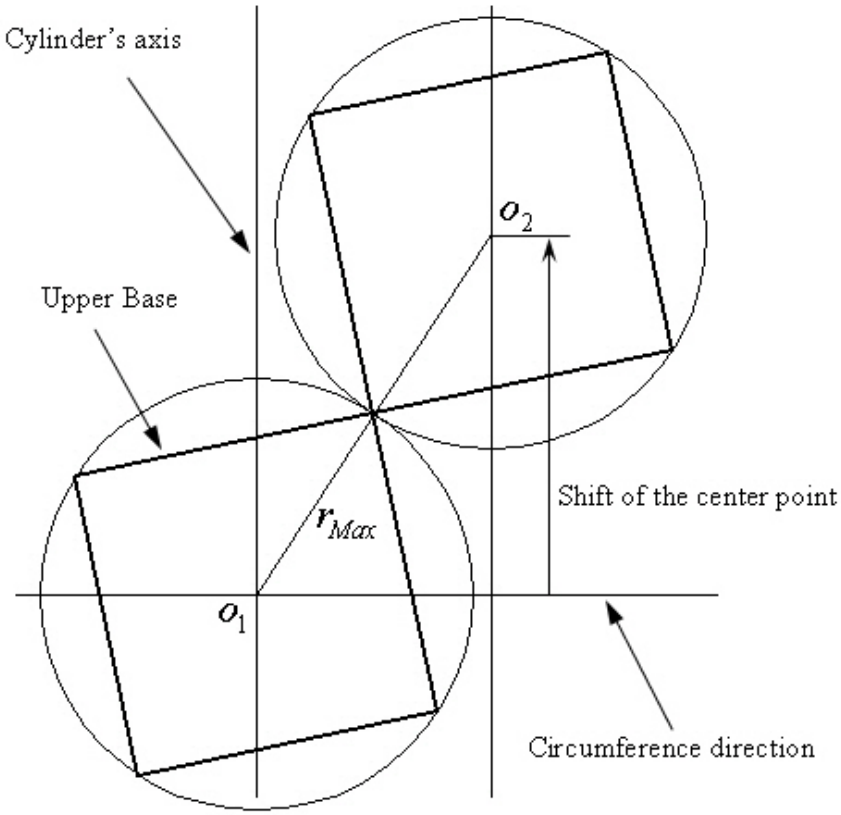


Figure 3-30: Geometric configuration of a maximum r

3.7.4 Graphical Representation of Structure-based Design

A MATLAB code named *vaultdgn3.m* has been developed to implement the above algorithms; all interrelated parameters and constrains are considered in this code. The input parameters for this code are structure span, height, unit height of structure, and the number of units along the cylindrical and circumference direction. The three methods developed in the previous section for the input of the upper-base overlap are coded and Figure 3-31 shows *vaultdgn3.m*.

```
% ======================================================================
% =   vaultdgn3.m   By Jinman Kim on 05/25/04                          =
% =   To generate vaulted tensegrity networks                          =
% =   using the Structure Based Design Algorithm                       =
% =   The Input data for this algorithm are below                      =
% =   W = Structure Span                                               =
% =   H = Structure Height                                             =
% =   ht  = Height of the Tensegrity Unit                              =
% =   ovt = Overlap length of the Upper base        ( for the CASE 1)  =
% =   ovt_P = Overlap percentage of the Upper base  ( for the CASE 2)  =
% =   T_L  = The Upper base length                  ( for the CASE 3)  =
% =   num_unit_l = The number of units along the cylindrical axis      =
% =   num_unit_r = The number of units along the circumference axis    =
% ======================================================================
clear all;      % Clear all previous data
%-------------------------------------------------------------------
W = 10;          % Structure Span
H=2;            % Structure Height
ht = 1;          % Height of the Tensegrity Unit

% Overlap length of the Upper base along the cylindrical axis (for CASE 1)
ovt = 1.2;
% Overlap percent of the Upper base along the cylindrical axis (for CASE 2)
ovt_P = 50;
T_L=10;          % The Upper base length   (for CASE 3)
num_unit_l = 6;    % The number of units along the cylindrical axis
num_unit_r = 4;    % The number of units along the circumference axis

% #=#=#=#=#   Choose  CASE Flag No.  #=#=#=#=#
flg = 2;     % 1 = Case 1: Proceed with the actual overlap length (ovt)
             % 2 = Case 2: Proceed with the overlap percentage (ovt_P)
             % 3 = Case 3: Proceed with the fixed Upper base length (T_L)

file_name='test2.txt';     % File name for the Input of VBA of Microstation

% #=#=#=#=#   Options to Control Output Display    #=#=#=#=#
Disp_Node_No_TOP = 0;  % Option to display Node No. of upper base, (1=on, 0=ff)
Disp_Node_No_BOT = 0;  % Option to display Node No. of lower base, (1=on, 0=ff)
Disp_BAR = 1;          % Display Strut on or off, ( 1=on, 0=off)
Disp_Lateral_cable = 1;% Display lateral cable on or off, ( 1=on, 0=off)
grid_on=0;             % Turn on or off the grid (1=on, 0=off)
xy_label_on=1;         % Turn on or off the x, y and z label (1=on, 0=off)
%#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=
```

Figure 3-31: Example of input parameters of *vaultdgn3.m*

The output provided from this code is almost identical to the output generated from “Unit-based design Algorithm” and an example of output is shown in Figure 3-32.

This parametric code allows for an initial form exploration of tensegrity structures by displaying geometric information in a 3D graphical environment.

```
==============   Given Data   =====================
* Structure Span = 10.0008
* Structure Hight = 2.0003
* Height of the Tensegrity Unit = 1
* No. of units along the cylindrical axis = 6
* No. of unit along the circumference axis = 4

=================   OUTPUT   ======================
* Structure Radius = 7.2503
* Structure opening angle = 87.2094(degree)
* Rotation Angle between units = 21.0483(degree)
* Angle of a BAR = 13.2792(degree)
* Length of a BAR = 3.4827

   -------- << For the Upper Base >> -----------
* Length of the Upper Base = 2.6384
* Upper based overlap along the cylindrical axis = 1.3192
* Upper based overlap % along the cylindrical axis = 50 %

* Upper based overlap along the circumference axis = 1.7869
* Upper based overlap % along the circumference axis = 67.7267 %

   --------- << For the Lower Base >> ---------
* Length of the Lower Base  = 2.4677
* Lower based overlap along the cylindrical axis = 1.7042
* Lower based overlap % along the cylindrical axis = 69.0614 %

* Lower based overlap along the circumference axis = 1.2042
* Lower based overlap % along the circumference axis = 48.7979 %
================================================
```

Figure 3-32: Example of output parameters from *vaultdgn3.m*

Figure 3-33 illustrates 4x6 models of vaulted tensegrity structures generated by method of case 1 of a structure-based design and the following data (Table 3-4) were used:

Table 3-4: Input data for 4x6 models of a structure-based design (Case 1)

Structure Span(W)	10 m
Structure Height(H)	2m
Height of the Tensegrity Unit(ht)	1m
Design method	Case 1
Overlap length of the upper base along the cylindrical axis	1.2m
Number of units along the cylinder axis	6
Number of units along the circumference direction	4

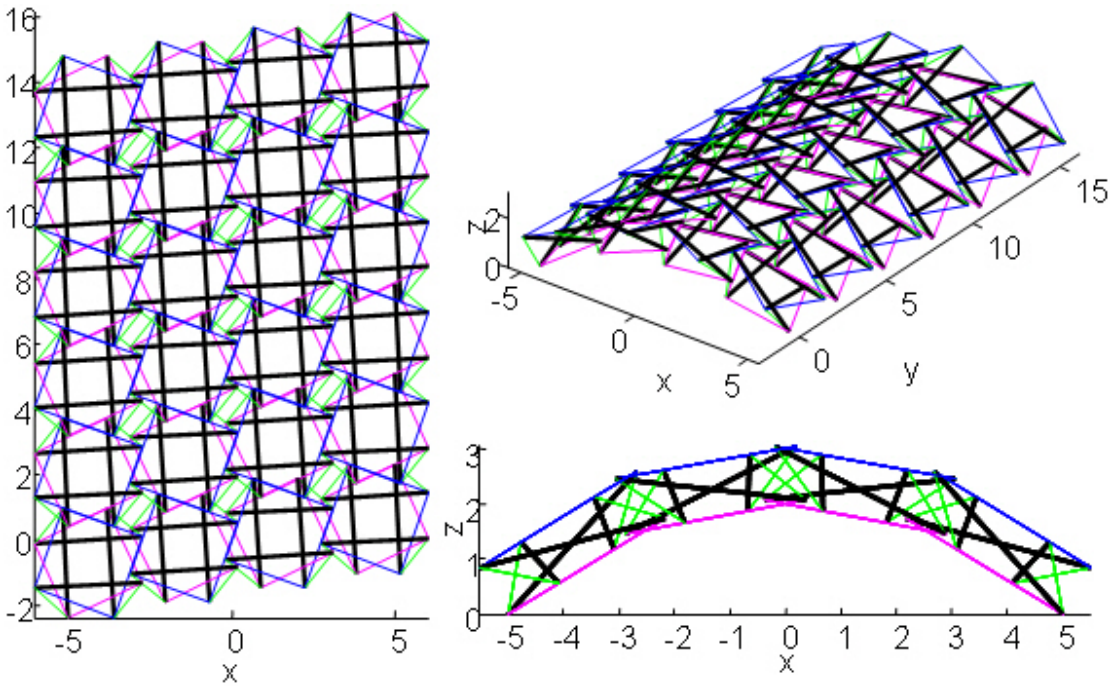


Figure 3-33: 4x6 model of a vaulted tensegrity structure generated by structure-based design (Case 1)

Table 3-5: Output of 4x6 structure (Case 1)

```
============== Given Data ====================
* Structure Span = 10
* Structure Hight = 2
* Height of the Tensegrity Unit = 1
* No. of units along the cylindrical axis = 6
* No. of unit along the circumference direction = 4

================ OUTPUT ======================
* Structure Radius = 7.2503
* Structure opening-angle = 87.2095(degree)
* Rotation Angle between units = 21.0514(degree)
* Angle of a BAR = 13.3347(degree)
* Length of a BAR = 3.472

 -------- << For the Upper Base >> -----------
* Length of the Upper Base = 2.5985
* Upper-base overlap along the cylindrical axis = 1.2
* Upper-base overlap % along the cylindrical axis = 46.1804 %

* Upper-base overlap along the circumference direction = 1.6746
* Upper-base overlap % along the circumference direction = 64.445 %

 --------- << For the Lower Base >> ---------
* Length of the Lower Base = 2.4907
* Lower-base overlap along the cylindrical axis = 1.8017
* Lower-base overlap % along the cylindrical axis = 72.3381 %

* Lower-base overlap along the circumference direction = 1.3066
* Lower-base overlap % along the circumference direction = 52.4585 %
```

Figure 3-34 illustrates 5x6 models of vaulted tensegrity structures generated by method of case 2 of a structure-based design method and the following data (Table 3-6) are used:

Table 3-6: Input data for 5x6 models of a structure-based design (Case 2)

Structure Span(W)	10 m
Structure Height(H)	2m
Height of the Tensegrity Unit(ht)	1m
Design Method	Case 2
Overlap percent of the upper base along the cylindrical axis	50%
Number of units along the cylinder axis	6
Number of units along the circumference direction	5

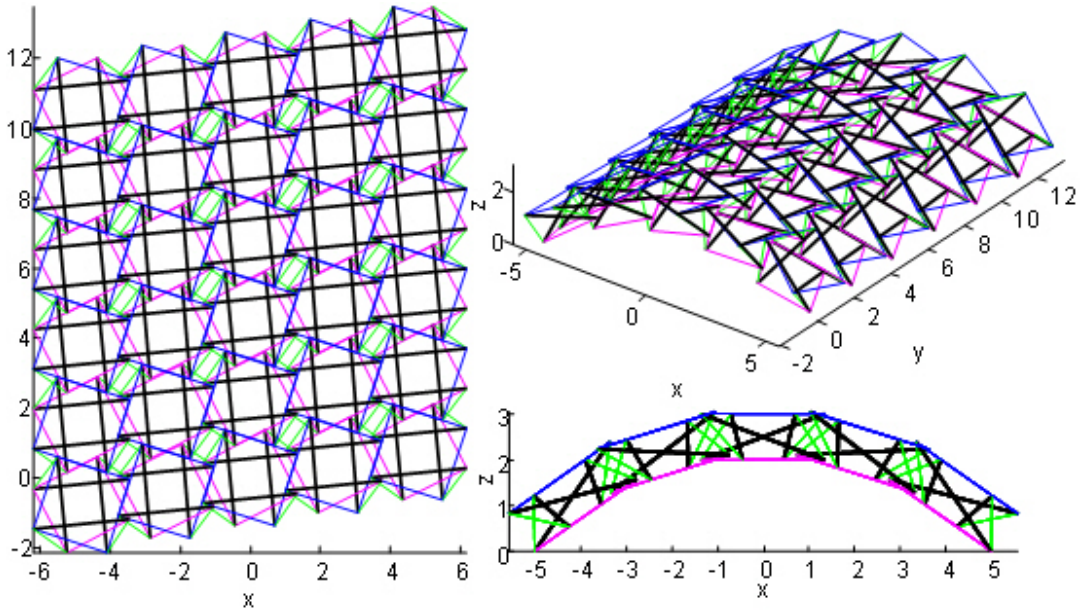


Figure 3-34: 5x6 model of a vaulted tensegrity structure generated by structure-based design (Case 2)

Table 3-7: Output of a 5x6 structure (Case 2)

```
============== Given Data ====================
* Structure Span = 10
* Structure Hight = 2
* Height of the Tensegrity Unit = 1
* No. of units along the cylindrical axis = 6
* No. of unit along the circumference direction = 5

================ OUTPUT ======================
* Structure Radius = 7.2503
* Structure opening-angle = 87.2094(degree)
* Rotation Angle between units = 17.4558(degree)
* Angle of a BAR = 16.0968(degree)
* Length of a BAR = 2.9116

 -------- << For the Upper Base >> -----------
* Length of the Upper Base = 2.1636
* Upper-base overlap along the cylindrical axis = 1.0818
* Upper-base overlap % along the cylindrical axis = 50 %

* Upper-base overlap along the circumference direction = 1.4681
* Upper-base overlap % along the circumference direction = 67.8523 %

 --------- << For the Lower Base >> ---------
* Length of the Lower Base = 2.0218
* Lower-base overlap along the cylindrical axis = 1.397
* Lower-base overlap % along the cylindrical axis = 69.1 %

* Lower-base overlap along the circumference direction = 0.98367
* Lower-base overlap % along the circumference direction = 48.6542 %
```

Figure 3-35 illustrates a 4x6 model of a vaulted tensegrity structure generated by case 3 of the structure-based design method and the following data (Table 3-8) are used:

Table 3-8: Input data for a 4x6 structure (Case 3)

Structure Span(W)	10 m
Structure Height(H)	2m
Height of the Tensegrity Unit(ht)	1m
Design Method	Case 3
Upper base length	2.8
Number of units along the cylinder axis	6
Number of units along the circumference direction	4

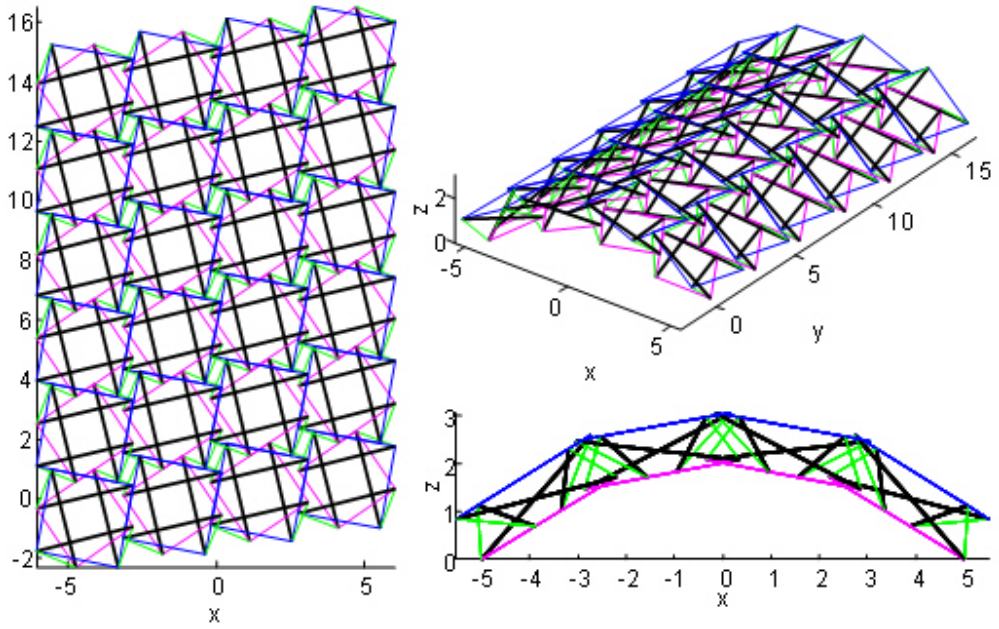


Figure 3-35: 4x6 model of a vaulted tensegrity structure generated by structure-based design (Case 3)

Table 3-9: Output of a 4x6 structure (Case 3)

```
============== Given Data ====================
* Structure Span = 10
* Structure Hight = 2
* Height of the Tensegrity Unit = 1
* No. of units along the cylindrical axis = 6
* No. of unit along the circumference direction = 4

================ OUTPUT ======================
* Structure Radius = 7.2496
* Structure opening-angle = 87.2194(degree)
* Rotation Angle between units = 21.1078(degree)
* Angle of a BAR = 13.4131(degree)
* Length of a BAR = 3.5172

 -------- << For the Upper Base >> -----------
* Length of the Upper Base = 2.8
* Upper-base overlap along the cylindrical axis = 1.8287
* Upper-base overlap % along the cylindrical axis = 65.3101 %

* Upper-base overlap along the circumference direction = 2.2646
* Upper-base overlap % along the circumference direction = 80.8801 %

 --------- << For the Lower Base >> ---------
* Length of the Lower Base = 2.3585
* Lower-base overlap along the cylindrical axis = 1.2747
* Lower-base overlap % along the cylindrical axis = 54.0477 %

* Lower-base overlap along the circumference direction = 0.75711
* Lower-base overlap % along the circumference direction = 32.1019 %
```

CHAPTER 4: HELICOID TENSEGRITY STRUCTURES

4.1 GEOMETRIC CONFIGURATION

A single-curvature geometry that presents lot of interest as a form but that involves a non-uniform curvature is that offered by a helicoid shape that fits on the curved surface of a cone. The square-based unit that was used in the configuration of a vaulted structure will be used for the helicoid shape as well.

For this study the shape originally selected was the conical. In the progress of this investigation it was found that the geometric constraints that occur from the use of a square-based tensegrity unit and the overlap condition related to the utilization of this unit do not allow the creation of the geometry of the cone. Specifically it was proven that the rotation axes of adjacent units cannot meet in one single point which should have been the vertex of the cone, if that construction were feasible. Indeed as Figure 4-1 shows, a small circle is generated from the tangent lines of the rotation axes. This geometric construction is due to the manner adjacent tensegrity units of a square-base overlap to create a rotation axis that connects the mid-point of all overlapping. Because of this geometric constraint, a helicoid tensegrity structures was deemed as a more appropriate geometric form for further investigation.

Because the helicoid structure is also considered a single-curvature structure, upper and lower-base overlaps of adjacent units along the radial direction should meet in order to allow the structure to bend one way only. In contrast to the features of the geometry of vaulted structures which can be constructed from identical units, the sizes of units in a helicoid structure should be reduced proportionally in the radial direction. Therefore, to create a helicoid the center points of adjacent tensegrity units need not only

to shift but also to rotate. Figure 4-2 shows flat configurations of helicoid structures in detail.

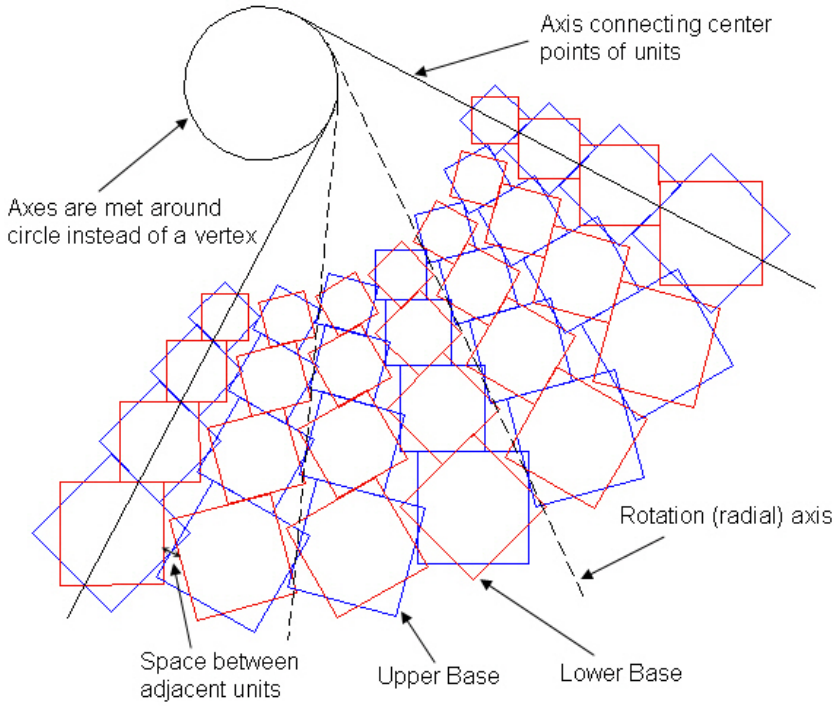


Figure 4-1: Geometric configurations of a helicoid structure

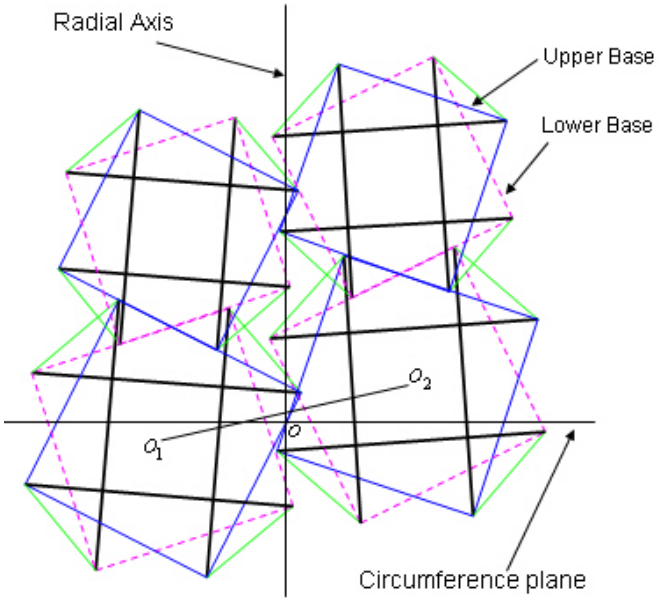


Figure 4-2: Flat configuration of a helicoid structure

Taking into account these observations and principles, the following conditions concerning the geometry of helicoid structures are summarized:

(a) The upper and lower bases of units along the radial direction are reduced proportionally to generate helicoid surfaces.

b) The center points between adjacent units across the radial direction must be shifted as well as rotated in order to create single-curvature helicoid tensegrity structures.

c) The rotation (radial) axis is not perpendicular to the line connecting two center points O_1, O_2 along circumference direction.

d) The line connecting two center points O_1, O_2 does not pass through a mid point o of the overlap because of a rotation of the center points followed by shifting.

e) The amount of upper and lower-base overlaps between adjacent units along the radial axis can be determined from geometric configurations.

f) The amount of overlap between adjacent units varies along the circumference direction, due to the proportional reduction.

g) The gap between lower bases should be kept constant along the circumference direction.

4.2 GEOMETRIC LIMITATIONS

In a helicoid structure, there also exists geometric limitation for the overlap of an upper or lower base. Figure 4-3 shows respective geometric limitations of the overlap for the upper and lower bases. A zero lower overlap configuration is possible when the center line passes through the vertex of the lower base, as shown in Figure 4-3(a). On the other hand, if the center line passes through the vertex of the upper base, as shown in Figure 4-3(b), a zero upper-base overlap can be created. In practice, the center line must fall within these two to create a certain amount of an overlap.

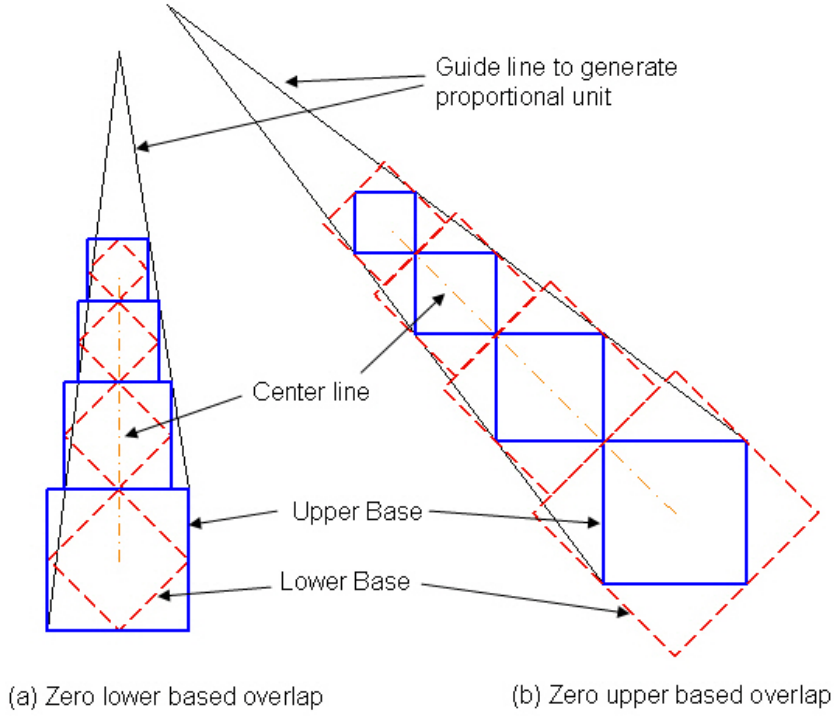


Figure 4-3: Geometric limitation for zero upper or lower-base overlap

4.3 DESIGN ALGORITHM FOR HELICOID TENSEGRITY STRUCTURE

The design process for a helicoid structure requires calculating many intersection points in 3D space, and the algorithm developed in the previous Chapter 3.4 can be used to find intersection points for a helicoid structure. The main difference between helicoid and vaulted tensegrity structures is the unit size of the helicoid is reduced proportionally along the radial axis. Moreover, keeping lower gaps between adjacent units constant along the circumference is important because the irregularities of lower gaps do not allow them to adjoin one another to create the curvature.

4.3.1 Guide Lines for Helicoid Tensegrity Structures

The design process for helicoid structures starts by creating units with the size decreasing proportionally along the radial axis. To ensure proportional decrease in unit sizes, two guide lines with a certain angle are drawn. Tensegrity units will be created on the inside of these two lines, starting from the largest unit in size. The ratio of reduction in unit size (thus the actual size of individual units) depends on the angle of two guide lines. A high reduction ratio is associated with a wide angle between guide lines, and will cause a high stress to the overlap cables of the upper and lower bases between adjacent units along the circumference direction. The reverse is true for a low reduction ratio. Accordingly, the angle between guide lines should be taken into account when determining the unit size for helicoid structures. Figure 4-4 shows smooth and rough connection with a small angle δ_2 and a large angle δ_2, respectively. Although the largest units in Figure 4-4 (a) and (b) are equal in size, the structure in Figure (a) should have a larger radius that that of Figure (b).

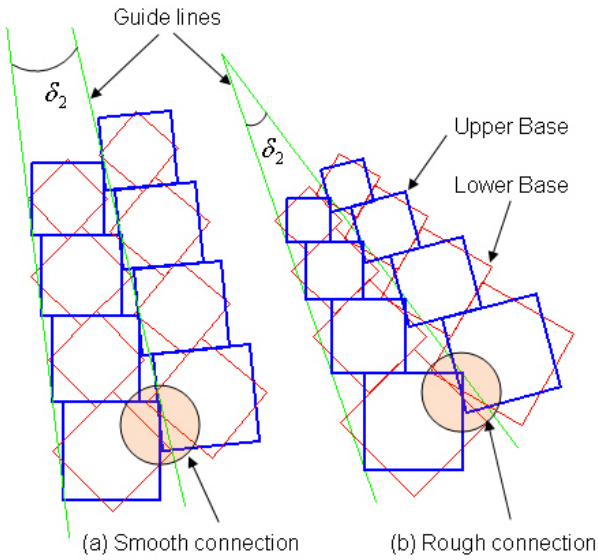


Figure 4-4: Geometric configurations with different angles δ_2 between guide lines

4.3.2 Tensegrity Column along the Radial Axis

Based on the guide lines, tensegrity units are drawn from the largest to the smallest unit. All upper bases of units are generated before the lower bases since lower bases can be created only after drawing a center line that connects center points between the largest and smallest units. The scheme in Figure 4-5 shows the geometric configuration of a tensegrity column along the radial axis. The initial point P_1 for the largest unit can be determined from a given radius of the structure, and the point P_2 on the guide line B is determined by a point where the 45° diagonal starting at P_1 meets with the guide line B. The angle δ_1 plays an important role in this design process deciding the overlap length of units along the radial axis.

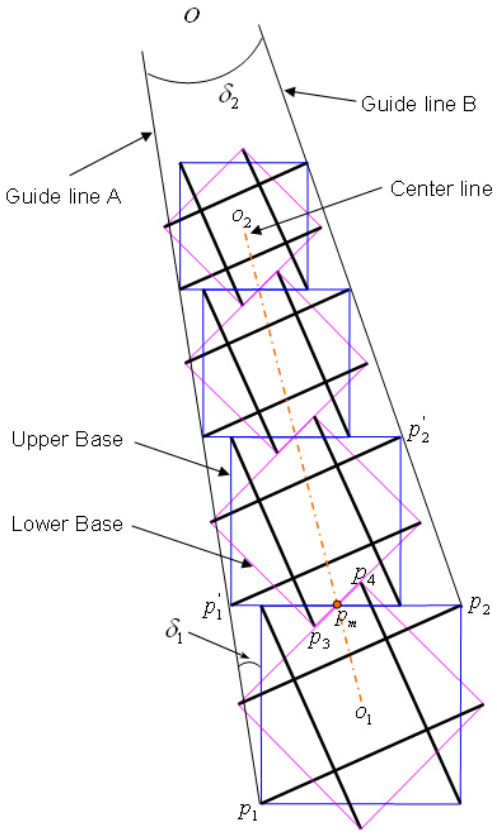


Figure 4-5: Geometric configuration of a tensegrity column along the radial axis

In order to place the next largest unit, the point $P_1^{'}$ is found by intersecting the guide line A with a horizontal line drawn from the point P_2. The point $P_2^{'}$ is then obtained in a similar manner as before. Upper bases of the remaining units are generated by following this iterative process.

To be able to create lower bases, center points of upper bases should be connected to form a center line. This center line intersects the largest upper base at P_m. Drawing a 45° line passing through P_m delineates the largest lower base. The actual size of the largest lower base is then given by selecting the point P_4 on the 45° line. The remaining lower bases can be created in a similar manner.

4.3.3 Overlap Conditions

When the first column is generated as described above, another column can be created next to the first with some overlap. All 3D coordinates of the second column are obtained by translating and rotating the first column. At this point, it should be decided whether the overlap between adjacent units will initiate at the largest or the smallest unit. If the overlap starts from the largest unit, the smallest unit may end up with no overlap. This can happen since the actual size of the units in the radial direction is dictated by the rule of proportional reduction. Figure 4-6 (a) illustrates this case in which no overlap occurs at the smallest unit. Thus in helicoid structures, initiating overlap at the largest unit is not recommended. Overlapping should start at the smallest unit if every unit is to have some overlap, as illustrated in Figure 4-6 (b) and (c). Especially, Figure 4-6 (c) shows the geometric configuration where the smallest unit has zero overlap.

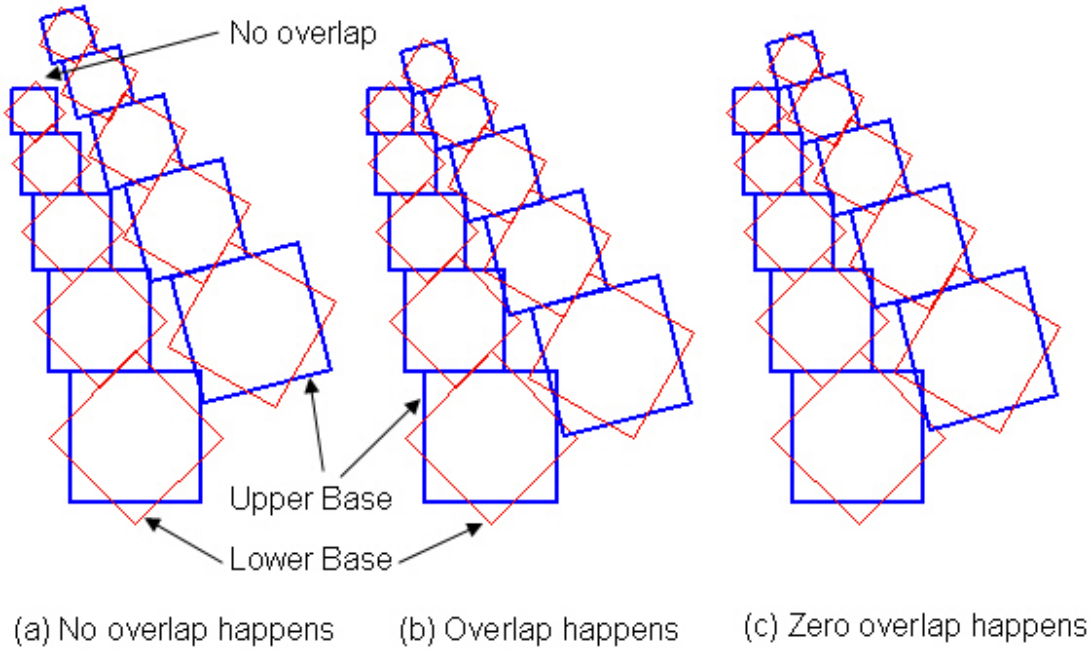


Figure 4-6: Geometric configurations with different overlap conditions

4.3.4 Creating Identical Gaps

Creating identical gaps between adjacent units along the radial direction is most critical to designing a helicoid structure; otherwise the constant rotation angle between units cannot be guaranteed. As shown in Figure 4-7, two points S_0 and S_m are necessary to form a rotation axis which will be used to make sure the identical gap between adjacent units. The point S_0 is a mid point of the upper-base overlap and serves as an origin of two circles passing S_1 and S_2, respectively. Drawing a new circle that falls exactly half way between the existing two circles gives the point S_m.

The gap between units determines the rotation angle and is affected by the following conditions:

• A small overlap of the upper base of the smallest unit will increase the gap.

• Theoretically, the maximum gap given the upper-base overlap can be generated if the angle δ_1 is set to zero.

• Increasing the angle δ_1 reduces the gap.

• The gap depends on the number of units along the radial axis and the angle δ_2 between guide lines. A smaller number of units or a smaller angle δ_2 can generate a larger gap (an instance with a smaller angle is illustrated in Figure 4-4).

• The rotation angle between adjacent units is affected by the unit height.

• The zero overlap configurations can theoretically generate the largest gap.

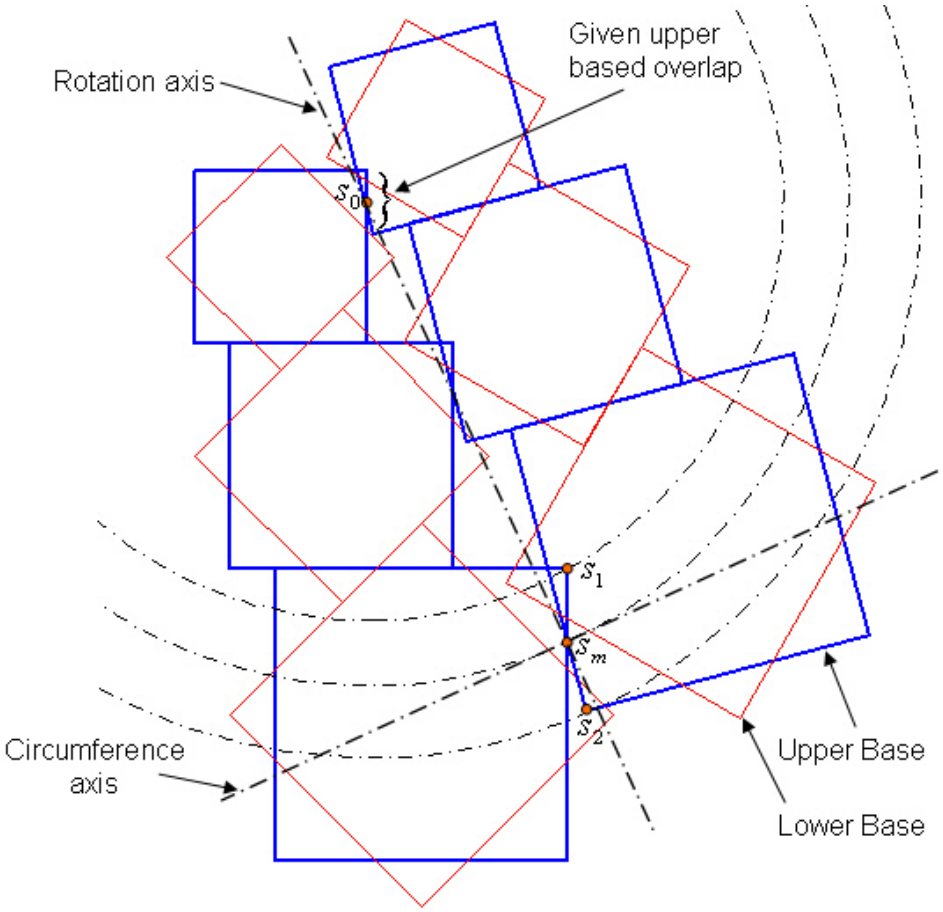


Figure 4-7: Creating identical gaps

4.4 GRAPHICAL REPRESENTATION OF HELICOID STRUCTURES

The code implementing the above algorithm has been developed in a MATLAB software environment, and has been called ***Helicoid_dgn_v2.m***. The code takes into account all interrelated parameters and constrains, and allows the user to modify the following inputs: a) angle between guide line and unit, b) angle between two guide lines, c) structure radius, d) overlap percentage of the upper base, e) height of the tensegrity unit, and f) number of units along the radial and circumference direction. Figure 4-8

shows an example of ***Helicoid_dgn_v2.m*** input.

```
%  ======================================================================
%  =   Helicoid_dgn_v2.m   By Jinman Kim on 03/11/04                    =
%  =   To design generate helicoidal tensegrity networks                =
%  =                                                                    =
%  =   The Input data for this algorithm are below                      =
%  =   deltal = Angle between guide line and unit                       =
%  =   delta2 = Angle between guide lines                               =
%  =   R = Structure Radius                                             =
%  =   ovt_P = Overlap percentage of the Upper base                     =
%  =   ht  = Height of the Tensegrity Unit                              =
%  =   num_unit_l = The number of units along the radial axis           =
%  =   num_unit_r = The number of units along the circumference axis    =
%  ======================================================================

clear all;      % Clear all previous data
%---------------------------------------------------------------------
deltal=8;     % Angle between guide line and unit (degree)
delta2=18;    % Angle between guide lines (degree)
R=40;         % Structure Radius
ovt_P = 20;   % Overlap percentage of the Upper base
ht = 4;       % Height of Unit Tensegrity

num_unit_l = 4;     % # of unit along Radius
num_unit_r = 12;    % # of unit along Circumference

% #=#=#=#=#   Control DISPLAY OPTION   #=#=#=#=#
Disp_Node_No_TOP=0;   % Option to display Node Number of TOP square, (1=on, 0=ff)
Disp_Node_No_BOT=0;   % Option to display Node Number of Bottom square, (1=on, 0=ff)
Disp_BAR = 1;         % Option to display Bar on or off, ( 1=on, 0=off)
Disp_Lateral_cable=1; % Option to display Cable on or off, ( 1=on, 0=off)
grid_on = 1;          % Option to turn on or off the grid (1=on, 0=off)
xy_label_on = 0;      % Option to turn on or off the x, y and z label (1=on, 0=off)
%---------------------------------------------------------------------
```

Figure 4-8: Input parameters of *Helicoid_dgn_v2.m*

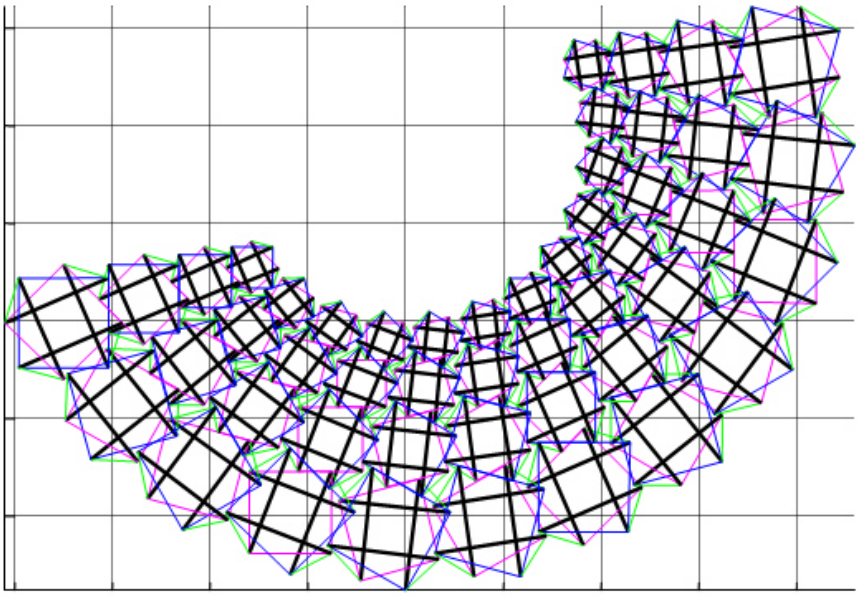


Figure 4-9: 4x12 model of a helicoid tensegrity structure before unit rotation

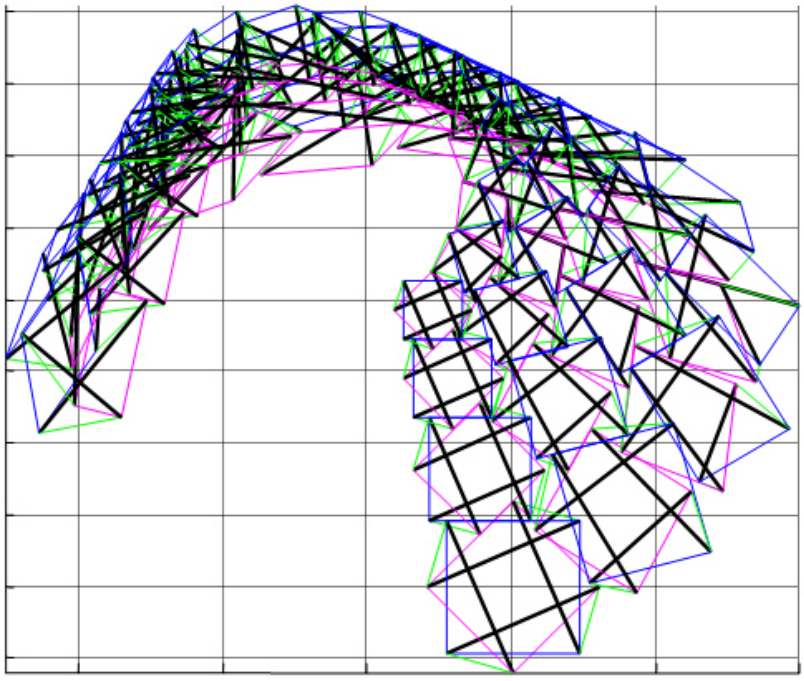


Figure 4-10: 4x12 model of a helicoid tensegrity structure (Isometric view)

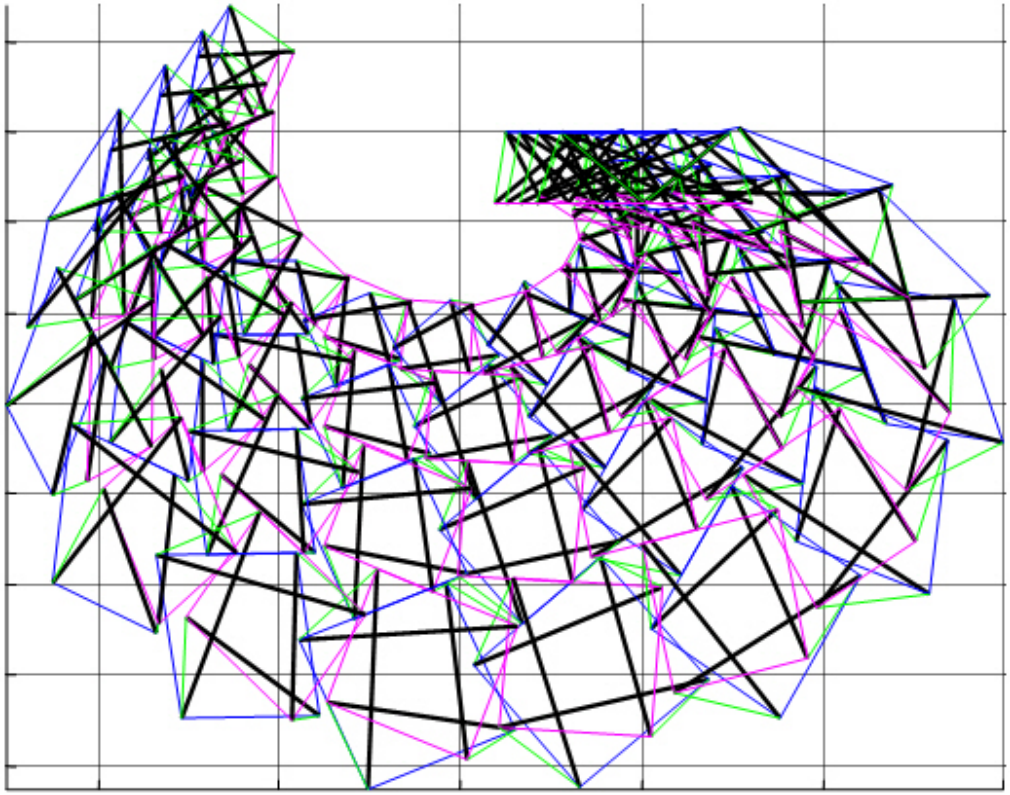


Figure 4-11: 4x12 model of a helicoid tensegrity structure (Top view)

CHAPTER 5: INTEGRATION OF PARAMETRIC MODELS WITH STRUCTURAL ANALYSIS

5.1 INTRODUCTION

Continuous tension cables and discontinuous compression bars classify tensegrity structures as highly nonlinear structures. Due to nonlinearity pre-stressed or externally loaded configurations of tensegrity structures are typically different from those of initial geometry. Therefore, the structural analysis of tensegrity structures is essential when their geometric form is to be considered in real world applications.

Tassoulas (Tassoulas, 2003) developed a FORTRAN code called "NONSA0" (Nonlinear Structural Analysis) for nonlinear analysis, and like most analysis programs, this code requires various input data such as (1) material properties, (2) element 3D node coordinates, (3) element connectivity, and (4) applied load information. However, creating a new input file each time the design algorithm generates a new geometry, which may actually reflect just a minor change in anyone of the interrelated parameters, would render the parametric approach proposed in this work a rather impossible application. Moreover, the Geometric design algorithms and the analysis code are developed in different software environments. NONSA0 is written in FORTRAN, and the Geometric design algorithms are developed in MATLAB environments. As such, it is necessary to integrate 3D coordinates generated from MATLAB into the analysis program as a part of an input. The codes developed here provide parametric models of pre-stressed or loaded tensegrity structures and support visualization of the flow of stresses within the structure.

The visualization environment that has been developed is intended to become a very useful tool that will allow the designer to compare the initial conceptual geometry to

more realistic pre-stressed configurations. A structural analysis study to determine the effect of each one of the interrelated geometric parameters to the configuration of tensegrity structures has already been conducted by Liu (Liu, 2004) who used both "NONSA0" and MATLAB codes ("*deform2.m*" and "*make_input70103.m*") developed in this research. Detailed results of this analysis are not discussed in this chapter.

5.2 NONLINEAR STRUCTURAL ANALYSIS PROGRAM (NONSA0)

Tensegrity structures require geometric nonlinear techniques for the analysis. NONSA0 is a nonlinear structural analysis code developed by Tassoulas (Tassoulas, 2003) and adapted by the same author for the analysis of tensegrity structures. This program is based on the direct stiffness method (DSM) and employs an iterative analysis procedure using Newton-Raphson method to account for the geometrically nonlinear behavior of tensegrity structure. The loads (external loads and weight) are subdivided into a series of load increasements and applied as sequence of load increasement then iterate to convergence for each load level.

NONSA0 can determine pre-stress values on all members of the tensegrity structure by taking into account bar elongation values; this feature of the code is consistent with the actual method of pre-stressing tensegrity units. Internal forces, support reactions, and displacements are calculated by NONSA0. More detailed notes on the method in NONSA0 are provided in Appendix A.

The material properties chosen for this study are close to those used in the physical models built by Liapi (Liapi, 2000) and listed in Table 5.1. Bar specifications are in conformance with ASTM A543 Grade B or A501 standard weight steel pipe and structural tubing. Cable specifications are in conformance with ASTM 475-98 for steel wires (Liu 2004). English units have been used for materials and geometric data.

Table 5-1: Material properties of bars and cables (Liu 2004)

Bar	
Area	1.07 in^2
Diameter (Outside)	2.375 in
Diameter (Inside)	2.067 in
Type	Steel
Unit Weight	0.2843 lbs/in^3
Young's Modulus	3.00E+07 lbs/in^2
Moment of Inertia	0.666 in^4

Cable	
Area	0.0767 in^2
Diameter	5/16 in
Breaking Load	8000 lbs force
Grade	High S
Unit Weight	0.2227 lbs/in^3
Young's Modulus	2.70E+07 lbs/in^2

5.3 IMPORTING 3D COORDINATES INTO STRUCTURAL ANALYSIS

All numerical 3D node coordinates generated by the parametric design algorithms are converted into a format that can be read by the analysis code to study the behavior of the structure. The input file of "NONSA0" specially developed for the analysis of tensegrity structure is text based and requires many sets of input data such as (1) material properties, (2) element 3D node coordinates, (3) element connectivity, and (4) applied load information as shown in Figure 5-1. Among these data, element 3D node coordinates and element connectivity are classified as geometric data and consist the most extensive section of the input file. In general, many case-by-case studies should be implemented to investigate the effects of design parameters under the pre-stressed or loaded form, and their geometric data should be modified whenever different initial geometric shapes are generated from the use of the parametric design algorithms.

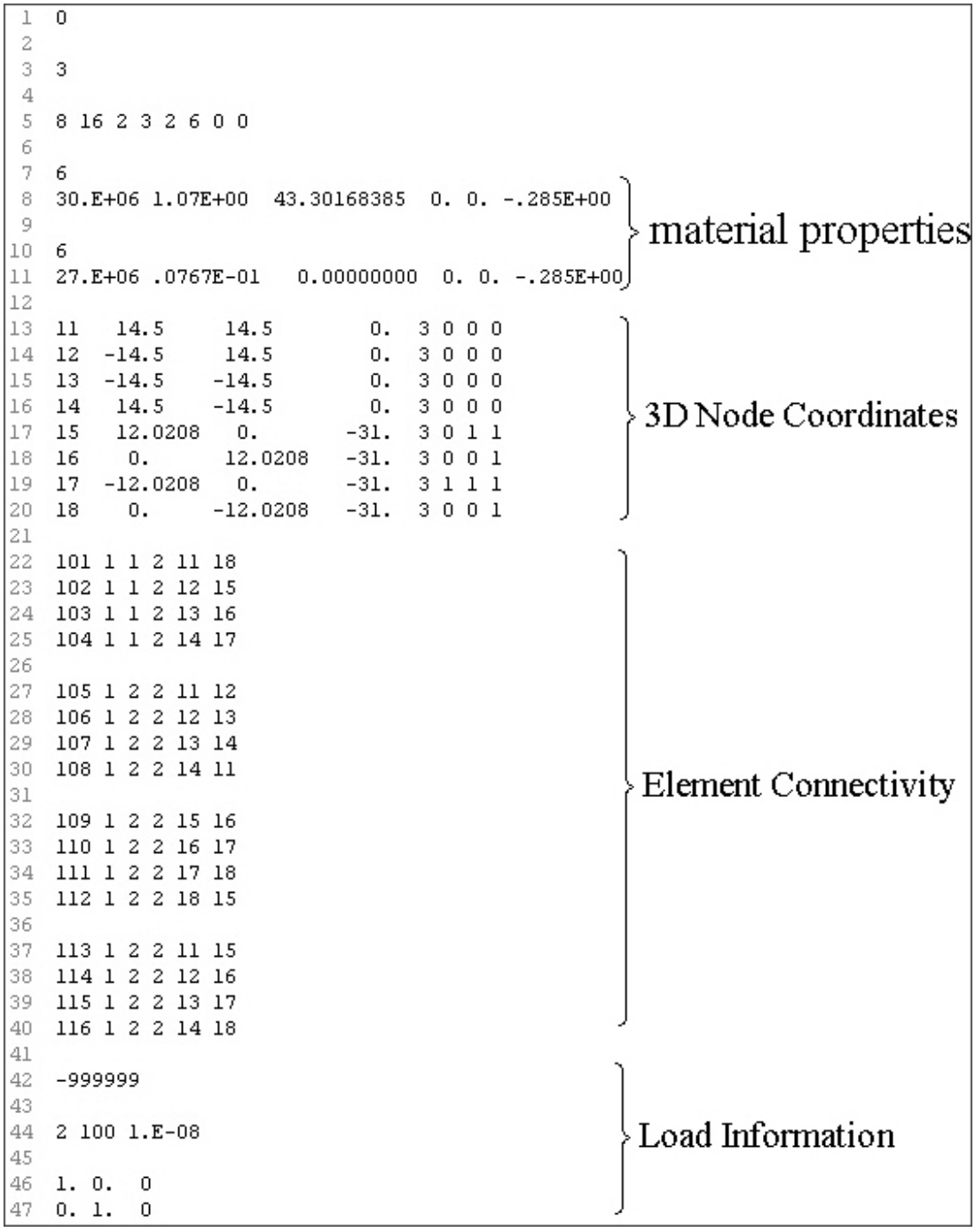


Figure 5-1: Sample input data for structure analysis

Each line of the input file in Figure 5-1 is described in detail as follows.

• Line 1: Indicates the index for start or restart of the program. 0 indicates that the code generates new results.

• Line 3: Indicates the number of dimensions.

• Line 5: Consists of the following information:

- Number of nodes
- Number of elements
- Number of materials
- Maximum number of degrees of freedom per node
- Maximum number of nodes per element
- Maximum number of constants per material
- Maximum number of integration points
- Maximum number of state parameters

• Line 7 – 11: Indicates material properties.

• Lines 7 and 10: Represent the number of constants for each material.

• Lines 8 and 11: These lines consist of the following information:

- Young's modulus (lbs/in^2)
- Member's cross section area (in^2)
- Initial length of each member (in)
- Gravity of materials

• Line 12 – 20: This section represents the node information and each line consists of:

- Node label
- X, Y, Z coordinates
- Number of DOF per node

- Node constraints (0: node is allowed to displace and 1: node is supported)

• Line 22 – 40: This section represents the element connectivity and each line consists of:

- Element label
- Element type
- Material type
- Number of nodes per element
- First node label
- Second node label

• Line 42 – 47: Indicates load information

• Line 44: Consists of the number of load steps, maximum number of cycle at each step, and tolerance.

In resolving this tedious process and integrating previous design algorithms into the analysis code, the code "*make_input70103.m*" plays an important role. The "*make_input70103.m*" takes all 3D nodes coordinates data that are generated by the design algorithm code and automatically converts them into nodal data and element connectivity data, thereby creating an input file for the analysis code. Since material properties and loading information are not included in design algorithms they would have to be manually modified if any change occurs. The sections of the input file that need to be manually modified are shown in Figure 5-2.

```
clear all;

load all_data    % load geometric data

totaln=num_unit_l*num_unit_r;

nn=8*totaln;      % Total # of node

% Total # of element
numel=24*(num_unit_l-2)*(num_unit_r-2)+22*((num_unit_l-2)*2+ ...
      (num_unit_r-2)*2)+20*4;

file_name='testl.txt';     % output file name for analysis
% ==============================
%     Manual Input Data      =
% ==============================
nmat=2;           % # of material
ndof=3;           % # of dof
mnode=2;          % Maximum # of node
mncm = 6;         % Maximum # of constrain material
mnip=0;           % Maximum # of iteration port
mnsp=0;           % Maximum # of state parameter

% Materal Input
ncm1=6;           % # of constant for the first material
ncm2=6;           % # of constant for the second material
ym1=30e6;         % Young's modulis for the first material
ym2=27e6;         % Young's modulis for the second material
csar1=1.07;       % Cross section area for the first material
csar2=0.0767;     % Cross section area for the second material
e01=lng_Bar+lng_Bar*1/100;   % Initial length for the first material
e02=0;            % Initial length for the second material
grav=[0; 0; -0.285];  % Gravity of material
% ------------------------------------------------------
```

Manual Input Part

Figure 5-2: Sections of input file that require manual modification

5.4 VISUALIZATION OF DISPLACEMENTS AND FLOW OF STRESSES

The output file from "NONSA0" contains a new pre-stressed or loaded form of tensegrity structures in Cartesian coordinates, as well as internal forces and support reactions subjected to external loads. The pre-stressed shape in the output can be very different from the initial geometric design due to the nonlinear nature of the structure.

Therefore, the output provides very informative data for the designer or engineer to explore the behavior of the tensegrity structure.

However, all of these outputs are composed of text based numerical values like the input file. These data can be analyzed without difficulty matching each numerical value with corresponding node or element, as far as a small number of units are concerned. But this is not the case with the tensegrity structure composed of many units, and it would be extremely difficult to match each numerical value of displacement or internal force with the corresponding node point or element of geometry, respectively. Thus, visualizing displacements of nodes and member forces from the numerical data is very instrumental in properly analyzing large tensegrity structures.

From this rationale, a MATLAB code "*deform2.m*" was developed to provide a parametric visualization of the pre-stressed or loaded form, and can be a useful tool for the designer to compare the initial conceptual geometry to the actual pre-stressed configuration. This program is executed by preparing input data from two sources, i.e., design algorithms and the analysis code "NONSA0". Geometric data come from design algorithms, including all 3D coordinates and general information like the number of units along the cylindrical and circumference axes, and numerical data come from the "NONSA0", including node displacements and internal forces of members.

Although the analysis code saves all data over the entire process of loading step, only those resulting from the final loading step are used by the *deform2.m* code for visualization. This final stage output can be obtained automatically by using the routine of searching inside *deform2.m* and sorted as displacement and force data. Next the visualization code organizes input data drawn from "NONSA0" into the same format as in design algorithms so that it can combine geometric and numerical data. Finally, initial and final deformed shapes are displayed together, and the magnitude of the stress

assumed by each member is visualized in different colors. Figure 5-3 is an example display of initial and final deformed geometries that the visualization code produced. Solid and dashed lines indicate the initial geometric configuration and deformed shapes, respectively. It is obvious that the deformed shape depends on the amount of pre-stress and external loading condition.

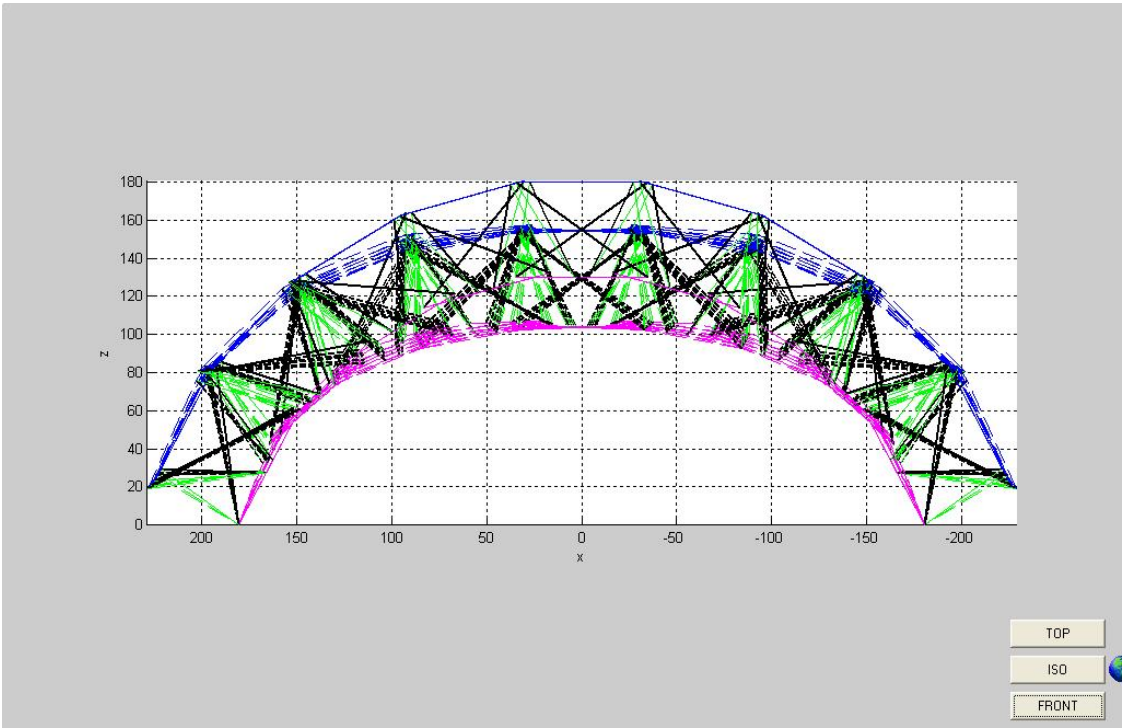


Figure 5-3: Example of initial and final deformed shape

Similarly, visualizing the flow of stresses within a tensegrity structure can be displayed as shown in Figures 5-4, 5, 6 and 7, providing further insights into tensegrity behavior. Stresses acting on each structural member are classified into compression and tension which are represented by two different color schemes. These color bars show the maximum and minimum of the stresses with actual numerical values. Depending on the

range of stresses to be displayed, a specific color is associated with a particular magnitude of stress, and the elements that assume a certain amount of stress are displayed in the corresponding color. This visualized output also can be seen in various views including top, isometric, and front view.

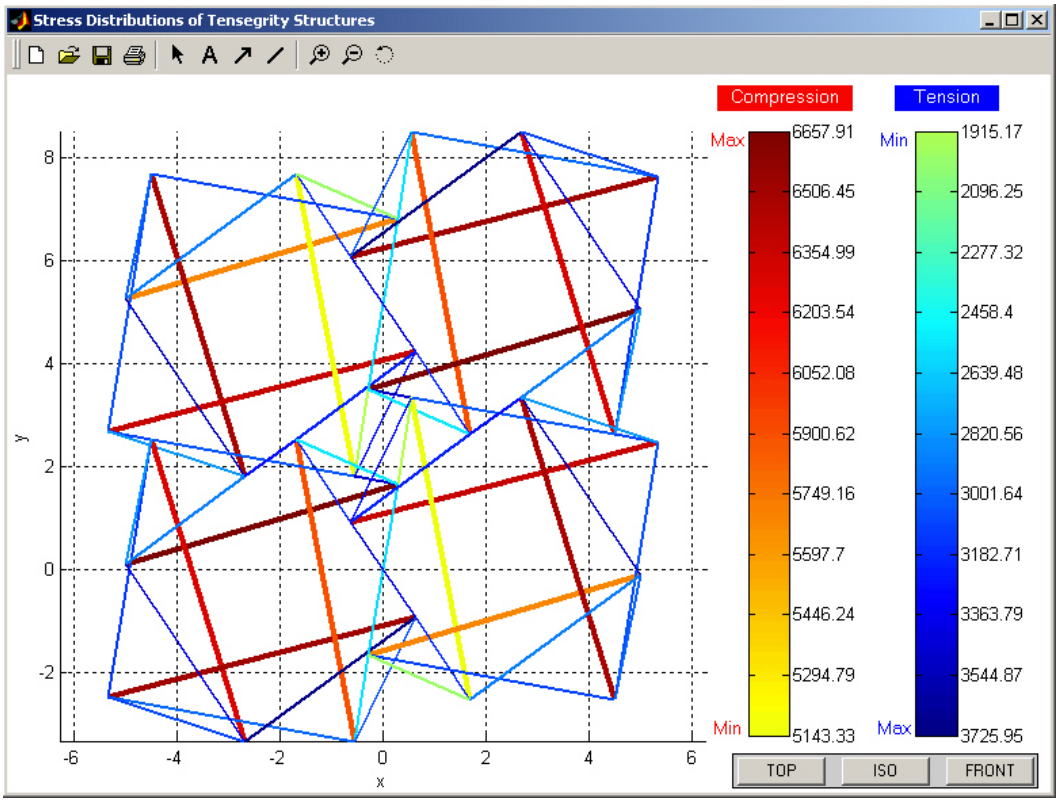


Figure 5-4: Flow of stresses in a 4-unit structure (Top view)

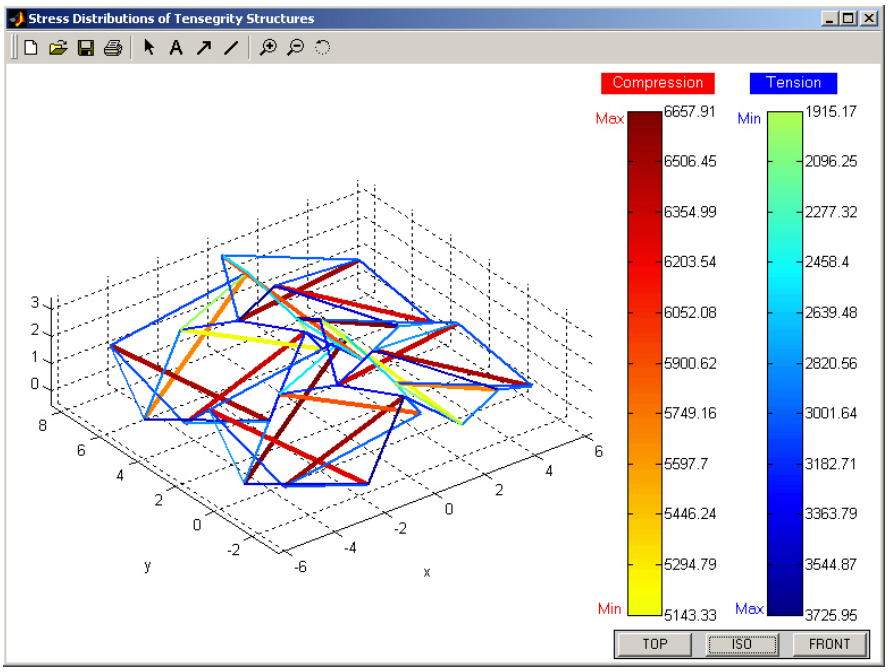


Figure 5-5: Flow of stresses in a 4-unit structure (Isometric view)

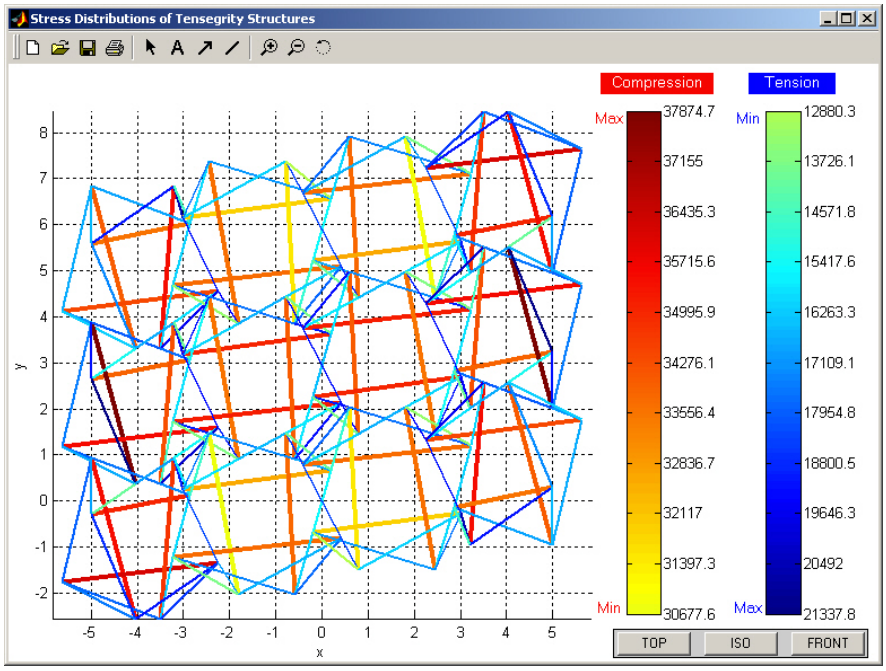


Figure 5-6: Flow of stresses in a 12-unit structure (Top view)

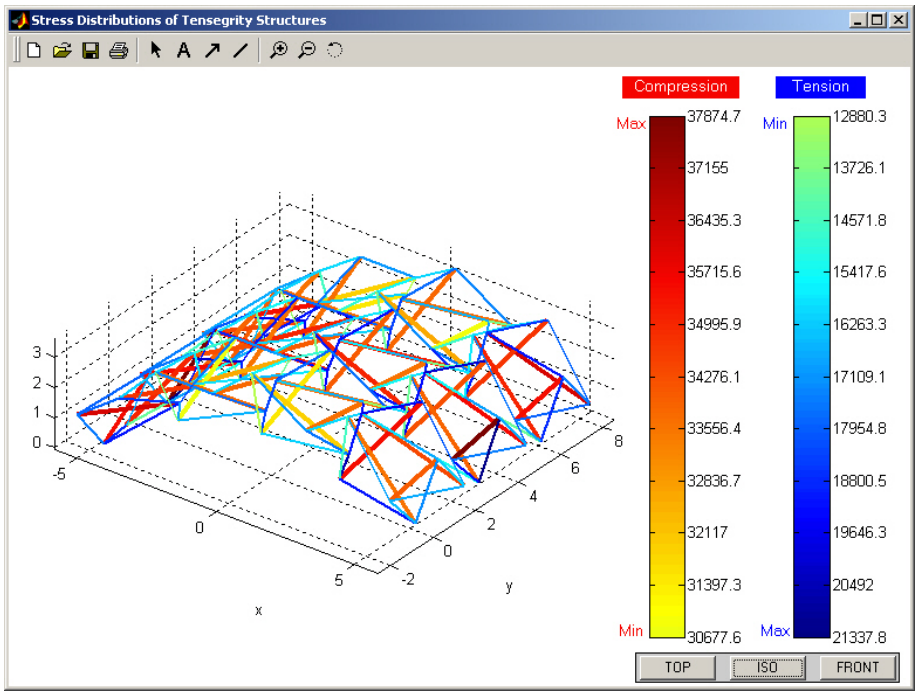


Figure 5-7: Flow of stresses in a 12-unit structure (Isometric view)

CHAPTER 6: VISUALIZATION IN A CAD ENVIRONMENT

6.1 INTRODUCTION

Since designers and engineers indicate a strong preference towards user-friendly commercially available CAD software, it is necessary that the spatial model that occurs from the application of the parametric algorithms, as well as the model that displays the final pre-stressed configuration be imported and available for access in a CAD environment. This will render parametric design a useful tool to designers and engineers.

Microsoft Visual Basic for Applications (VBA) is one of the most widely used programming languages in Visual Basic and other applications that take advantage of the Windows environment and allows for task automation. Indeed, in addition to Microsoft Office applications, several other products that feature VBA can provide their own programming environment for VBA, which is similar to the one found in Visual Basic. VBA can be used to extend and customize the host application instead of creating standalone applications in a VBA host application. Hence the user can develop VBA codes, for instance, for adding new dialog boxes, creating toolbar buttons, or automating procedural tasks.

MicroStation V8 provides the VBA environment as a programming option, which can extend and enhance the existing MicroStation environment which recognizes MDL codes, macro codes, windows functions, and visual basic codes. MVBA (MicroStation VBA) allows for rapid development of custom tools and applications within MicroStation. In addition, the VBA environment in MicroStation allows other applications to access files originated in MicroStation.

This Chapter describes a VBA code developed to combine the MATLAB with the MicroStation environment. Although all 3D node coordinates generated from the design algorithms are visualized in the MATLAB environment, it is necessary to transfer these 3D coordinates to a CAD environment so that the architect and engineer can access and explore the structures in CAD environment in the form of a 3D solid models. The VBA code developed for this purpose is the "*Tensegrity_DGN_V1.mvba*", the user form of which is shown in Figure 6-1. This VBA code serves as a bridge between MATLAB and MicroStation, importing final geometry from MATLAB and creating 3D solid models automatically in MicroStation. Thus, by integrating MATLAB into the MicroStation environment, the exploration and study of 3D tensegrity models in the CAD environment has been made possible.

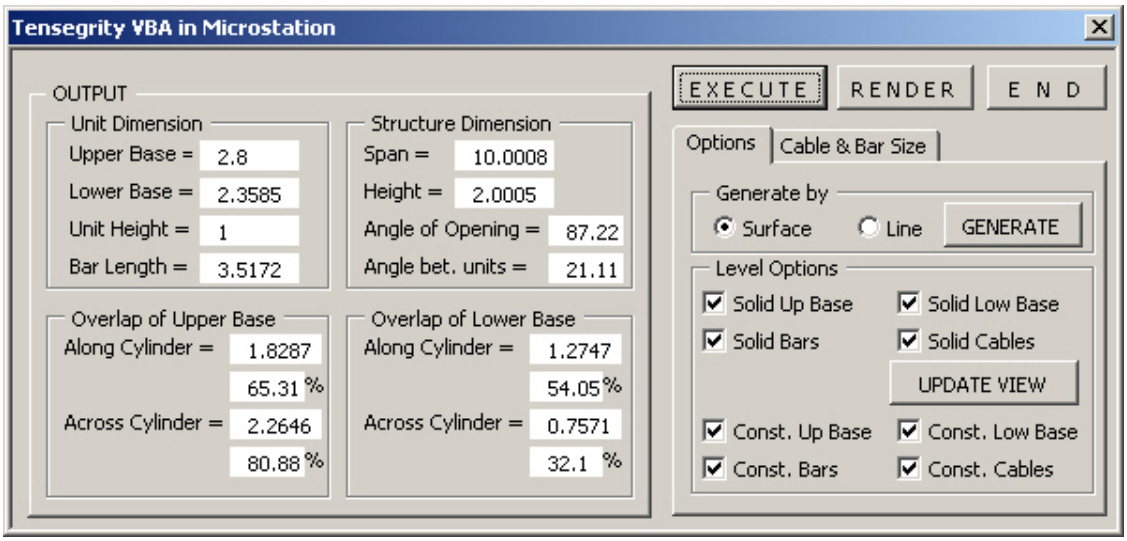


Figure 6-1: "*Tensegrity_DGN_V1.mvba*" applet on MicroStation VBA

6.2 VBA CODES TO INTERFACE MICROSTATION AND MATLAB

Developing a VBA tool in MicroStation requires development of macro commands and VBA programming. VBA macros can be stored, reusable, and run as many times as necessary. This section describes the overall procedure for developing macros to facilitate interfacing between MicroStation and MATLAB, and provides a detailed description of the most important steps.

The VBA code starts with the routing that opens the text file containing the 3D coordinates of a tensegrity structure under consideration:

```
On Error GoTo ErrOut
dlgDialog.FileName = ""
dlgDialog.Filter = "TXT File(*.txt)|*.txt"
dlgDialog.DialogTitle = "Open Input File for Visualization of Tensegrity Structure"
dlgDialog.ShowOpen
```

The module "dlgDialog" is the Microsoft Common Dialog Box that handles several common tasks of communicating. This pre-programmed object, "dlgDialog", is used to open, write, or print files. The last command line "dlgDialog.ShowOpen" opens a new window, shown in Figure 6-2, where the user can search and select the input file that originated from the design algorithms and can be visualized in MicroStation.

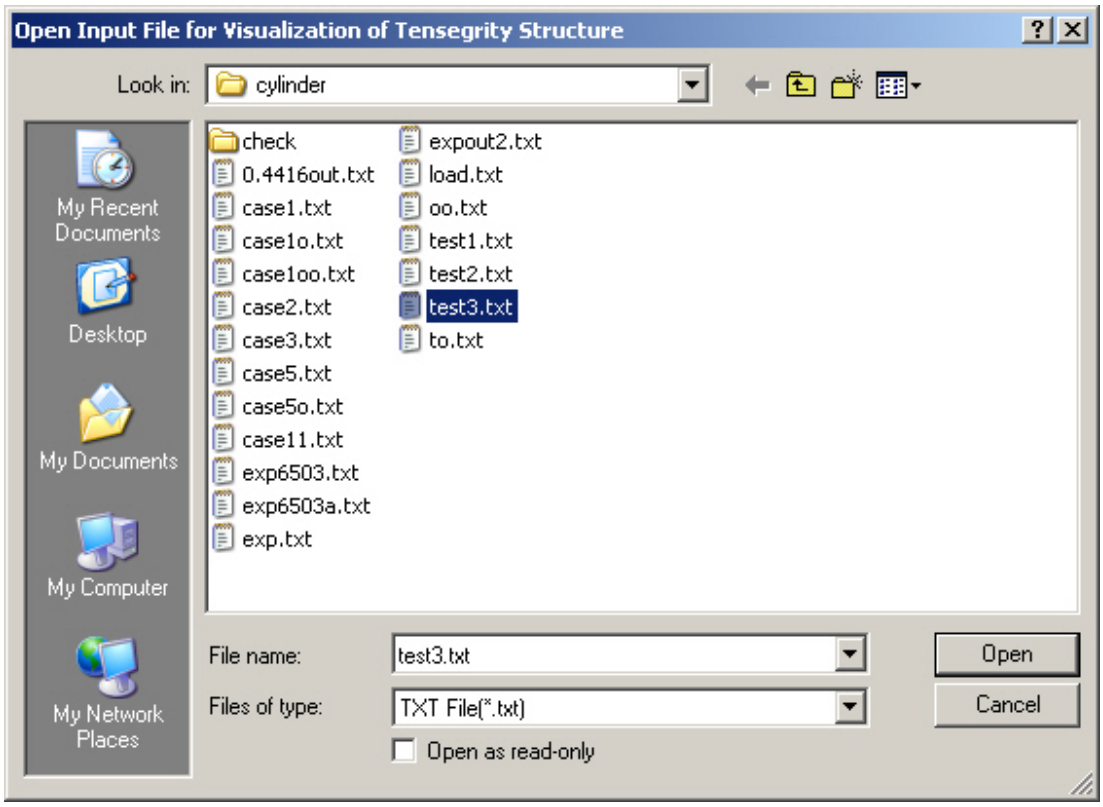


Figure 6-2: Dialog box to select an input file for visualization

The next step is to read the input file line by line through for-loops and to save the contained numerical data in a matrix form as the variable "All_ten_points". The input file is text based and contains a list of numerical data about the unit and structure in the first line, including the number of units along the circumference and cylinder's direction, the length of upper and lower bases, unit height, bar length, upper and lower-base overlap lengths in measurement units and in percentage, etc. All 3D coordinates are listed from the second line to the end of the input file. The code for performing this routine is listed below. The function "Split( )" is used to split a string of numerical data delimited by spaces, into a variant array which will form a matrix.

```
Open dlgDialog.FileName For Input As #1
Do While Not EOF(1)
    If i = 1 Then
        Line Input #1, Dummy
        X = Split(Dummy, " ")
        num_r = X(0): num_l = X(1):    UP_B = X(2):    LOW_B = X(3)
        ht = X(4):      lng_BAR = X(5): ovt = X(6):      ovtP = X(7)
        Top_ovt_L = X(8):      Top_pct_L = X(9):          dist_W = X(10)
        HH = X(11): theta_D = X(12): del_D = X(13)
        Btn_ovt_R = X(14): Btn_pct_R = X(15): Btn_ovt_L = X(16)
        Btn_pct_L = X(17)
        ReDim All_ten_points(8, num_l * 3, num_r)
    Else
        If jj > 8 Then
            jj = 1
            k = k + 1
        End If
            Line Input #1, Dummy
            X = Split(Dummy, " ")
'   MsgBox X(1)
            For j = 1 To num_l * 3
                All_ten_points(jj, j, k) = X(j - 1)
            Next
    End If
i = 2
jj = jj + 1

Loop
Close #1
```

After loading the data, the next routine involves creating CAD levels in MicroStation which are comparable to layers in AutoCAD. Unlike the AutoCAD layers that are identified only by names, MicroStation levels can be identified by both level numbers and names. Creating different levels allows controlling the display of each level independently. Eight levels are created by the developed VBA code; four levels for line drawing and the other four levels for solid drawing. The different levels are associated with different line color, style and weight, and are assigned one of the following names for identification:

- const_BARS
- const_UP_Base

• const_LOW_Base

• const_Cables

• Solid_BARS

• Solid_UP_Base

• Solid_LOW_Base

• Solid_Cables

An example code for creating the level where all members in a tensegrity structure identified as bars are placed and called "const_BARS" is shown below. First, the total number of levels available in the current design file is counted, and then it is determined if any one of the available levels is identified with the level "const_BARS". If unavailable, the level "const_BARS" is created by the MicroStation macro command "ActiveDesignFile.AddNewLevel", and the color and weight are assigned to it accordingly. Other levels are created following the similar procedure.

```
Set Level_tmp = Nothing
tot_lvl = ActiveDesignFile.Levels.Count

For cc = 1 To tot_lvl
    If ActiveDesignFile.Levels.Item(cc).Name = "const_BARS" Then
        Set Level_tmp = ActiveDesignFile.Levels.Item(cc)
    CadInputQueue.SendKeyin "LEVEL SET BYLEVEL COLOR 22   ""const_BARS"""""
    CadInputQueue.SendKeyin "LEVEL SET BYLEVEL WEIGHT 2   ""const_BARS"""""
        ShowStatus "(is) Found Existing const_BARS level ..."
        Exit For
    End If
Next

If Level_tmp Is Nothing Then
    Set Level_tmp = ActiveDesignFile.AddNewLevel("const_BARS")
    CadInputQueue.SendKeyin "LEVEL SET BYLEVEL COLOR 22   ""const_BARS"""""
    CadInputQueue.SendKeyin "LEVEL SET BYLEVEL WEIGHT 2   ""const_BARS"""""
End If
```

With all levels created, line and solid models based on input data are constructed. Constructing the model starts by changing an active level name and executing a command that places a line or cylinder in advance. The routine listed below illustrates the steps in the procedure that displays solid models of the upper bases (other objects can be created in a similar manner). Essentially the routine retrieves three points necessary to construct each cylinder from the variable "All_ten_points". Thus, completing the solid model of an upper base involves iterating several for-loops to retrieve each data point from a matrix containing the necessary data. The last command "CadInputQueue.SendReset" resets the line drawing.

```
' ##############################
'     Create Solid for Upper Base
' ##############################

CadInputQueue.SendCommand "ACTIVE LEVEL Solid_UP_Base"
CadInputQueue.SendCommand "PLACE CYLINDER ICON "

For k = 1 To num_r
      For j = 1 To num_l
            For i = 1 To 4

                  If i = 4 Then
                        point.X = CDbl(All_ten_points(i, 3 * (j - 1) + 1, k))
                        point.Y = CDbl(All_ten_points(i, 3 * (j - 1) + 2, k))
                        point.Z = CDbl(All_ten_points(i, 3 * (j - 1) + 3, k))
                        CadInputQueue.SendDataPoint point, 1

                        point.X = CDbl(All_ten_points(i, 3 * (j - 1) + 1, k)) + TextBox19.Value
                        point.Y = CDbl(All_ten_points(i, 3 * (j - 1) + 2, k))
                        point.Z = CDbl(All_ten_points(i, 3 * (j - 1) + 3, k))
                        CadInputQueue.SendDataPoint point, 1

                        point.X = CDbl(All_ten_points(1, 3 * (j - 1) + 1, k))
                        point.Y = CDbl(All_ten_points(1, 3 * (j - 1) + 2, k))
                        point.Z = CDbl(All_ten_points(1, 3 * (j - 1) + 3, k))
                        CadInputQueue.SendDataPoint point, 1
                  Else

                        point.X = CDbl(All_ten_points(i, 3 * (j - 1) + 1, k))
                        point.Y = CDbl(All_ten_points(i, 3 * (j - 1) + 2, k))
                        point.Z = CDbl(All_ten_points(i, 3 * (j - 1) + 3, k))
                        CadInputQueue.SendDataPoint point, 1

                        point.X = CDbl(All_ten_points(i, 3 * (j - 1) + 1, k)) + TextBox19.Value
```

```
                    point.Y = CDbl(All_ten_points(i, 3 * (j - 1) + 2, k))
                    point.Z = CDbl(All_ten_points(i, 3 * (j - 1) + 3, k))
                    CadInputQueue.SendDataPoint point, 1

                    point.X = CDbl(All_ten_points(i + 1, 3 * (j - 1) + 1, k))
                    point.Y = CDbl(All_ten_points(i + 1, 3 * (j - 1) + 2, k))
                    point.Z = CDbl(All_ten_points(i + 1, 3 * (j - 1) + 3, k))
                    CadInputQueue.SendDataPoint point, 1
                End If
            Next
'           Send a reset to the current command
            CadInputQueue.SendReset
        Next
Next
```

6.3 IMPLEMENTATION OF MVBA

To be able to use the "*Tensegrity_DGN_V1.mvba*", the code needs to be saved in the MicroStation Workspace directory under \system\vba\. The MVBA application can then be loaded by clicking on the "Load Project" button in the VBA Project Manager window (opened by selecting Utilities>Macro>Project Manager from the MicroStation main menu), as shown in Figures 6-3 and 6-4. The entire source code of the application can be viewed by clicking on the Visual Basic Editor button as indicated in Figure 6-3.

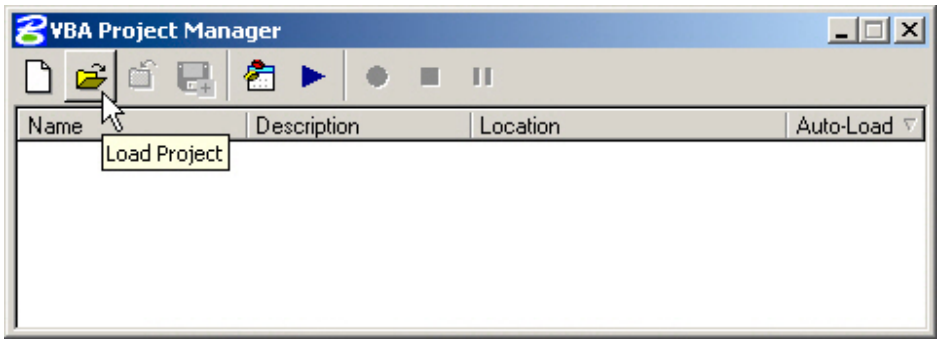


Figure 6-3: VBA project manager window in MicroStation

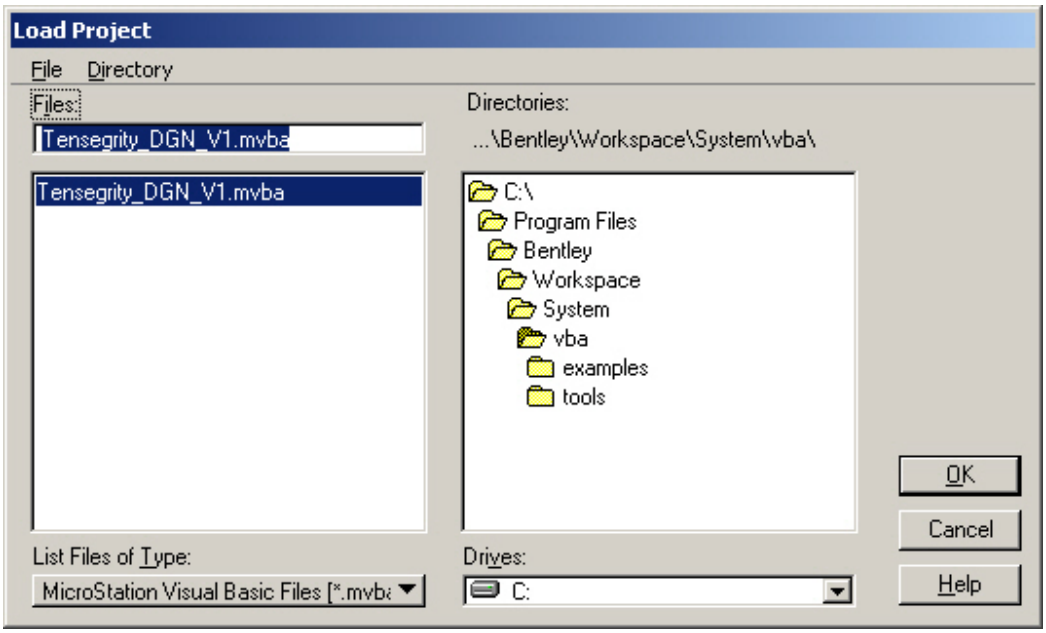


Figure 6-4: Load project window in MicroStation

Once the project "*Tensegrity_DGN_V1.mvba*" is loaded, clicking on the Macro button  in the VBA Project Manager dialog box opens the Macros dialog box shown in Figure 6-5, in which the user can select the macro "*Tensegrity_VBA1*". Clicking on the Run button opens the VBA application "Tensegrity VBA in MicroStation" as shown in Figure 6-6.

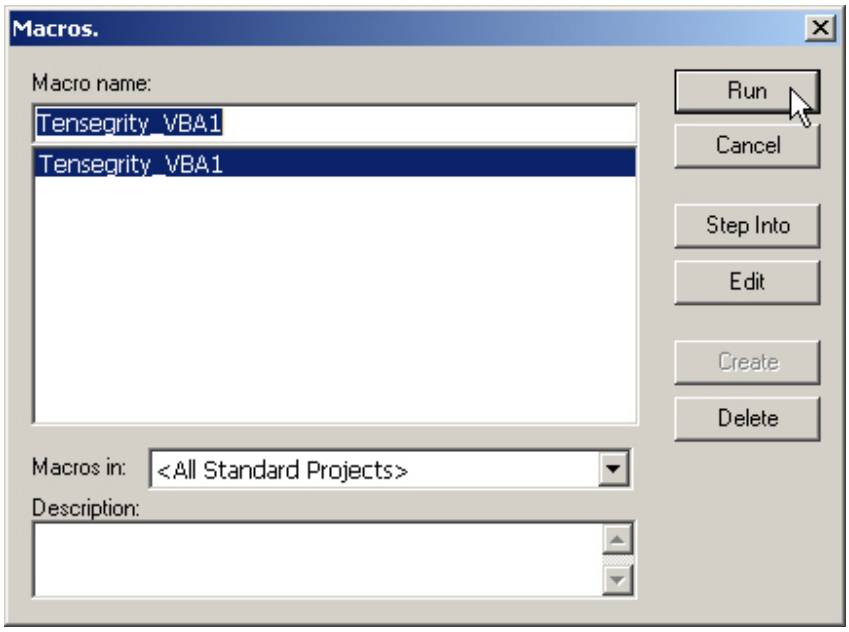


Figure 6-5: Dialog window of macro selection

The VBA application titled "Tensegrity VBA in MicroStation" takes various options from the user and creates a solid model of a tensegrity structure in the MicroStation environment using the 3D coordinates generated from the design algorithms in the MATLAB environment. This application is organized in two columns. The left column displays the numerical values of output, such as unit and structure dimensions, and overlap values of the upper and lower bases. The right side hosts three command buttons "Execute", "Render" and "End", and menus by which the user can specify the size of cable and bar, and several options to control display of resulting 3D solid models.

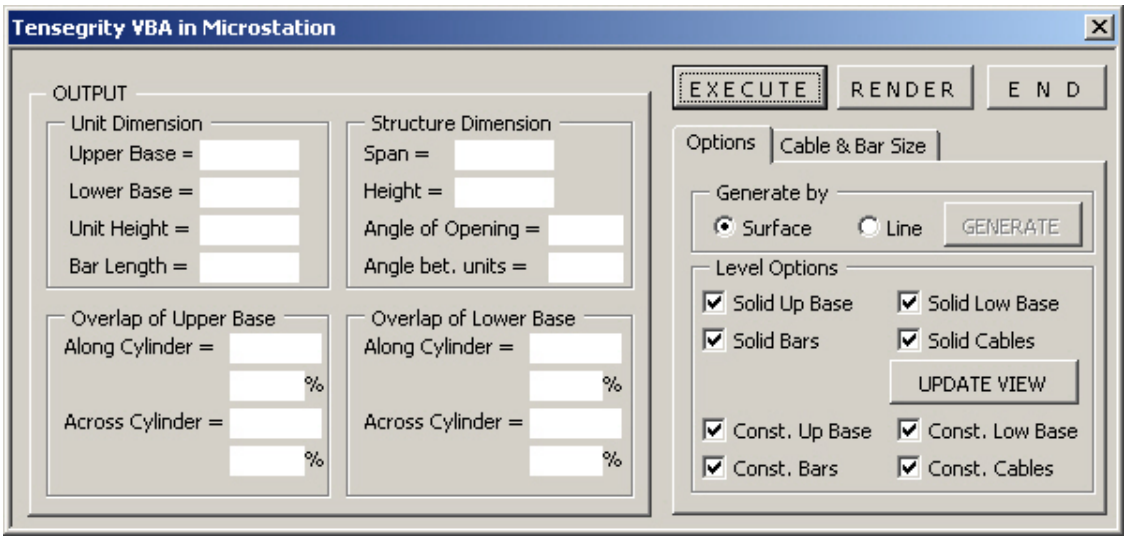


Figure 6-6: Dialog window of tensegrity VBA in MicroStation

While the VBA application remains open, an input file can be loaded by clicking on the "Execute" button which opens the dialog box shown in Figure 6-2, prompting the user to select an input file for visualization. When an input file is selected, the VBA code is automatically executed, generating a solid model and displaying numerical values for the resulting output model on the left side of the user form, as shown in Figure 6-1. The display of the solid model at different levels can be controlled by checking or unchecking appropriate boxes from the option menu. The user can also adjust cable sizes as shown in Figure 6-7. The "Render" button is to create a rendered 3D image of a solid in view 2. Two examples are shown in Figures 6-8 and 6-9 for 2 x 2 and 6 x 4 vaulted tensegrity structures, respectively.

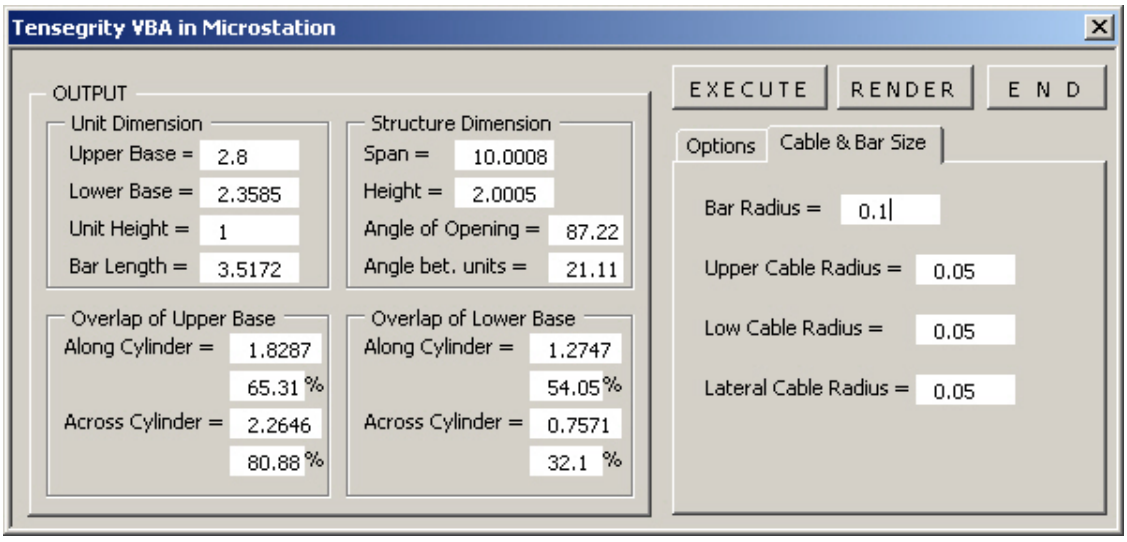


Figure 6-7: Option menu for changing cable and bar sizes

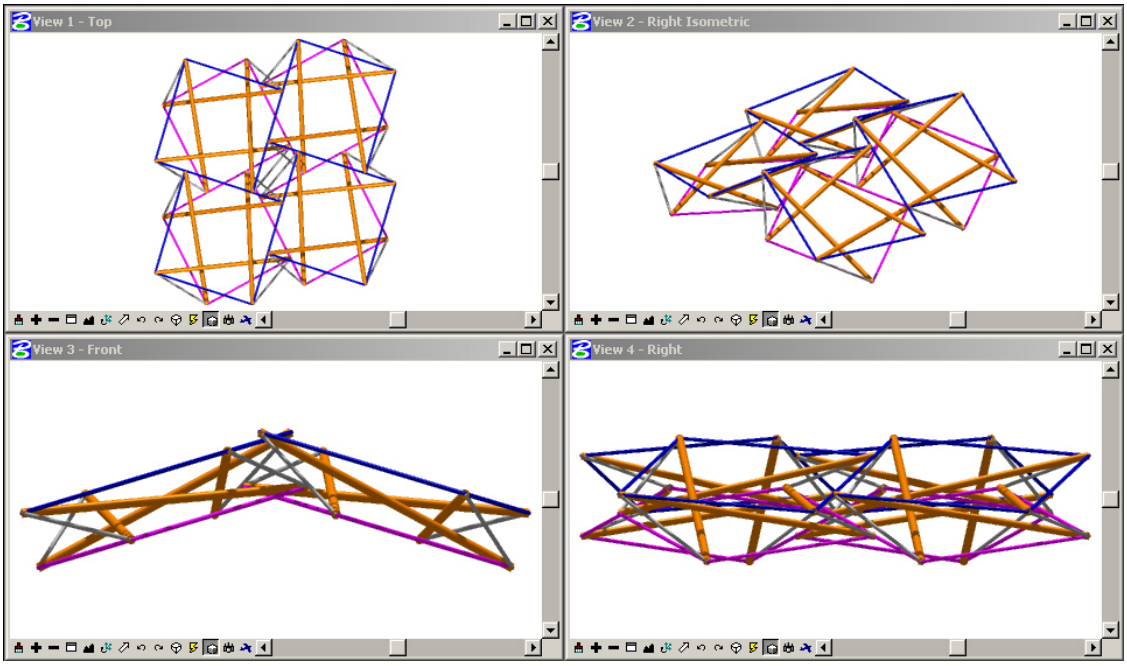


Figure 6-8: 2 x 2 vaulted tensegrity structure generated in MicroStation

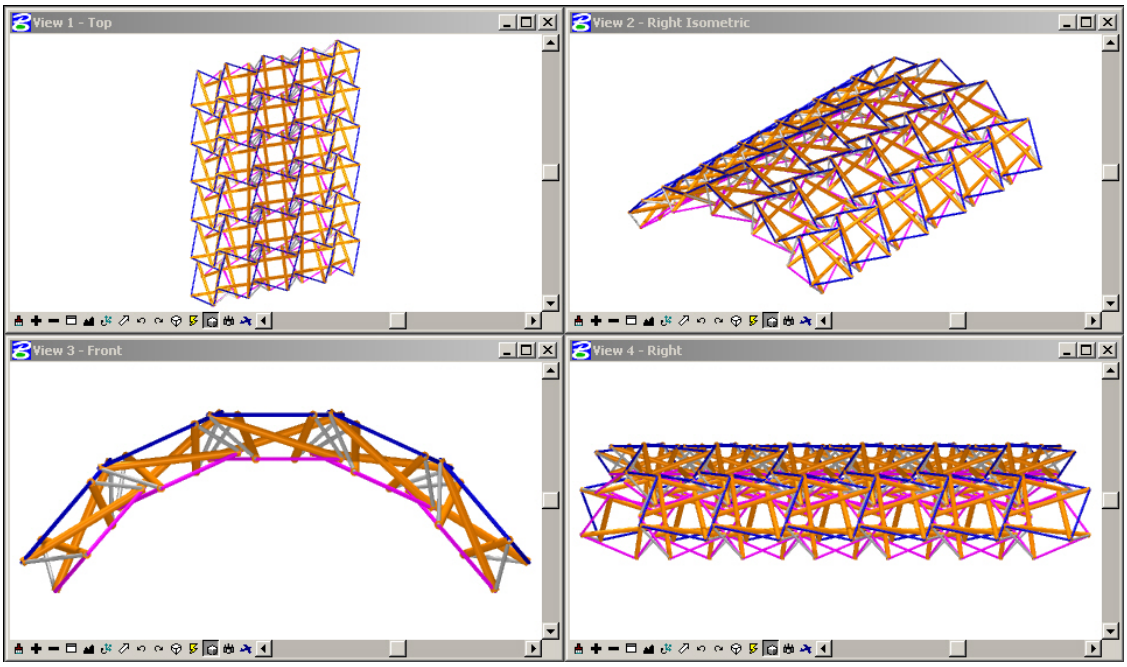


Figure 6-9: 5 x 6 vaulted tensegrity structure generated in MicroStation

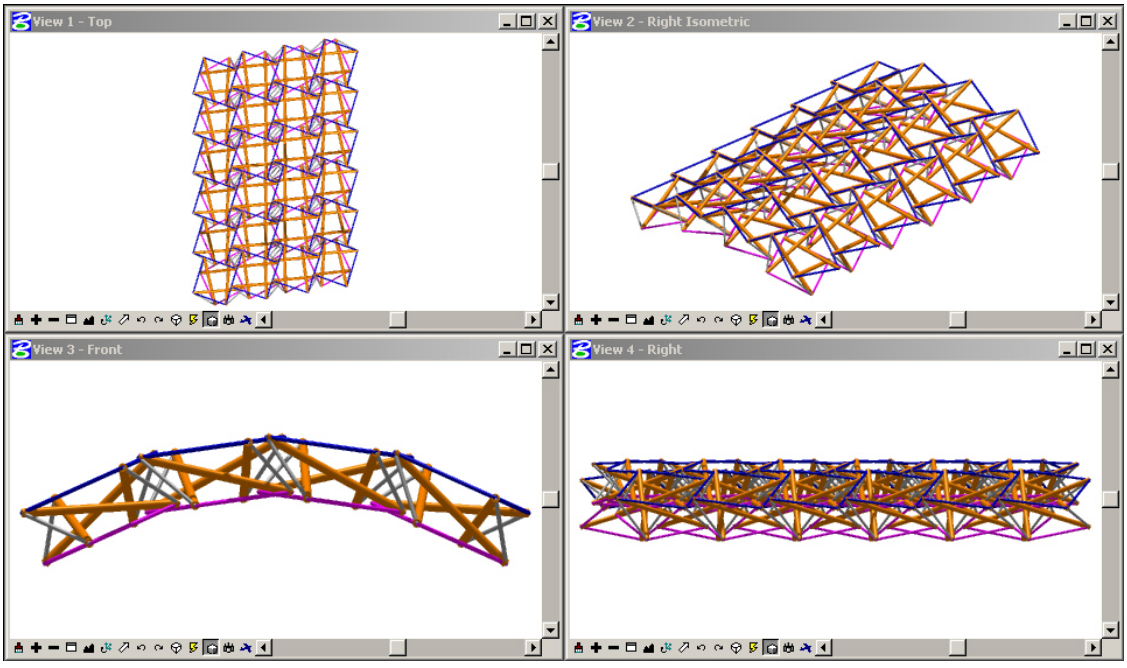


Figure 6-10: 4 x 6 vaulted tensegrity structure generated in MicroStation

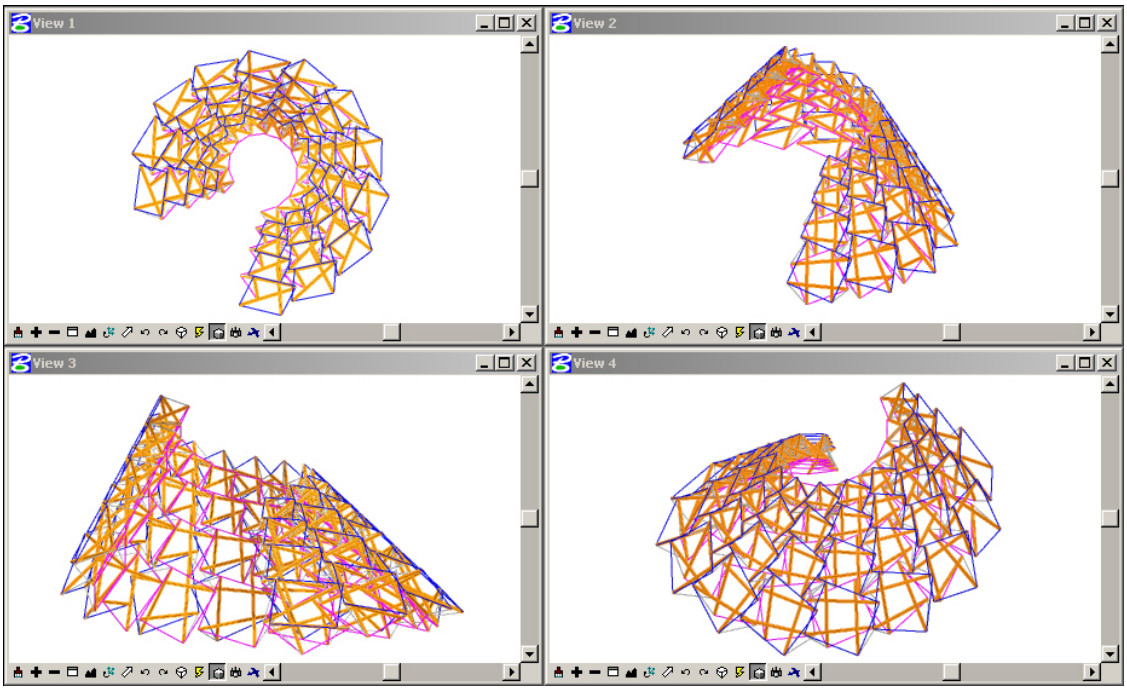


Figure 6-11: 4 x12 helicoid tensegrity structure generated in MicroStation

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

7.1 REVIEW OF RESEARCH OBJECTIVES

The primary objective of this research was to develop a design methodology for single-curvature tensegrity structures.

- Algorithms that generate parametric models of the initial geometry of single-curvature configurations of tensegrity structures have been developed. Two single-curvature configurations are addressed: a uniformly curved configuration (vaulted) and a non-uniformly curved (helicoid).
- Methods and codes for importing models of initial geometry into structural analysis have been developed.
- An integrative visualization environment that displays models of initial geometry and pre-stressed or loaded configurations has been developed. The stress flows in the structure can be also displayed.
- Methods for importing models of tensegrity structures in a commercially available Computer Aided Design environment have been developed.

The developed design algorithms generate parametric models of tensegrity structures, allow for exploring alternate configurations and provide flexibility in the design approach. Specifically two algorithms/methods for vaulted tensegrity structures

were developed that respond to different design scenarios. These are a) unit-based design algorithm and b) structure-based design algorithm.

The first design method is based on the assumption that tensegrity modules can be pre-fabricated and deployed on site before assembly. This design method provides a potential solution in the following instances:

- When the designer wishes to explore alternate configurations that can occur from the use of a module of given dimensions.
- When the designer wishes to re-use existing tensegrity modules to create a configuration different than a previous one by adjusting the amount of overlap between adjacent bases.

The second design method in which the structures' dimensions are given can provide solutions in the following instances:

- When an initial estimate on the desired proportions of the modules is obtained by the application of method 2, or when the exact dimensions of the modules are given (in the case of re-usable modules) and the designer wants to determine how to assemble units of given dimensions to generate a structure also of given dimensions. This seems to be the most common application of the modular assembly method.

The developed algorithm for helicoid structures provides a valuable design tool that helps architects and engineers explore the features of this geometry and consider it for applications in building design.

The developed integrative visualization displays models of (a) initial geometry and (b) geometry after pre-stress and service loads are applied. This tool can help the users identify design parameters critical to the pre-stressed or loaded geometry of tensegrity structures, and allows them to make more informed decision regarding the geometric configuration and performance of tensegrity structures.

The developed VBA application provides the interface with CAD tools to help architects and engineers to rapidly visualize the final geometry of a tensegrity structure in a CAD environment.

7.2 RECOMMENDATIONS FOR FUTURE WORK

Design and visualization methods for vaulted and helicoid tensegrity structure have been successfully developed in this research. Through the course of this research effort, the following topics are proposed for further research:

- Adaptation of the proposed algorithms to tensegrity structures with different topology and morphology, such as double-curvature structures composed of units of different polygonal base geometry.
- Verification of numerical results (obtained from the developed algorithms) through experiments and tests.
- Development of methods to study the dynamic response of tensegrity structures to wind loads or combined multiple live loads.

APPENDIX A: NONSA0 Algorithm provided by Tassoulas (2003)

NOTATION

Consider an arbitrary node (joint) of the tensegrity labeled i. Let

$$\hat{x}_i, \hat{y}_i, \hat{z}_i \tag{1}$$

be the coordinates of the node (with respect to a rectangular Cartesian coordinate system) in the initial estimate of the prestress configuration selected by the analyst. This information is input to NONSA0. Subsequent estimates of the configuration will be identified by the step and iteration in which they are computed. Thus,

$$x_i^{step,iter}, y_i^{step,iter}, z_i^{step,iter} \tag{2}$$

will denote the coordinates of node i at completion of iteration $iter$ in step $step$ of the analysis. Furthermore, a single superscript will be used to indicate the estimate reached after convergence in a step, i.e.,

$$x_i^{step}, y_i^{step}, z_i^{step} \tag{3}$$

will be the coordinates of node i after convergence of the iterations in step $step$ and, for consistency,

$$x_i^0 = \hat{x}_i, y_i^0 = \hat{y}_i, z_i^0 = \hat{z}_i \tag{4}$$

Finally,

$$x_i^{step,0}, y_i^{step,0}, z_i^{step,0} \tag{5}$$

will be understood as the coordinates at initiation of step $step$, prior to any iteration in the step. Note that

$$x_i^{step,0} = x_i^{step-1}, y_i^{step,0} = y_i^{step-1}, z_i^{step,0} = z_i^{step-1} \tag{6}$$

since the coordinates at initiation of step $step$ are those from the conclusion of the immediately preceding step $step-1$.

Next, consider an arbitrary member (strut or cable) of the tensegrity labeled i. The nodes to which the member is connected will be denoted by i_1 and i_2. Let

$$L_{.i} \tag{7}$$

be the length of the member in its unstressed state. This information is input to NONSA0. The length of the member in estimated configurations of the tensegrity will be identified by means of notation patterned after the one adopted for the nodes as described above. Thus,

$$\hat{L}_i = \sqrt{\left(\hat{x}_{i2} - \hat{x}_{i1}\right)^2 + \left(\hat{y}_{i2} - \hat{y}_{i1}\right)^2 + \left(\hat{z}_{i2} - \hat{z}_{i1}\right)^2} \tag{8}$$

will be the length of the member in the initial estimate of the configuration provided by the analyst. Similarly,

$$L_i^{step,iter} \tag{9}$$

will be the length of the member at completion of iteration *iter* in step *step*. The member is assumed to be prismatic with cross-section area

$$A_i \tag{10}$$

and linearly elastic with Young's modulus

$$E_i \tag{11}$$

Finally, the weight per unit volume of member i is specified in terms of its components (input to NONSA0) with respect to the coordinate directions:

$$b_{xi}, b_{yi}, b_{zi} \tag{12}$$

the negative z direction is often selected as coincident with the direction of gravity. Then,

$$b_{xi} = 0, b_{yi} = 0, b_{zi} = -\gamma_i \tag{13}$$

where γ_i is the unit weight of the material of member i.

The procedure implemented in NONSA0 is controlled by two *load factors*, one, λ_1, associated with prestress and self-weight and another, λ_2, that specifies the level of nodal forces applied on the tensegrity:

$$\mathbf{P} = \lambda_2 \cdot \hat{\mathbf{P}} \tag{14}$$

where $\hat{\mathbf{P}}$ is the combination of nodal forces ("loading direction") applied on the structure. In any given step *step*, the values of the load factors

$$\lambda_1^{step}, \lambda_2^{step} \tag{15}$$

or their increments are specified by the analyst (input to NONSA0).

PROCEDURE

For each step *step*, *step* =1, …, *nsteps* (*nsteps* is the number of steps in the analysis, input to NONSA0), iterations, *iter* = 1, …, *maxiter* (*maxiter* is the maximum number of iterations allowed, input to NONSA0) are carried out as follows:

1. For each node i, set

$$x_i^{step,0} = x_i^{step-1}, y_i^{step,0} = y_i^{step-1}, z_i^{step,0} = z_i^{step-1} \tag{16}$$

For iteration *iter* =1, …, *maxiter*, perform the following operations:

2. For each member i, calculate:

the length of the member in the most recent estimate of the configuration:

$$L_i^{step,iter-1} = \sqrt{\left(\hat{x}_{i2} - \hat{x}_{i1}\right)^2 + \left(\hat{y}_{i2} - \hat{y}_{i1}\right)^2 + \left(\hat{z}_{i2} - \hat{z}_{i1}\right)^2} \tag{17}$$

where:

$$x_{i1} = x_{i1}^{step,iter-1}, y_{i1} = y_{i1}^{step,iter-1}, z_{i1} = z_{i1}^{step,iter-1} \tag{18}$$

$$x_{i2} = x_{i2}^{step,iter-1}, y_{i2} = y_{i2}^{step,iter-1}, z_{i2} = z_{i2}^{step,iter-1} \tag{19}$$

• the components of the unit vector directed along member i from node i_1 to node i_2 in the most recent estimate of the configuration:

$$\begin{aligned} n_{xi}^{step,iter-1} &= \frac{x_{i2}^{step,iter-1} - x_{i1}^{step,iter-1}}{L_i^{step,iter-1}} \\ n_{yi}^{step,iter-1} &= \frac{y_{i2}^{step,iter-1} - y_{i1}^{step,iter-1}}{L_i^{step,iter-1}} \\ n_{zi}^{step,iter-1} &= \frac{z_{i2}^{step,iter-1} - z_{i1}^{step,iter-1}}{L_i^{step,iter-1}} \end{aligned} \tag{20}$$

• the length of the member in its unstressed state, as controlled by the load factor λ_1:

$$L_i^{step} = \hat{L}_i + \lambda_1^{step} \cdot \left(L_i - \hat{L}_i\right) \tag{21}$$

Note that for $\lambda_1^{step} = 1$, L_i^{step} becomes equal to the actual length, L_i, of the member in its unstressed state.

• the force in the member

$$F_i^{step,iter-1} = \frac{E_i \cdot A_i}{L_i^{step}} \cdot \left(L_i^{step,iter-1} - L_i^{step} \right) \tag{22}$$

• the member "right-hand side" for this Newton equilibrium iteration:

$$\begin{bmatrix} 1/2 \cdot \lambda_1^{step} \cdot b_{xi} \cdot A_i \cdot L_i^{step} + F_i^{step,iter-1} \cdot n_{xi}^{step,iter-1} \\ 1/2 \cdot \lambda_1^{step} \cdot b_{yi} \cdot A_i \cdot L_i^{step} + F_i^{step,iter-1} \cdot n_{yi}^{step,iter-1} \\ 1/2 \cdot \lambda_1^{step} \cdot b_{zi} \cdot A_i \cdot L_i^{step} + F_i^{step,iter-1} \cdot n_{zi}^{step,iter-1} \\ 1/2 \cdot \lambda_1^{step} \cdot b_{xi} \cdot A_i \cdot L_i^{step} - F_i^{step,iter-1} \cdot n_{xi}^{step,iter-1} \\ 1/2 \cdot \lambda_1^{step} \cdot b_{yi} \cdot A_i \cdot L_i^{step} - F_i^{step,iter-1} \cdot n_{yi}^{step,iter-1} \\ 1/2 \cdot \lambda_1^{step} \cdot b_{zi} \cdot A_i \cdot L_i^{step} - F_i^{step,iter-1} \cdot n_{zi}^{step,iter-1} \end{bmatrix} \tag{23}$$

• the member "tangent stiffness matrix" for this Newton equilibrium iteration:

$$\begin{bmatrix} \boldsymbol{k_i} & -\boldsymbol{k_i} \\ -\boldsymbol{k_i} & \boldsymbol{k_i} \end{bmatrix} \tag{24}$$

where:

$$\boldsymbol{k_i} = \begin{bmatrix} k_i \cdot n_{xi} \cdot n_{xi} + g_i & k_i \cdot n_{xi} \cdot n_{yi} & k_i \cdot n_{xi} \cdot n_{zi} \\ k_i \cdot n_{yi} \cdot n_{xi} & k_i \cdot n_{yi} \cdot n_{yi} + g_i & k_i \cdot n_{yi} \cdot n_{zi} \\ k_i \cdot n_{zi} \cdot n_{xi} & k_i \cdot n_{zi} \cdot n_{yi} & k_i \cdot n_{zi} \cdot n_{zi} + g_i \end{bmatrix} \tag{25}$$

$$k_i = \frac{E_i A_i}{L_i^{step,iter-1}} \tag{26}$$

$$g_i = \frac{F_i^{step,iter-1}}{L_i^{step,iter-1}} \tag{27}$$

and

$$\begin{aligned} n_{xi} &= n_{xi}^{step,iter-1} \\ n_{yi} &= n_{yi}^{step,iter-1} \\ n_{zi} &= n_{zi}^{step,iter-1} \end{aligned} \tag{28}$$

• assemble the member right-hand side and tangent stiffness matrix into the corresponding vector and matrix of the tensegrity, **RHS** and **K**.

3. Solve the system of equations:

$$\mathbf{K}\Delta\mathbf{U} = \lambda_2^{step}\,\hat{\mathbf{P}} + \mathbf{RHS} \tag{29}$$

4. Obtain the revised estimate of the configuration:

$$\begin{aligned} x_i^{step,iter} &= x_i^{step,iter-1} + \Delta u_i \\ y_i^{step,iter} &= y_i^{step,iter-1} + \Delta v_i \\ z_i^{step,iter} &= z_i^{step,iter-1} + \Delta w_i \end{aligned} \tag{30}$$

where: Δu_i , Δv_i and Δw_i are the increments of displacements of node i (contained in $\Delta\mathbf{U}$) in the *x*, *y* and *z* directions.

5. Check to determine whether convergence has been achieved. A number of alternatives are available in this direction. The criterion implemented in NONSA0 is:

$$(\mathbf{RHS}\cdot\Delta\mathbf{U})_{step,iter} < TOL \cdot (\mathbf{RHS}\cdot\Delta\mathbf{U})_{step,1} \tag{31}$$

where: *TOL* is the tolerance specified by the analyst (input to NONSA0). If the criterion is satisfied (convergence has been achieved), set

$$\begin{aligned} x_i^{step} &= x_i^{step,iter} \\ y_i^{step} &= y_i^{step,iter} \\ z_i^{step} &= z_i^{step,iter} \end{aligned} \tag{32}$$

and proceed to the next step, beginning with item 1 above. Otherwise, proceed to the next iteration, beginning with item 2 above.

BIBLIOGRAPHY

Adriaenssens, S. M. L., Barnes, M. R. (2001). “Tensegrity Spline Beam and Grid Shell Structures.” Engineering Structures, 23, 29-36.

Belkacem, S., (1987). “Recherche de forme par relaxation dynamique de systemes reticules spatiaux autocontraints.” These de Docteur Ingenieur, Universite Paul Sabatier, Toulouse.

Bin-Bing, W., Yan-Yun, L. (2001). “From Tensegrity Grids to Cable-strut Grids.” International Journal of Space Structures, 16(4), 279-314.

Charalambides, J. E., (2004). “Computer Method for the Generation of the Geometry of Tensegrity Structures”, Dissertation, Department of Civil Engineering, The University of Texas at Austin.

Chassagnoux, A., Chomarat, J. S. (1992). “A Study of Morphological Characteristics of Tensegrity Structures.” International Journal of Space Structures, 7(2), 165-172.

Connelly, R., Rigidity(1993), “Handbook of Convex Geometry.” Elsevier Publishers Ltd, in: P.M. Gruber and J.M. Wills, ed, 223-271.

Emmerich, D.G. (1988). “Structures Tendues et Autotendantes.”Ecole d’ Architecture de Paris- La Villette, Paris.

Emmerich, D.G. (1996). “Emmerich on Self-Tensioning Structures.” International Journal of Space Structures, 11(1&2), 29-36.

Fuller, R. B., (1962). “Tensile-integrity Structures.” US patent 3,063,521, 1962, 13.

Fuller, R. B., (1965). U.S. Patent No 3,169,611

Hanaor, A. (1992). “Aspects of Design of Double-Layer Tensegrity Domes.” International Journal of Space Structures, 7(2), 101-113.

Hanaor, A. (1993). “Double-Layer Tensegrity Grids as Deployable Structures.” International Journal of Space Structures, 8(1&2), 135-143.

Hanaor, A. (1998). “Tensegrity Theory and Application.” Beyond the Cube, edited by J. Francois Gabriel, John Wiley & Sons, Inc., 385-408.

Kazi, F. S., (2001). “Initial Geometric Visualization of Double Layer Tensegrity Structures”, Master’s Thesis, Department of Civil Engineering, The University of Texas at Austin.

Kenner, H.(1976). “Geodesic Math and How to Use It.”, University of California Press, Berkeley, California.

Liapi, K. A. (2000). "Tensegrity Structures and Structural and Architectural Conceptioning." Structural Morphology Colloquium, International Association for Shell and Spatial Structures (IASS), Proceedings, Delft, Netherlands, 235-240.

Liapi, K.A. (2001a), "Geometric Configuration and Graphical Representation of Tensegrity Spherical Networks." Proceedings, Association for Computer Aided Design in Architecture (ACADIA) 2001, Buffalo, NY, 258-267.

Liapi, K.A. (2001b). “A Visualization Method for the Morphological Exploration of Tensegrity Structures.” Proceedings, IEEE Computer Society Fifth International Conference on Information Visualization (IV 2001), July 25-26, 2001, London, 521-528

Liapi, K. A. (2002a). “A Novel Portable and Collapsible Tensegrity Unit for the Rapid Assembly of Tensegrity Networks." Fifth International Conference on Space Structures, University of Surrey, England, 39-46.

Liapi, K. A. (2002b). “Tensegrity Module, Structure and Method for Construction” Patent Application # 5119-07202, 5/28/2002, 5/28/2001(provisional), filed by “Conley, Rose, Tayon Intellectual Property Law,” sponsored by U.T. Office of Technology Licensing and Intellectual Property.

Liapi, K. A., and Kim, J. (2003) “A Parametric Approach to the Design of a Tensegrity Vaulted Dome for an Ephemeral Structure for the 2004 Olympics”, Association for Computer Aided Design in Architecture (ACADIA 22)

Liu, S. (2004). “Parametric Structural Analysis of Vaulted Double-Layer Tensegrity Structures” Master’s Thesis, Department of Civil Engineering, University of Texas at Austin.

Motro, R. (1987). “Tensegrity systems for Double-layer Space Structures.” Proceedings of the International Conference on the Design and Construction of Non-conventional Structures, 2, 43-51

Motro, R. (1990). “Tensegrity systems and geodesic domes.” International Journal of Space Structures, 5(3-4), 341-351.

Motro, R. (1992). “Tensegrity Systems: The State of the Art.” International Journal of Space Structures, Special Issue on Tensegrity Systems, 7(2), 75-84.

Motro, R., Bouderbala M. (1996). “Mobile Tensegrity Systems.” MARAS 96, Mobile and Rapidly Assembled Structures II, Proceedings, 103-113.

Motro, R. (2003). “Tensegrity: Structural Systems for the Future.” Kogan Page Science Press.

Nooshin H , Disney P., (1991). “Elements of Formian.” Computers and Structures, 41(6), 1183-1215.

Pellegrino, S., Tibert, A.G. (2003). “Review of form-finding methods for tensegrity structures.” International Journal of Space Structures, 18(4), 209-223.

Pugh, A., (1976). “An introduction to tensegrity” Berkeley: University of California Press.

Selfridge, R. G. (2001). “About Some Tensegrity Structures.” International Journal of Space Structures, 16(4), 231-235.

Snelson, K. D. (1965). “Continuous Tension, Discontinuous Compression Structures”, United States Patent 3, 169,611, February 16, 1965.

Stern, I. P., (1999). “Development of Design Equations for Self-Deployable N-Strut Tensegrity Systems”, Master’s Thesis, Department of Mechanical Engineering, University of Florida.

Sultan, C., (1999). "Modeling, Design, and Control of Tensegrity Structures with Applications.", Ph.D., Purdue University, West Lafayette.

Sultan, C., Corless, M., and Skelton, R. E. (2002). “Symmetrical Reconfiguration of Tensegrity Structures.” International Journal of Solids and Structures, 39, 2215-2234.

Tassoulas, J. L., (2002), “Personal communication”, Department of Civil Engineering, The University of Texas at Austin.

Tassoulas, J. L., (2003), “Implementation of Tensegrity Analysis in NONSA0”, Unpublished Notes, Department of Civil Engineering, The University of Texas at Austin.

Vassart, N., and Motro, R. (1999). “Multiparametered formfinding method: application to tensegrity systems.” International Journal of Space Structures, 14 (2), 147-154.

Vilnay, O., (1977). “Structures made of Infinite Regular Tensegric Nets.” IASS Bulletin, 18(63), 51-57.

Vilnay, O., (1981). “Determinate Tensegric Shells.” Journal of Structural Division, ASCE, 2029-2033.

Wang B. B., and Yan Yun Li, (1998). “Definition of tensegrity systems. Can dispute be settled?”, “Lightweight structures in architecture engineering and construction”, LSA98 Proceedings, 2, 713-719.

VITA

Jinman Kim was born in Changwon, Korea on November 23, 1967, the son of Byung Kyu Kim and Won Yi Kankg. Between 1986 and 1993 he attended the Ajou University, Suwon, Korea in which he received the degree of Bachelor of Science. He entered The University of Texas at Arlington in August 1995 and he received the degree of Master of Science in May 1998. In August 1998 he started his Ph.D. program at The University of Texas at Austin.

Permanent Address:
Kyung-Nam Changwon City Sang Nam Dong Dae-Ho Tower 503
South Korea 641-080

This dissertation was typed by Jinman Kim.