GEORGIA INSTITUTE OF TECHNOLOGY

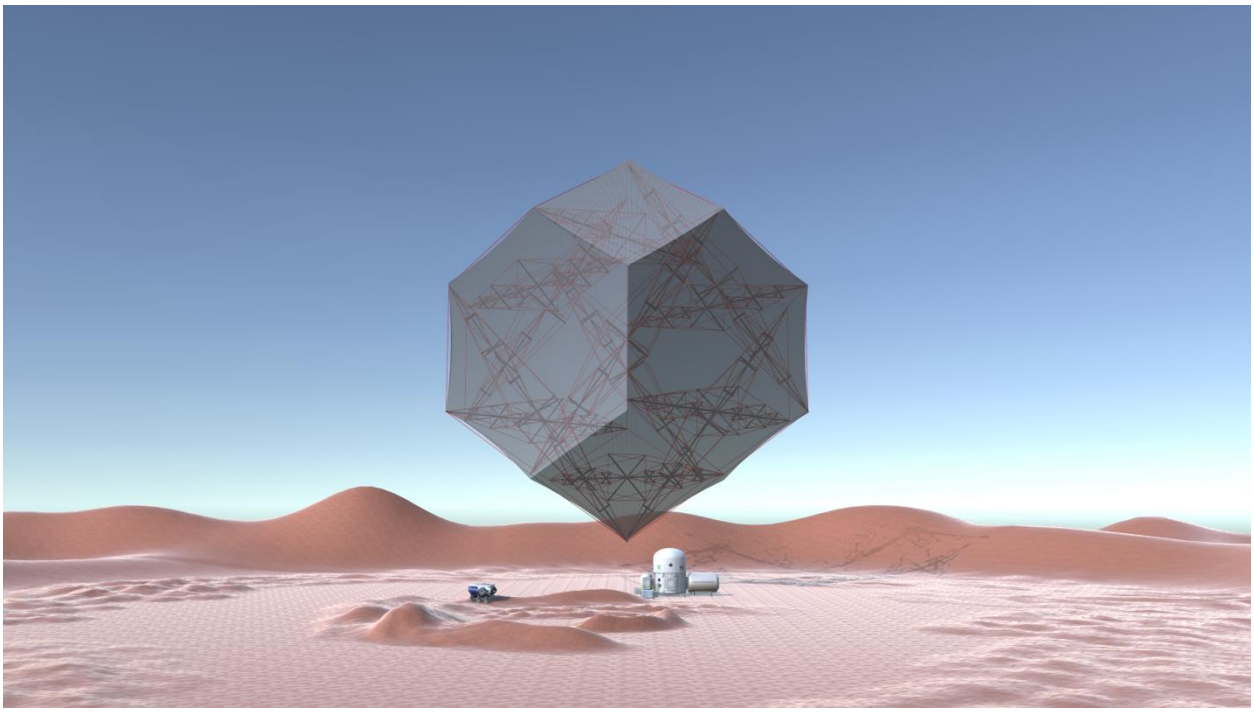
NASA Innovative Advanced Concepts

Evacuated Airship for Mars Missions

NIAC Phase I, 2017

Dr. John-Paul Clarke, Dr. Julian Rimoli, James Todd Gloyd, Hernan Logarzo, Julie Kraus

2/14/2018



Georgia Tech Air Transportation Laboratory

Table of Contents

Executive Summary

An evacuated or vacuum airship relies on the same principle of buoyancy used by standard balloons. However, unlike a balloon which uses a lighter than air gas to displace air and provide lift, the vacuum airship leverages a rigid structure to maintain a vacuum and displace air, thereby providing buoyancy. This method is similar to how a ship uses a rigid structure to displace water and fill the space with air; an evacuated airship uses the same mechanism, except air is displaced and the space remains vacant. Using this method, the evacuated airship is capable of utilizing the full potential of the displaced mass of air, which has interesting implications in the Martian atmosphere. Unlike other aerial vehicles, which are at a disadvantage in Martian atmospheric conditions, the evacuated airship benefits from the Martian atmosphere by virtue of the temperature and molecular composition. As a result, the evacuated airship offers an unprecedented payload capacity and, if implemented, may be used to transport current and future scientific instruments, other vehicles, rovers, and possibly even human habitations. A standard dirigible or balloon for Mars would have a severely limited span of operation and a very narrow field of study, nearly exclusively the atmosphere, but a vacuum airship can be used as a long term tool for many different missions: transportation, ground study, communications, atmospheric study, etcetera, thereby making it a far more economically sensible choice.

This investigation illustrates development of several different approaches to the evacuated airship which are dictated by different enabling technologies as well as those viable with current technology. For current materials technology, this investigation has addressed and solved the most core feasibility aspects of the concept, laying the foundation for further development of the mission and design. The current design of the evacuated airship uses a tensegrity structure, which is a truss structure comprised of bars in pure compression and cables in pure tension, to support an outer membrane. Beams of the tensegrity structure themselves are comprised of more intricate tensegrity structures, which reduce the overall mass of the design, enabling payload capacity. As such, this design is fully capable of supporting the load from atmospheric pressure on Mars while remaining light enough to have useful payload capacity, which was testing using detailed, non-linear finite element simulations, accounting for non-linearities in displacement, global and local buckling, and membrane failure criteria. This was further improved by combining and extending several design methods to reduce the overall mass of the structure. As can be shown, the current payload of the design is 500 kilograms, with projections through further implementation of the developed design methods to have a payload over one ton, and even more payload can be expected from further development of the design and mission. There still remain many other avenues for further mass reduction of the structure and optimization of the design in general, which can be used in conjunction with the methods developed over this investigation.

Additionally, protocols for the fundamentals of a mission implementing the evacuated airship on Mars are examined in this investigation. These protocols cover the transport, deployment, and planetary insertion of the evacuated airship on Mars, which are the main criteria for mission feasibility. Planetary insertion was analyzed using high fidelity numerical methods to observe the full scope of influences on the evacuated airship during entry into the atmosphere. As a result, the underlying analysis behind the installation of the evacuated airship on Mars has been covered sufficiently and provides a general framework which is fully capable of conforming to future changes and adaptations to the design.

Overall, the evacuated airship represents an exciting and revolutionary concept for Mars and will enable missions which would otherwise be impossible. Not to mention, a vast majority of the structural methods and theory developed for the evacuated airship can be applied to many other projects which solicit high load bearing capability, low mass, and/or deployability—terrestrial, Martian, or otherwise. The results of this investigation are foundational and should not be considered final. There are still many aspects of the evacuated airship concept which need to be further developed, however, this research covers the main points of feasibility for the concept.

Planetary Analysis

Earth's atmosphere is fairly adversarial towards the concept of an evacuated airship. However, the Martian atmosphere can be shown to be very well situated, by virtue of Mars' temperature and atmospheric composition, to be a prime candidate for implementation of the evacuated airship. Additionally, the evacuated airship represents an unparalleled advance in mission capabilities on Mars once implemented. The following analysis generalizes the advantageous nature of the Martian atmosphere as applied to the evacuated airship concept.

Using the statics of the vacuum airship and drastically simplified models, a preliminary mathematical model of the system may be created. Based on Archimedes' principle: in equilibrium, the mass of a system is equivalent to the mass of the volume of fluid displaced; therefore the lift of the vacuum airship is a function of the volume displaced, demonstrated in Figure 1.1 and Equation 1.1. By taking a differential section of the design shown in Figure 1.1, and applying statics (as shown in Figure 1.2), the stress in the shell can be found, shown in Equation 1.2. Taking the ratio of stress to lift by combining and simplifying Equations 1.1 and 1.2, the relation shown in Equation 1.3 can be derived to reveal the nature of the basic vacuum airship properties.

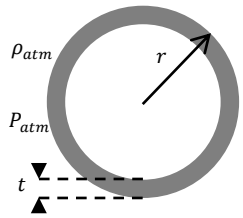


Figure 1.1

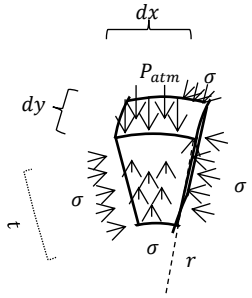


Figure 1.2

$$L = \rho_{atm} V = \rho_{atm} \frac{4}{3} \pi r^3 \tag{1.1}$$

$$\sigma = \frac{1}{2} P_{atm} \frac{r}{t} \tag{1.2}$$

$$\frac{\sigma}{L} = \frac{P_{atm}}{\rho_{atm}} \frac{3}{8\pi r^2 t} \tag{1.3}$$

Additionally, for a non-monocoque design utilizing beams of circular cross section as illustrated in Figure 1.3, the lift is again proportional to the volume displaced as shown in Equation 1.4. Stresses in the supporting beams can be found from statics, taking the summation of pressure over each face and applying an equivalent force, based on angle, to the beam connected to that face, resulting in Equation 1.5. As before, by taking the ratio of stress to lift by combining Equations 1.4 and 1.5, the relation represented in Equation 1.6 is derived, which shows similar relationships as Equation 1.3. Since the thickness of beam or shell of each design is determined by the material and the size determined by scale, then the atmospheric density and pressure become the elements of interest for this preliminary analysis.

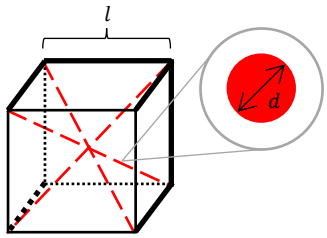


Figure 1.3

$$L = \rho_{atm} V = \rho_{atm} l^3 \tag{1.4}$$

$$\sigma = \frac{\sqrt{6}}{4\pi} P_{atm} \frac{l^2}{d^2} \tag{1.5}$$

$$\frac{\sigma}{L} = \frac{P_{atm}}{\rho_{atm}} \frac{\sqrt{6}}{4\pi d^2 l} \tag{1.6}$$

To this end, we studied the effects of atmospheric density and pressure on the efficiency of the airship. Starting with the ideal gas law, using the definition of density as the quotient of mass by volume, and the definition of molar mass as the ratio of mass per mole, a relationship for the ratio of pressure to density can be found to be proportional to the ratio between temperature and average molecular mass. From this derivation and by the observations made above, one can see a specific atmosphere's suitability for the vacuum airship is a function of the temperature and the molecular mass of the atmosphere, specifically the quotient of the two.

Since benefit may come from the design having a non-spherical or non-cubic form, and since the above models are exceedingly simplified, generalizing the previous relations to an arbitrary body is necessary. Such generalization can be accomplished by first taking an arbitrary body defined in frame e_1, e_2, e_3, and performing a change of basis to frame e_1', e_2', e_3', where $e_i = \alpha\, e_i'$, for $i = 1, 2, 3$, which is equivalent to changing the size of the body. This change of frame is shown in Figure 1.4 as it would apply to scaling an arbitrary body. The transformation is also represented in Equation 1.7 using a transformation matrix which in turn is represented in Equation 1.8.

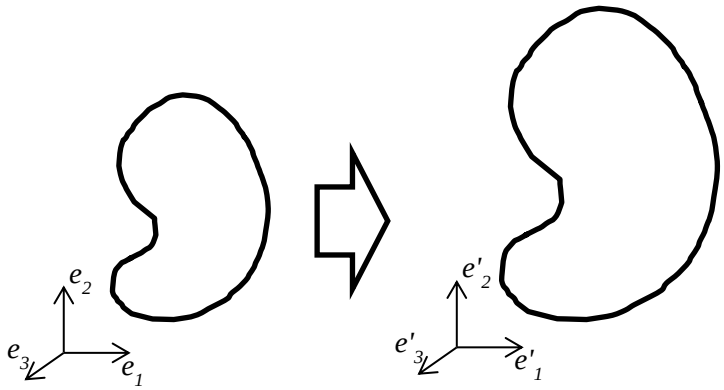


Figure 1.4

$$\underline{e_i'} = l_{ij}\underline{e_j} \tag{1.7}$$

$$l_{ij} = \frac{1}{\alpha}\delta_{ij} \tag{1.8}$$

The relationship between volume before and after the transformation is shown in Equation 1.9 and the same is shown for the surface area in Equation 1.10. Defining a specific transformation into a body with unit volume allows definition of α_0 for specific body geometry. Solving for the value of the aforementioned α_0 yields Equation 1.11 where A_0 is a constant: the surface area of an equivalently shaped body of unit volume. Therefore, for any two arbitrary bodies of equivalent volume, their surface areas are constantly proportional, regardless of scale.

$$V = \iiint \varepsilon_{ijk} dx_i dy_j dz_k = \alpha^3 V' \tag{1.9}$$

$$A = \iint \sqrt{\varepsilon_{ijk} dx_j dy_k \varepsilon_{imn} dx_m dy_n} = \alpha^2 A' \tag{1.10}$$

$$\frac{A}{A_0} = V^{2/3} \tag{1.11}$$

From this conclusion, and seeing from above the stress scales with surface area and the lift scales with volume, then the suitability of an atmosphere can be generalized to any design of the vacuum airship as proportional to the quotient of the average temperature and the average molecular mass. In essence, this conclusion allows application of the relation in general, even though the original models are far from the final form of the design. Observing each of the planets in the solar system, this generalized relationship shows Mars to have the best atmosphere for implementation of the vacuum airship.

Mars having the most suitable atmosphere for the vacuum airship is quite remarkable, since the Martian atmosphere is a severe detriment to all other flight vehicle designs. The Martian atmosphere is comprised almost entirely of carbon dioxide, so vehicles cannot use typical combustion fuels which require an atmospheric oxidizer. Glider, plane, and helicopter designs are all hindered by the atmosphere of Mars due to the low Reynolds number and relatively low density. Additionally, the vacuum airship provides benefit over other airship designs and superpressure balloons because of the inherent robustness of the design. Moreover, if damage is sustained, the vacuum airship can land, be repaired, and the re-evacuate to resume operation, whereas another airship would need to be refilled with a lighter than air gas. Therefore, the evacuated airship design thrives in an environment where most other aircraft are at a disadvantage.

Design Iterations

The design of the evacuated airship for Mars missions has been a process consisting of many design iterations. At onset, the simplest design possibilities were explored; when impossibilities were encountered, the design was modified by incorporation of new design technologies to mitigate the difficulties encountered with each iteration. Each design iteration expands the applicable knowledge of the problem, and enables more informed design decisions moving forward, culminating in a design which is now solidly founded in both structure and fundamental mission aspects.

Initial analysis and design for the vacuum airship consisted of several monocoque designs which had differing approaches to providing the strength needed while remaining lightweight enough to enable flight. The first design iteration using a homogeneous shell, though shown to be impossible given current materials, provided a reliable analytic foundation for later work. In the following design iterations sandwich structures were used to check if the material constraints could be overcome in such a way. These later iterations were closer to success, but again were found to be in feasible. This in no way should be considered failure however because these designs, though unsuccessful on their own, helped to build analytic groundwork for the current design. Additionally, the sandwich structure based designs are simply hindered by manufacturing constraints imposed by current materials technology. As such, these designs are perfectly valid if and when materials technology advances to their points of manufacture. Furthermore, the sandwich structure designs may be preferable in the case where their material constraints are met since they have slightly higher theoretical payloads than the current design, though the current design is vastly more feasible with current technology.

Homogeneous Shell

Simplest of the designs for an evacuated airship is a homogeneous monocoque shell. The following analysis for a vacuum airship comprised of a homogeneous monocoque shell uses the model shown in Figure 2.1. From this model, the accompanying equation (2.4) governing the buoyancy of the airship can be obtained. In Equation 2.1, total mass for the system is calculated from the product of the shell volume and the material density in addition to an efficiency factor, which represents how far the vacuum is from absolute, and the payload. An explicit representation of the efficiency factor is shown in Equation 2.2. Mass displaced by the system is shown in Equation 2.3 as the product of the total system volume and the atmospheric density. By combining Equations 2.1 and 2.3, Equation 2.4 is obtained.

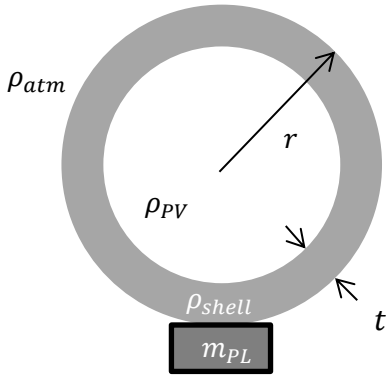


Figure 2.1

$$m_{ship} = \frac{4}{3}\pi(r^3 - (r-t)^3)\rho_{shell} + \eta_v \rho_{atm} + m_{PL} \tag{2.1}$$

$$\eta_v = \frac{4}{3}\pi(r-t)^3 \frac{\rho_{PV}}{\rho_{atm}} \tag{2.2}$$

$$m_{disp} = \frac{4}{3}\pi r^3 \rho_{atm} \tag{2.3}$$

$$\Delta L = m_{disp} - m_{ship} = \frac{4}{3}\pi[r^3\rho_{atm} - (r^3 - (r-t)^3)\rho_{shell}] - \eta_v\rho_{atm} - m_{PL} \tag{2.4}$$

Taking a differential section of the above model, a model for analysis of the stresses on the shell can be created as shown in Figure 2.2 and 2.3. Stresses in the shell can be obtained using static analysis, shown in Equation 2.5, of the forces acting on the differential section. The stress in the shell, shown in Equation 2.6, can then be obtained by rearranging Equation 2.5 and assuming stresses are uniform along the X and Y directions.

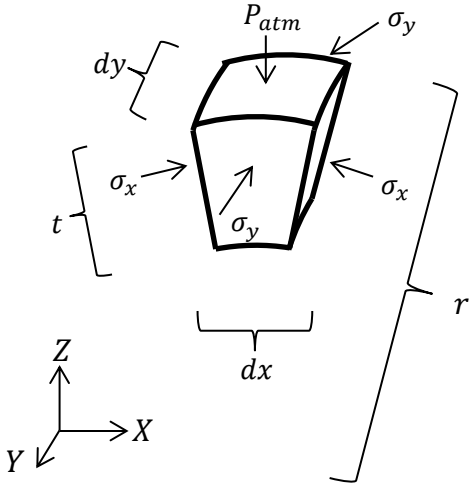


Figure 2.2

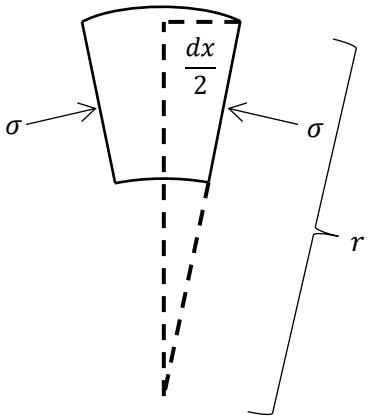


Figure 2.3

$$\sum F_z = 0 = P_{atm}dxdy - \left(2\sigma_x t \cdot dy\frac{dx}{2r} + 2\sigma_y t \cdot dx\frac{dy}{2r}\right) \tag{2.5}$$

$$\sigma = \frac{P_{atm}}{2}\frac{r}{t} \tag{2.6}$$

Using the value found for stress (Equation 2.6) and the buckling equation for a spherical plate under uniform external pressure[1], shown in Equation 2.7, leads to a representation of the critical buckling pressure for a homogeneous monocoque sphere (Equation 2.8). Now, Equation 2.4 can be reduced to Equation 2.9 by assuming neutral buoyancy, i.e. mass displaced equivalent

to the mass of the airship, and assuming a virtually perfect evacuation in addition to no payload. Equation 2.9 can subsequently be reduced to Equation 2.10, providing value for the required ratio of shell thickness to the radius imposed by neutral buoyancy. By assuming minimum structure to support atmospheric pressure, the critical buckling pressure from Equation 2.8 becomes atmospheric pressure. Rearranging Equation 2.8 gives Equation 2.11 representing the ratio of shell thickness to the radius imposed by buckling. Imposing the limitations from both buoyancy and buckling, Inequality 2.12 can be created.

$$\sigma_{cr} = \frac{E}{\sqrt{3(1-\nu^2)}}\frac{t}{r} \tag{2.7}$$

$$P_{cr} = \frac{2E}{\sqrt{3(1-\nu^2)}}\left(\frac{t}{r}\right)^2 \tag{2.8}$$

$$0 = \frac{4}{3}\pi\{r^3\rho_{atm} - [r^3 - (r-t)^3]\rho_{shell}\} \approx 4\pi\left(\frac{r^3}{3}\rho_{atm} - r^2\rho_{shell}t\right) \tag{2.9}$$

$$\frac{t}{r} = 1 + \frac{(\rho_{atm}\rho_{shell}^2 - \rho_{shell}^3)^{1/3}}{2\rho_{shell}} \tag{2.10}$$

$$\frac{t}{r} = \sqrt{\frac{P_{atm}\sqrt{3(1-\nu^2)}}{2E}} \tag{2.11}$$

$$\sqrt{\frac{P_{atm}\sqrt{3(1-\nu^2)}}{2E}} \leq 1 + \frac{(\rho_{atm}\rho_{shell}^2 - \rho_{shell}^3)^{1/3}}{2\rho_{shell}} \tag{2.12}$$

In searching material properties, the inability of current materials to satisfy Inequality 2.12 becomes apparent for the atmospheric conditions on Mars. Therefore, with current materials, a homogeneous shell design for the vacuum airship is infeasible. There is very little likelihood of materials technology advancing far enough for these limitations to be surpassed since they are so extreme. However, this analysis is still useful for other designs, as will be seen below.

Sandwich Shell: Lattice

The next iteration of the evacuated airship's design sought to find a solution to the problem of shell buckling while maintaining a lightweight design. In order to increase resistance of the monocoque shell to buckling, a sandwich structure may be employed. Said sandwich structure increases the effective stiffness of the material, which in turn reduces the susceptibility of the material to elastic instability, i.e. buckling. The increase in effective stiffness is due to the spacing of two layers of material created by the addition of a core. For the design discussed in this section, a core comprised of a lattice structure was chosen and subsequently analyzed.

Extending the analysis performed for the homogenous shell, the forces acting on the sandwich shell can be represented as shown in Figure 3.1. The shell equilibrium can be shown with Equation 3.1, which is very similar to the equilibrium equation developed for a homogeneous shell. Rearranging Equation 3.1 and assuming both shell layers support the same stress gives Equation 3.2, representing the stress in the shell depending on the geometry and atmosphere.

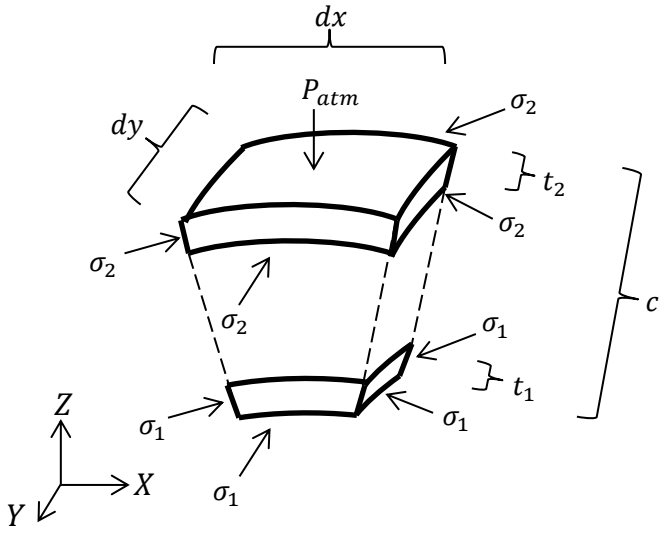


Figure 3.1

$$\sum F_z = 0 = P_{atm}dxdy - \left[4\sigma_2 t_2 \cdot dydx\frac{1}{2r} + 4\sigma_1 t_1 \cdot dydx\frac{(r-c)}{2r^2}\right] \tag{3.1}$$

$$\sigma = \frac{P_{atm}}{2}\frac{r}{t_1\frac{r-c}{r} + t_2} \tag{3.2}$$

To determine the load held by the lattice structure, Figure 3.1 must be modified slightly, shown in Figure 3.2, to account for the force transfer done by the lattice instead of simply assuming the load is transferred, as prior. From the balance of forces in Figure 3.2, equilibrium can be represented by Equation 3.3, using much the same process as above. Again, assuming both layers of material support the same stress, Equation 3.3 can be rearranged and subsequently combined with Equation 3.2 to represent the core stresses as shown in Equation 3.4.

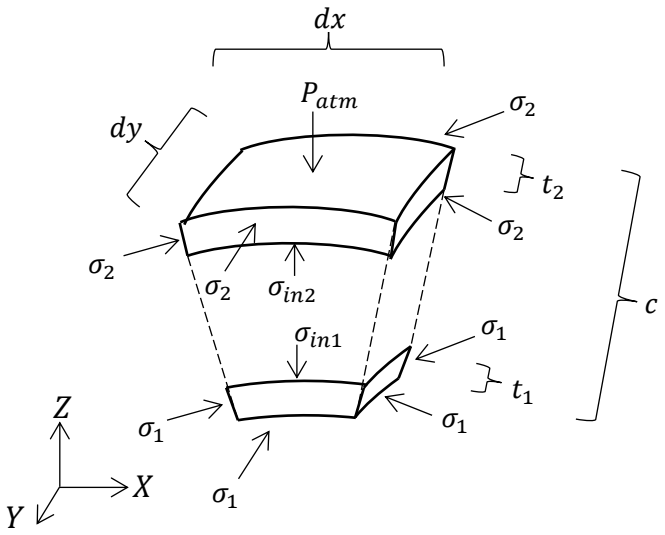


Figure 3.2

$$P_{atm}dxdy = 2\sigma_2 t_2 dydx\frac{1}{r} + \sigma_{in1}dxdy\left(\frac{r-c}{r}\right)^2 \tag{3.3}$$

$$\sigma_{in1} = \frac{2t_1\sigma}{r-c} = P_{atm}\frac{t_1}{t_2 + t_1\frac{r-c}{r}}\frac{r}{r-c} = \sigma_{in2}\left(\frac{r-c}{r}\right)^2 \tag{3.4}$$

Additional consideration must be made to the lattice structure's effective properties. The lattice structure, in order to properly transmit the force between the inner and outer layers of the shell, must deform consistently with the other material layers. This deformation can be seen in Figure 3.3, which shows a two-dimensional cross section of a differential portion of the shell in deformed and original forms. From the Hooke's law and the circumference of a circle, Equation 3.5 can be derived to represent the outer layer's deformation. Using Equation 3.5, a formula for the deformed outer radius can be found, shown in Equation 3.6. In much the same way, the deformed inner radius can be found, as shown in Equation 3.7.

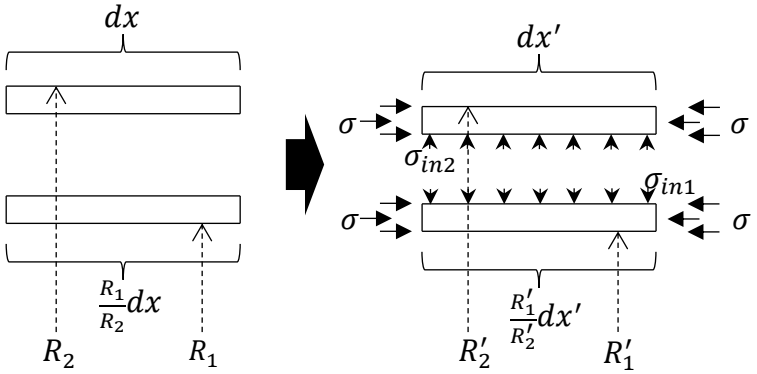


Figure 3.3

$$\frac{\sigma}{E} = \frac{dx' - dx}{dx} = 2\pi \frac{R_2' - R_2}{R_2} \tag{3.5}$$

$$R_2' = \left(\frac{\sigma}{2\pi E} + 1\right) R_2 \tag{3.6}$$

$$R_1' = \left(\frac{\sigma}{2\pi E} + 1\right) R_1 \tag{3.7}$$

Using the definition of strain, the effective strain imposed on the lattice can be seen in Equation 3.8, substituting Equations 3.6 and 3.7 and simplifying. Assuming the stresses in the lattice have approximately a linear distribution, Equation 3.8 can be used to create Equation 3.9 from Hooke's law. Equation 3.9 can then be rearranged to give an expression for the effective stiffness of the lattice as imposed by the geometry and shell material properties, given in Equation 3.10.

$$\varepsilon_{lattice} = \frac{(R_2' - R_1') - (R_2 - R_1)}{R_2 - R_1} = \frac{\sigma}{2\pi E} \tag{3.8}$$

$$\sigma_{in} = E_{lattice}\varepsilon_{lattice} = \frac{E_{lattice}}{E}\frac{\sigma}{2\pi} \tag{3.9}$$

$$E_{lattice} = \frac{2\pi\sigma_{in}E}{\sigma} \tag{3.10}$$

The lattice geometry for this design was chosen to be a tetrahedral based lattice with unit cells as can be seen in Figure 3.4. Definition for the geometry in relation to a single variable, *a*, can also be seen in the accompanying Equation 3.11. From these, a full description of the cell volume can be obtained, as well as the material volume occupied by the lattice, supplemented with additional geometric constraints as will be shown.

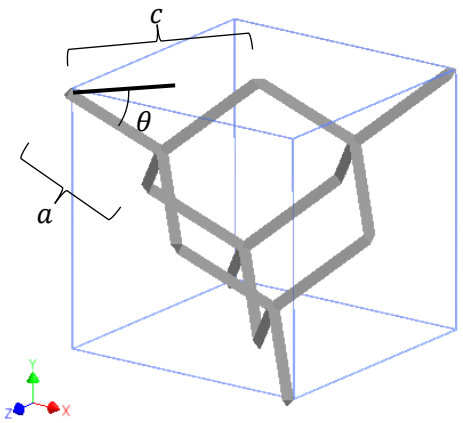


Figure 3.4

$$\hat{b} = \frac{4}{\sqrt{6}}a \tag{3.11}$$

In order to determine the effective deformation of a general lattice, the simple lattice model in two-dimensions shown in Figure 3.5 is used. Assuming the lengths of all the members are equal with equal cross sectional areas and then balancing the forces for equilibrium results in the same force being applied to each member. Consequently, the strains on each member are equivalent, as denoted in Equation 3.12. Applying the definition of strain to the simple lattice's total strain, replacing the terms with their components based on the geometry, and simplifying shows the strain on the entire lattice is equivalent to the strain experienced by the individual members, shown in Equation 3.13. This analysis can be easily extended to the general case since the only assumptions made were the equal lengths and cross sectional areas of the members. Therefore, as long as angles between the members are all equal, then the strain equivalence will be true if the lattice is in equilibrium for both two- and three-dimensional lattice structures.

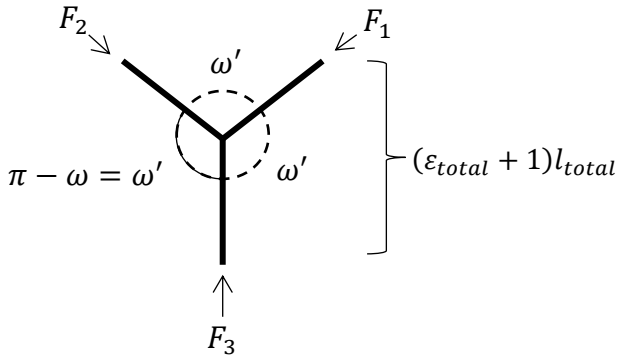


Figure 3.5

$$F_1 = F_2 = F_3 = F$$

$$l_1 = l_2 = l_3 = l \tag{3.12}$$

$$\therefore \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon$$

$$\varepsilon_{total} = \frac{l'_{total} - l_{total}}{l_{total}} = \frac{[l(1+\varepsilon) + l(1+\varepsilon)\cos\omega] - (l + l\cos\omega)}{(l + l\cos\omega)} \tag{3.13}$$

$$= \frac{(1+\varepsilon)l(1+\cos\omega) - l(1+\cos\omega)}{l(1+\cos\omega)} = \varepsilon$$

Using the geometry from Figure 3.4, the balance of forces on the lattice cell can be found in order to determine the forces on the lattice members. The aforementioned balance of forces is shown in Equation 3.14. Based on the geometry, the angle in Equation 3.14 can be described as shown in Equation 3.15. From the combination of Equations 3.14 and 3.15, the stress on the lattice members can be represented as shown in Equation 3.16. Knowing the stress on each lattice member permits the calculation of the strain of the members from Hooke's law. Due to the result in Equation 3.13, the strain in the members is equivalent to the strain in the lattice as a whole, which results in Equation 3.17 representing the lattice strain. Since both the strain on the lattice and the stress supported by the lattice are defined, again, using Hooke's law, the effective stiffness of the lattice, shown in Equation 3.18, can be resolved. Combining this formula for effective stiffness with the previous representation of the effective stiffness of the lattice as necessitated by the force transfer between shell layers and rearranging results in Equation 3.19, defining the cross sectional area of the lattice members.

$$2\hat{b}^2\sigma_{in} = 4F_{lattice}\sin\theta \tag{3.14}$$

$$\theta = \cos^{-1}\frac{\hat{b}}{2a} = \cos^{-1}\frac{\frac{4}{\sqrt{6}}a}{2a} = \cos^{-1}\frac{2}{\sqrt{6}} \tag{3.15}$$

$$\sigma_{lattice} = \frac{F_{lattice}}{A_{lattice}} = \frac{8a^2}{\sqrt{3}A_{lattice}}\sigma_{in} \tag{3.16}$$

$$\varepsilon_{lattice} = \varepsilon = \frac{\sigma_{lattice}}{E} = \frac{8a^2}{\sqrt{3}A_{lattice}}\frac{\sigma_{in}}{E} \tag{3.17}$$

$$E_{lattice} = \frac{\sigma_{in}}{\varepsilon_{lattice}} = \frac{\sqrt{3}A_{lattice}E}{8a^2} \tag{3.18}$$

$$A_{lattice} = \frac{16a^2\pi}{\sqrt{3}}\frac{t_1[r^2 + (r-c)^2]}{r^2(r-c)} \tag{3.19}$$

Additional constraints on the lattice geometry are driven by the elastic stability of the lattice members. Figure 3.6 shows the buckling analysis model for the lattice as well as the cross section definition for the lattice members. An annular cross section was chosen for the ability to change the susceptibility of the beam to buckling while maintaining the same cross sectional area, as will be shown shortly. From the definition of column buckling, the critical force on the members can be represented as seen in Equation 3.20 for the geometry of the lattice members. The second moment of area for an annular cross section can be seen in Equation 3.21, which can be reposed in terms of area as shown. Combining the definition of the area of an annulus and Equations 3.19, 3.20, and 3.21 allows definition of the outer radius of the lattice member, shown in Equation 3.22 in terms of the geometry and a ratio between the two radii. This ratio of radii can also be found in terms of the geometry and the force in the lattice as a minimum value in order to prevent buckling of the lattice member, shown in Inequality 3.23.

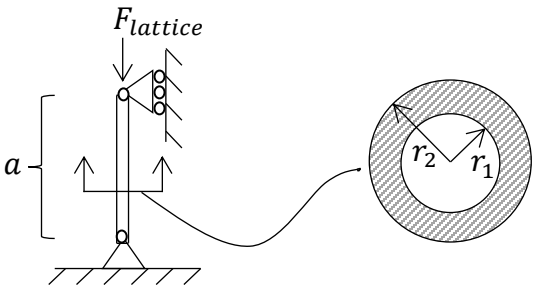


Figure 3.6

$$P_{cr} = \frac{\pi^2 EI}{l^2} = \frac{\pi^2 EI}{a^2} \tag{3.20}$$

$$I = \frac{\pi}{4}(r_2^4 - r_1^4) = \frac{r_2^2 + r_1^2}{4} A_{lattice} \tag{3.21}$$

$$r_2 = \frac{4a}{3^{1/4} r} \sqrt{\frac{t_1[r^2 + (r-c)^2]}{(r-c)(1-\psi^2)}} \qquad where \quad \psi \equiv \frac{r_1}{r_2} \tag{3.22}$$

$$\psi \leq \sqrt{\frac{\sqrt{3}\sigma_{in} r^4 (r-c)^2 - 8E\pi^3 t_1^2 [r^2 + (r-c)^2]^2}{\sqrt{3}\sigma_{in} r^4 (r-c)^2 + 8E\pi^3 t_1^2 [r^2 + (r-c)^2]^2}} \qquad where \quad \sigma_{in2} \leq \sigma_{in} \leq \sigma_{in1} \tag{3.23}$$

Returning to the geometry of the lattice cell as defined in Figure 3.4, the solid volume occupied by the lattice can be represented as shown in Equation 3.24, as stated earlier. Additionally, the volume occupied by the lattice cell can be represented as shown in Equation 3.25 in relation to the geometric variable *a*. The ratio between these two volumes can be denoted as shown in Equation 3.26 with the lattice member cross sectional area being substituted from Equation 3.19. This ratio gives the proportion of the volume occupied by the lattice as a whole which is actually occupied by material.

$$V_{solid} = a \cdot A_{lattice} \cdot 16 \tag{3.24}$$

$$V_{cell} = \left(\sqrt{2}\hat{b}\right)^3 = \frac{64}{3\sqrt{3}} a^3 \tag{3.25}$$

$$\gamma_V = \frac{V_{solid}}{V_{cell}} = \frac{12\pi[r^2 + (r-c)^2]t_1}{r^2(r-c)} \tag{3.26}$$

Using Equation 3.26, the total mass of the airship can be determined. Here, neutral buoyancy will be assumed as well as a perfect evacuation of the airship and no payload. This is the same threshold case used for the determining the feasibility of the earlier monocoque design. As such, the mass of the airship can be computed as shown in Equation 3.27. For neutral

buoyancy, the mass displaced is equal to the mass of the airship. Using Equations 3.27 and 2.3, Equation 3.28 can be created representing neutral buoyancy, which can be further simplified to Equation 3.29.

$$m_{ship} = \frac{4}{3}\pi\{r^3 - (1-\gamma_V)[(r-t_2)^3 - (r-c+t_1)^3] - (r-c)^3\}\rho_{shell} \tag{3.27}$$

$$\frac{4}{3}\pi r^3\rho_{atm} = m_{ship} = \frac{4}{3}\pi\{r^3 - (1-\gamma_V)[(r-t_2)^3 - (r-c+t_1)^3] - (r-c)^3\}\rho_{shell} \tag{3.28}$$

$$r^3\rho_{atm} = \{r^3 - (1-\gamma_V)[(r-t_2)^3 - (r-c+t_1)^3] - (r-c)^3\}\rho_{shell} \tag{3.29}$$

Here, observing the possible instabilities associated with the sandwich structure, the thicknesses of the two sandwich faces should be equal, defined in Equation 3.30. Doing so reduces the risk of the elastic instability associated with wrinkling of the faces since each face has the same stiffness and therefore neither is more susceptible to instability. Now, using the methods described by Bruhn[1], the bending stiffness of the sandwich structure can be represented as shown in Equation 3.31 with the accompanying model of the sandwich structure shown in Figure 3.7. The parameter lambda in this equation (3.31) is a property of the material as defined in Equation 3.32. Additionally, the effective stiffness, *E'*, of the plates can be represented as in Equation 3.33: a function of the geometry and material properties.

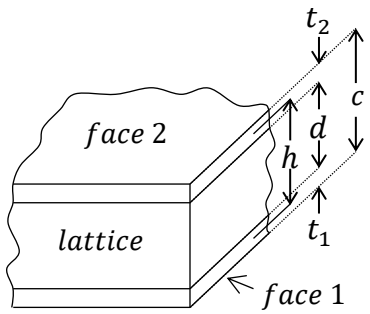


Figure 3.7

$$t_1 = t_2 = t_f \tag{3.30}$$

$$D = \frac{E' t_f h^2}{2\lambda} + \frac{E' t_f^3}{6\lambda} \tag{3.31}$$

$$\lambda = 1 - \nu^2 \tag{3.32}$$

$$E' = E\frac{2t_f}{c} = E\frac{2t_f}{2t_f + d} \tag{3.33}$$

The effective stiffness of the lattice structure as a whole can be represented as shown in Equation 3.34, based on the bending stiffness and geometry. This effective stiffness can now be injected into the same equation for critical buckling pressure of a sphere (2.8) used in the earlier analysis. Combining these two equations results in Equation 3.35, representing the critical buckling pressure of a sandwich based monocoque sphere as a function of the effective bending stiffness, geometry, and material. Equation 3.36 is formed by applying the previously used threshold conditions and combining Equations 3.31, 3.32, 3.33, and 3.35, which represents the final criteria for this airship design to withstand global buckling.

$$E_s = \frac{12(1-\nu^2)D}{t^2 \cdot 2t_f} \qquad where \;\; t = c = 2t_f + d \tag{3.34}$$

$$P_{cr} = \frac{4\sqrt{3(1-\nu^2)}D}{r^2 \cdot t_f} \tag{3.35}$$

$$P_{cr} = \frac{4E}{\sqrt{3(1-\nu^2)}} \frac{t_f\left(3c^2 - 6c \cdot t_f + 4t_f^2\right)}{c \cdot r^2} = P_{atm} \tag{3.36}$$

Due to the greater design space allowed for this design by the addition of the sandwich parameters, this design is not as constrained by the material properties available. That is to say, the insufficiency of material properties can be circumvented by modulating the thickness of the core in relation to the other geometric variables. In this way, both criteria denoted in Equations 3.29 and 3.36 can be met with current materials. However, under further scrutiny, the thickness required for the lattice members, in order to meet the criteria mentioned earlier, is thinner than current manufacturing processes are capable of. Therefore this design, though possible, is currently confined to theory due to manufacturing limitations. But the design still remains conditionally viable, should materials technology advance to a point encompassing these manufacturing needs. If this were to be the case, this design would be highly beneficial due to the exceedingly high associated payload capacity. These advancements would likely be associated with composite materials, due to their high strength and stiffness for a given mass.

Sandwich Shell: Honeycomb

For the next design iteration, the goal was to find a viable solution to the material manufacturing limitations associated with the previous design. The design described herein uses a sandwich structure to increase resistance of the monocoque shell to buckling, much like the lattice based sandwich shell described above. Though the same method is employed in order to reduce susceptibility of the shell to elastic instability, here the core is comprised of a honeycomb structure. Implementation of the honeycomb structure in this design is chosen to possibly avoid the material limitations encountered with the lattice structure as well as avoid the increased complexity of the lattice to manufacture and transport. This implementation of the honeycomb was also chosen for the possibility of implementation of aerogel-like materials or materials with similar characteristics of extremely low density with reasonable strength and stiffness.

Much of the analysis necessary for the honeycomb sandwich structure is similar to that of the lattice based structure, and will be used correspondingly. However, the honeycomb core of

this sandwich structure brings other stability considerations to account for. Where the lattice stability was examined through local buckling of the individual lattice members, here, the honeycomb must be examined for wrinkling and intra-cell buckling as well as plate buckling of the individual cell walls. Analysis for these modes of instability is based the work by Bruhn[1], with the geometry of the honeycomb sandwich structure as shown in Figure 4.1.

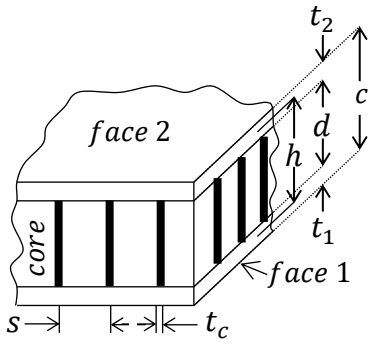


Figure 4.1

Wrinkling in the faces of the sandwich structure can be determined using Equation 4.1, which shows the critical wrinkling stress for the honeycomb sandwich structure. Here, as assumed prior, the stress in all directions is equal since the structure is symmetric on the faces of the sandwich structure. The effective core stiffness, E_c', is defined as shown in Equation 4.2 as a function of the honeycomb cell geometry as laid out in Figure 4.2. Additionally, the effective rigidity of the core, G_c', is defined in Equation 4.3, depending on the honeycomb cell geometry. Combining Equations 4.1, 4.2, 4.3 and the previous equation for the stress in the sandwich faces (3.2) results in Equation 4.4: the critical wrinkling pressure for the honeycomb sandwich structure as used in a monocoque sphere.

$$\sigma_{cw} = \frac{0.43(E \cdot E_c' \cdot G_c')^{1/3}}{\left(1 + \left(\frac{\sigma_y}{\sigma_x}\right)^3\right)^{1/3}} = \frac{0.43(E \cdot E_c' \cdot G_c')^{1/3}}{2^{1/3}} \tag{4.1}$$

$$E_c' = E_c \cdot \left(\frac{2 \cdot s \cdot t_c - t_c^2}{s^2}\right)^{1.415} \cdot 2.13 \tag{4.2}$$

$$G_c' = G_c \cdot \frac{2 \cdot s \cdot t_c - t_c^2}{s^2} \frac{1}{3} \tag{4.3}$$

$$P_{cw} = \frac{2t_f(2r - c)}{r^2} \left(\frac{2 \cdot s \cdot t_c - t_c^2}{s^2}\right)^{0.805} (E \cdot E_c' \cdot G_c')^{1/3} \cdot 0.305 \tag{4.4}$$

Addressing the intra-cell buckling of the sandwich structure can be accomplished using Equation 4.5 for the intra-cell buckling stress. Here, the same assumption of equivalent strain in all directions can be used. Intra-cell buckling is a buckling mode of the face in the space within

the honeycomb cells, unsupported by the honeycomb structure. Combining Equation 4.5 with the equation for the stress in the sandwich faces (3.2) results in Equation 4.6 for the critical intra-cell buckling pressure on the sphere.

$$\sigma_{ci} = \frac{0.75E}{\left(1+\left(\frac{\sigma_y}{\sigma_x}\right)^3\right)^{1/3}}\left(\frac{t_f}{s}\right)^{3/2} = \frac{0.75E}{2^{1/3}}\left(\frac{t_f}{s}\right)^{3/2} \tag{4.5}$$

$$P_{ci} = \frac{2t_f(2r-c)}{r^2}\cdot\frac{0.75E}{2^{1/3}}\left(\frac{t_f}{s}\right)^{3/2} \tag{4.6}$$

Using the results from Equation 4.4, a representation for the thickness of the honeycomb walls can be developed as shown in Equation 4.7 based on the wrinkling criteria. In much the same way, a formula for the honeycomb cell size can be created as shown in Equation 4.8 based on the intra-cell buckling criteria from Equation 4.6. Now, based on the geometry from Figure 4.2, a ratio for the proportion of the space occupied by the honeycomb which is actually filled by material can be defined as seen in Equation 4.9. Subsequently, the values for wall thickness and cell size from Equations 4.7 and 4.8 can be substituted into Equation 4.9 and simplified, as shown.

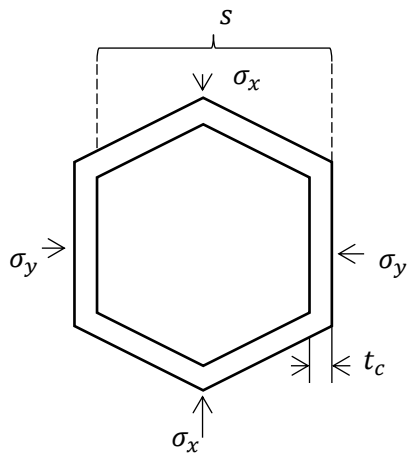


Figure 4.2

$$t_c = s\left\{1-\sqrt{1-\left[\frac{5r^2P_{atm}}{3(E\cdot E_c\cdot G_c)^{1/3}(2r-c)t_f}\right]^{5/4}}\right\} \tag{4.7}$$

$$s = \left[\frac{9E^2}{2^{8/3}P_{atm}^2}\frac{(2r-c)^2}{r^4}t_f^5\right]^{1/3} \tag{4.8}$$

$$\gamma_V = \frac{2 \cdot s \cdot t_c - t_c^2}{s^2} = \left[\frac{5P_{atm}r^2}{3(E \cdot E_c \cdot G_c)^{1/3}(2r - c)t_f}\right]^{5/4} \tag{4.9}$$

Local buckling within the honeycomb is addressed by modeling each wall of the cells in the honeycomb structure as a simply supported plate. This model is shown in Figure 4.3, and using this model, the plate buckling stress can be represented as shown in Equation 4.10. Converting between the dimensions in Figures 4.3 and 4.2, the geometry for plate buckling can be put in terms of the honeycomb geometry as defined in Equation 4.11. The buckling coefficient in Equation 4.10 is defined for a simply supported plate as seen in Equation 4.12, which depends on the ratio between the height and width of the plate since this ratio determines the propensity of the plate to develop full waves, as can be observed in the equation. To better observe the behavior of the buckling coefficient in terms of the plate dimensions, the value of the buckling coefficient has been plotted for various aspect ratios of the plate in Figure 4.4. Combining Equations 4.10, 4.11, and 4.12 results in a full expression for the plate buckling stress in the honeycomb cell walls, shown in Equation 4.13.

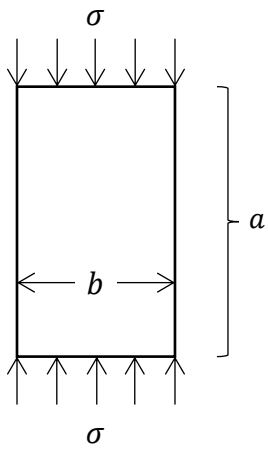


Figure 4.3

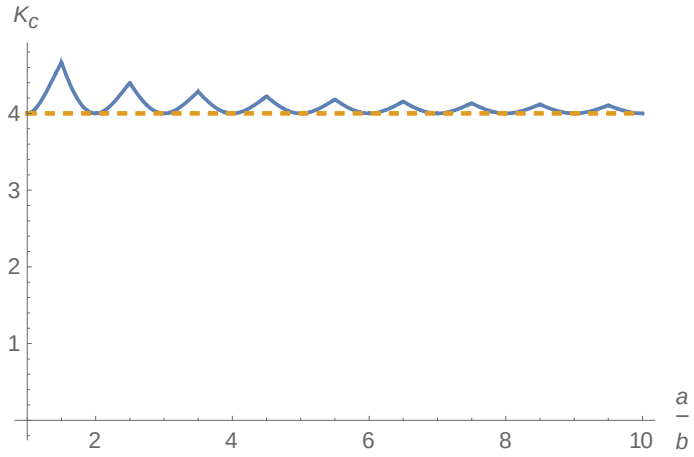


Figure 4.4

$$\sigma_{cp} = \frac{\pi^2 K_c E_c}{12(1 - \nu^2)}\left(\frac{t_c}{b}\right)^2 \tag{4.10}$$

$$a = c - 2t_f \quad ; \quad b = \frac{1}{\sqrt{3}}s \tag{4.11}$$

$$K_c = \frac{b}{a}\left\{1 - \text{abs}\left[\cos\left(\pi\frac{a}{b}\right)\right]\right\} + 4 \tag{4.12}$$

$$\sigma_{cp} = \frac{\pi^2 E_c}{4(1 - \nu^2)}\left(\frac{t_c}{s}\right)^2 \cdot \left\{4 + \frac{s}{\sqrt{3}(c - 2t_f)}\left(1 - \text{abs}\left[\cos\left(\pi\frac{\sqrt{3}(c - 2t_f)}{s}\right)\right]\right)\right\} \tag{4.13}$$

The stress experienced by the honeycomb core of the sandwich structure can be found from the product of the stress transmitted by the core (3.4) and the core volume ratio (4.9). This stress in the honeycomb is shown in Equation 4.14, relying on Equations 3.4 and 4.9 from earlier development. Rearranging Equation 4.14 and applying the critical plate buckling stress from Equation 4.13 results in Equation 4.15, expressing the critical plate buckling pressure for the honeycomb sandwich structure of the monocoque sphere.

$$\sigma_c = \sigma_{in} \cdot \gamma_V = \frac{r-c}{2r-c} P_{atm} \gamma_V \tag{4.14}$$

$$P_{cp} = \frac{\sigma_{cp}}{\gamma_V} \frac{2r-c}{r-c} \tag{4.15}$$

The global buckling for this sandwich structure can be represented in exactly the same way the previous sandwich structure's global stability was modeled. That is, global stability for the honeycomb sandwich structure can be determined using Equation 3.36. As such, the entirety of the sandwich structure's stability is accounted for.

Regrettably, once all of the constraints from the various modes of elastic instability have been applied to the design, there is no material which is able to meet the criteria while remaining within manufacturing constraints. Even when using two different materials for the face and core of the sandwich structure, and thereby tailoring each to the specific needs of the respective component, has no viable solution. As such, the evacuated airship design utilizing a monocoque sphere with honeycomb sandwich shell is not feasible given current materials. Therefore, other options must be explored for structural implementation in the vacuum airship design.

However, this design, like the previous sandwich structure based shell, is still viable if materials advance to a point where the aforementioned constraints can be met. For this design, enabling technology would be associated with advances in materials comparable to aerogels. That is to say, advances in materials with extraordinarily low densities while retaining reasonable strength and stiffness. Therefore, between the sandwich structures described here, there are viable avenues for the evacuated airship for future advancements in either composites or low density materials. For current materials to be a viable option, further design iterations were necessary.

Tensegrity Based Approach

In order to create a feasible design for the evacuated airship using currently available materials, further iterations of the design were necessary. For the subsequent iterations of the design, an entirely different method was needed to deal with the forces on the structure and the resulting inherent elastic instabilities.

Moving away from the monocoque designs, the current rendition of the design makes use of a tensegrity structure which supports a membrane to maintain the vacuum. Using this method of supporting atmospheric pressure mitigates the problem of shell buckling, however the beams used in the tensegrity macro-structure are still subject to buckling. By employing several different mass saving methods to the design, the total structural mass can be reduced to a point where the system has neutral buoyancy as well as a net lift in order to incorporate useful payload. A twelve beam tensegrity design was chosen because higher numbers of beams begin to become susceptible to shell buckling modes and lower numbers of beams lose efficiency of mass displaced, support, and mass savings.

For the tensegrity design, the three dimensional shape of the airship is modeled using a regular truncated octahedron for simplicity, as shown in Figure 5.1 for a clear and opaque solid, even though a more structurally optimum shape is a slight variation on the truncated octahedron. Using this geometry, the length of the beams can be represented as shown in Equation 5.1. Subsequently, formulae for the volume and surface area of the airship can be found, shown in Equations 5.2 and 5.3 respectively, as a function of beam length.

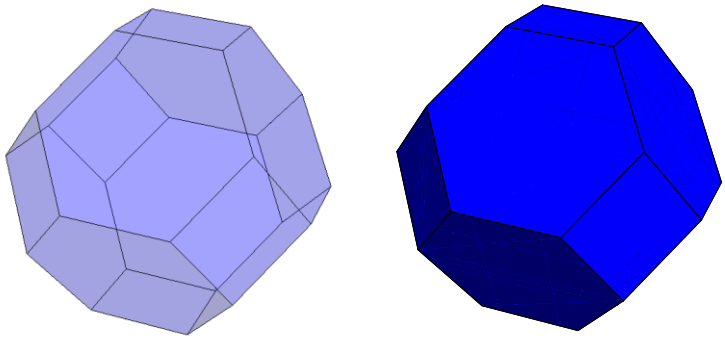


Figure 5.1

$$l_{beam} = \sqrt{6}a \tag{5.1}$$

$$V = 8\sqrt{2}a^3 = \frac{4}{3\sqrt{3}} l_{beam}^3 \tag{5.2}$$

$$A_{surf} = \left(6 + 12\sqrt{3}\right)a^2 = \left(1 + 2\sqrt{3}\right)l_{beam}^2 \tag{5.3}$$

As such, the mass displaced by the tensegrity design can be derived as the product of atmospheric density and volume of the truncated octahedron, denoted in Equation 5.4. Additionally, the total load on the beams can be calculated from the product of atmospheric pressure and the surface area of the airship. Each beam can be assumed to carry virtually the

same load as the other beams because of the symmetry of the shape. Therefore the total load is distributed equally to each beam, resulting in Equation 5.5 for beam load.

$$m_{disp} = \rho_{atm} \frac{4}{3\sqrt{3}} l_{beam}^3 \tag{5.4}$$

$$F_{beam} = \frac{1}{12}\frac{F_{total}}{2} = \frac{P_{atm}A_{surf}}{24} = \frac{P_{atm}}{24} l_{beam}^2\left(1 + 2\sqrt{3}\right) \tag{5.5}$$

Using the column buckling formula, shown in Equation 5.6, an expression for the mass required for beams of the tensegrity structure can be developed. Therefore, a minimal mass beam with circular cross section has a cross sectional area as shown in Equation 5.7, from rearranging the Euler buckling formula. For minimal mass of the beams, the load on the beams should be equivalent to the buckling load of the beams. Applying the previous statement, Equations 5.5 and 5.7 can be combined to obtain Equation 5.8 as a solution for the beam's cross sectional area required for a given beam length.

$$F = \frac{\pi^2 EI}{l^2} = \frac{\pi^2 E}{l^2} \cdot \frac{\pi r^4}{4} \tag{5.6}$$

$$A_{beam} = \pi r^2 = \sqrt{\frac{F}{\pi E}} \cdot 2l_{beam} \tag{5.7}$$

$$A_{beam} = \sqrt{\frac{P_{atm}\left(1 + 2\sqrt{3}\right)}{6\pi E}} l_{beam}^2 \tag{5.8}$$

Assuming the mass of the membrane is negligible for now as are losses in displaced mass due to deformation of the membrane and imperfect vacuum the airship mass is equivalent to the structural mass; later, the mass of the membrane and losses will be accounted for using other methods. Using the above simplification, Equation 5.9 gives the structural mass of this airship design based on the scale and material properties. Applying a neutral buoyancy threshold as done with the previous designs, Equations 5.4 and 5.9 are combined, resulting in Inequality 5.10 which gives the minimum material criteria for the tensegrity design.

$$m_{struct} = 12\rho_{struct} l_{beam} A_{beam} = 12\rho_{struct} l_{beam}^3 \sqrt{\frac{\left(1 + 2\sqrt{3}\right)P_{atm}}{6\pi E}} \tag{5.9}$$

$$\frac{P_{atm}}{\rho_{atm}^2} \leq \frac{2\pi}{81\left(1 + 2\sqrt{3}\right)} \frac{E}{\rho_{struct}^2} \tag{5.10}$$

As before, these criteria are well beyond current materials; however, there are many methods which may be employed to greatly reduce the mass of the structure. Therefore, by using such methods, as described below, the material limitations may be mitigated. For reference, renderings of both the design and the tensegrity internal structure can be seen in Figures 5.2 and 5.3 respectively.

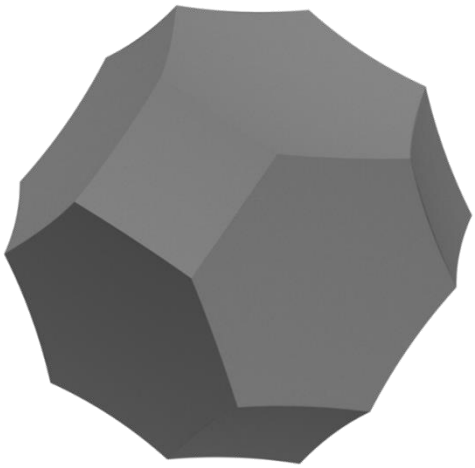


Figure 5.2

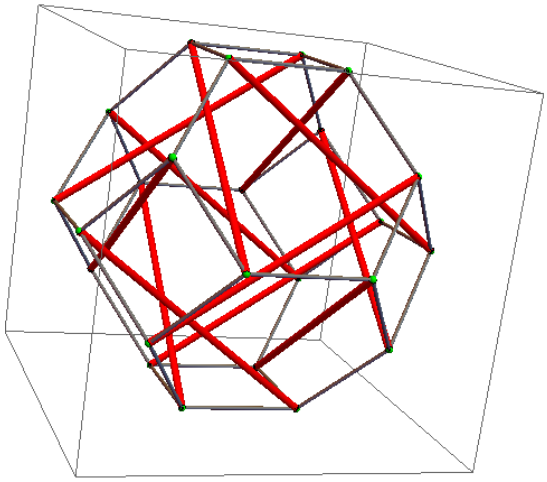


Figure 5.3

Minimal Mass Tensegrity Beams

One method by which beam mass can be reduced is replacing the beams with tensegrity beam structures, from Skelton and Montuori[2]. Here follows an abbreviated overview of the method used, since the original method has been slightly modified for three-dimensional implementation. Mass savings from this method are achieved by reinforcement of the beam at key displacement points in the beam, therefore reducing the susceptibility of the beam system to local buckling as well as global buckling. Minimal mass is further achieved for the design by sizing the components to fail under the same total external load, ensuring no component is over-engineered.

Each beam in the tensegrity structure can be sized using a similar process as that in Equations 5.6, 5.7, and 5.8. The final product of this method is shown in Equation 6.1, representing the mass of a beam for a given compressive load, length, and material respectively. In much the same way, tensile members of the structure can be represented using Equation 6.2 for a given tensile load, length, and material respectively. Under the additional constraint of approximately constant total length, as in Skelton and Montuori[2], the pre-stress loads in the structure can be found using Equation 6.3, dependent on the corresponding compressive load and angle to the tensile member. From Equation 6.3, the load on the reinforcing beams as well as their lengths can be found from Equations 6.4a and 6.4b respectively.

$$m_b = A \cdot l \cdot \rho_b = \sqrt{f} l^2 \frac{2\rho_b}{\sqrt{\pi E_b}} \tag{6.1}$$

$$m_{s,i} = A_{t,i} s_i \rho_s = t_i s_i \frac{\rho_s}{\sigma_s} \tag{6.2}$$

$$t_i = \frac{f_0}{2\cos(\alpha_i)} \tag{6.3}$$

$$f_{v,i} = f_0 \tan(\alpha_i) \tag{6.4a}$$

$$l_{v,i} = l_i \tan(\alpha_i) \tag{6.4b}$$

In order to extend the method developed by Skelton and Montuori[2] to three-dimensions, there must be buckling support implemented in an orthogonal direction. Here, the simplest example is used to demonstrate this extension, shown in Figure 6.1 as the two-dimensional model from Skelton and Montuori[2] and in Figure 6.2 as extended to three-dimensions. This three-dimensional model is simply the addition of an identical set of supports as in Figure 6.1 rotated to be orthogonal to the plane of the two-dimensional model. To account for fixing the structure in three-dimensions, four new cables are added connecting the supporting beams. Advantageously, this extension to three-dimensions does not affect the physics of the method; therefore the same process for design can be used for both cases since exclusively the mass calculations are affected by the addition of the new supports.

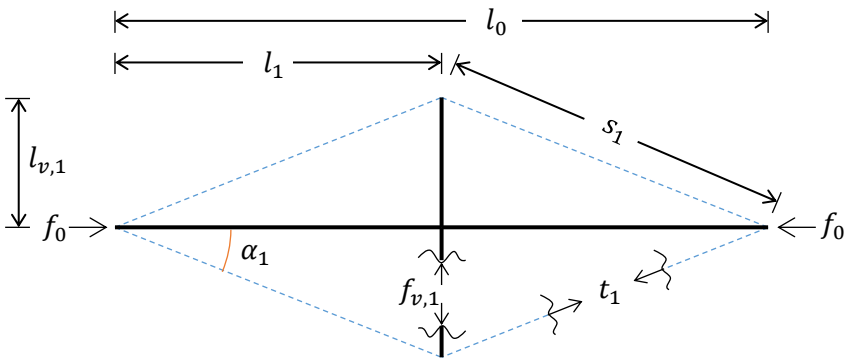


Figure 2.1

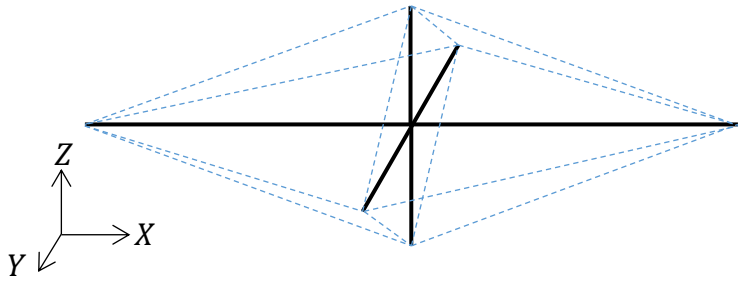


Figure 2.2

As an extension of the method used by Skelton and Montuori[2] has been developed in order to obtain further weight savings: iteratively applying the same principles of reinforcement, increasing the complexity of the design. Figure 6.3 shows this extension to higher levels of complexity in two-dimensions, with the same extension for three-dimensions described above still applicable. This extension uses slightly different notation than the original, however these differences can be seen in Figure 6.3 as well as Figure 6.4, which shows the schematization of

the model in Figure 6.3. Unless otherwise noted, the rest of the notation is consistent with the original method.

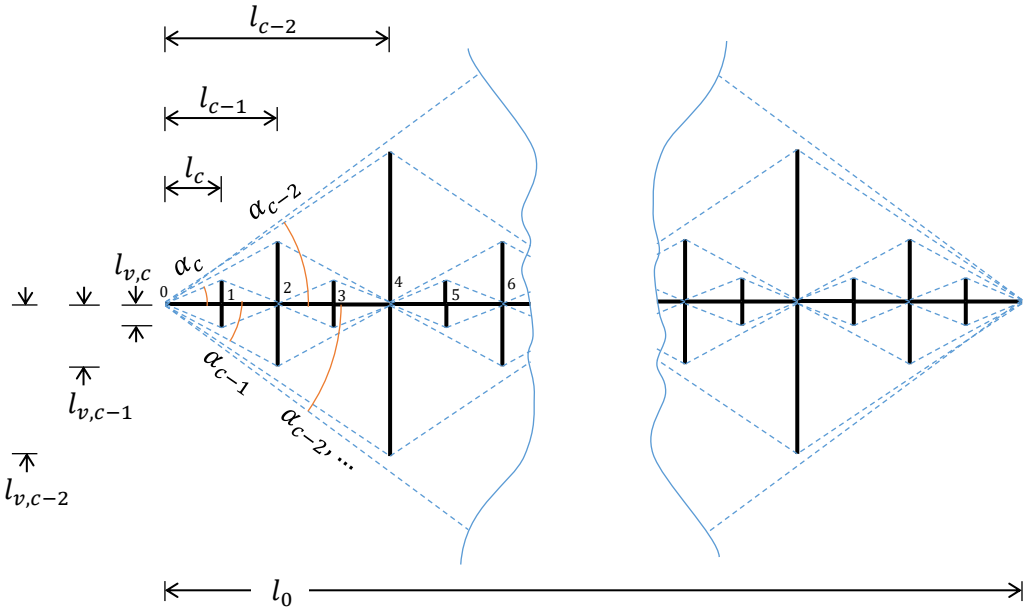


Figure 6.3

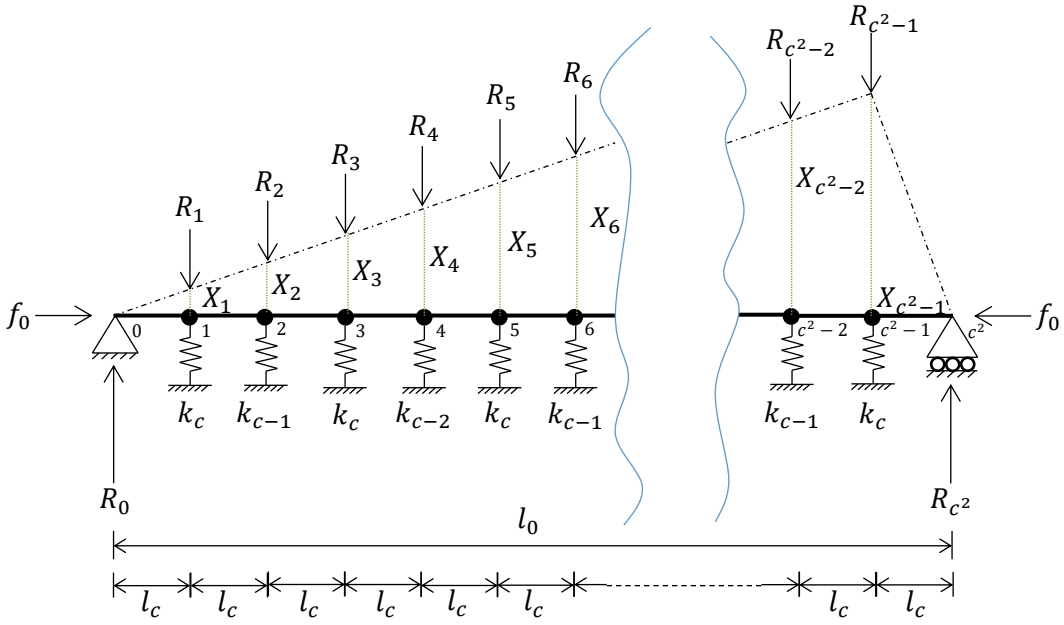


Figure 6.4

Equation 6.1 is sufficient to address local stability of the structure for higher complexity designs; therefore the only issue to be resolved is that of global buckling. To determine the global stability of the structure, the rotational equilibrium equations for each of the joints must be

found. Here, the complexity of the structure will be denoted *c*, which is the number of iterations the method has been applied; to see the evolution of increasing the complexity on the tensegrity beam design in three dimensions, refer to Figure 10.1. The reaction force at each join in the structure can be represented by Equation 6.5 with the relevant spring constant equivalents represented in Equation 6.6. Rotational equilibrium can therefore be represented at each joint by Equation 6.7. Additionally, for a given complexity *c*, there will be 2^c joints and consequently 2^c simultaneous equations.

$$R_i = K_i X_i \tag{6.5}$$

$$K_i = 2k_{c-b_i} \cdot \sin^2\left(\alpha_{c-b_i}\right) \qquad with\ \ b_i\ \ s.t.\ \ \frac{i}{2^{b_i}} \in Odd \tag{6.6}$$

$$R_0 \cdot i \cdot l_c - N \cdot X_i - \sum_{j=1}^{i} (i-j) R_j l_c = 0 \tag{6.7}$$

Solving Equation 6.7 for the first joint in the beam gives the reaction force at one end of the structure as shown in Equation 6.8. Then combining Equations 6.5, 6.7, and 6.8, the equilibrium of the system for each joint becomes as shown in Equation 6.9. This leads to one less equation in the total system of equations, i.e. Equation 6.9 is applicable for integer values of *i* from one to 2^c*-1*.

$$R_0 = N \frac{X_1}{l_c} \tag{6.8}$$

$$N \cdot X_1 i - N \cdot X_i - \sum_{j=1}^{i} (i-j) K_j X_j l_c = 0 \tag{6.9}$$

In order to determine the critical global buckling load, Equation 6.9 must be satisfied for all joints in the structure in the non-trivial case, where not all displacements are zero. Solutions for the system of equations based on Equation 6.9 can be given by the determinate of the coefficient matrix equal to zero. Said coefficient matrix can be represented as shown in Equation 6.10, which is separated into three components for simplicity, represented in Equations 6.11, 6.12, and 6.13. The critical load for the structure is the minimum value for *N* which leads to the determinate of *M* being zero.

$$M_{ij} = B_{ij} + C_{ij} + D_{ij} \qquad M \in \mathbb{R}^{2^c-1 \times 2^c-1} \tag{6.10}$$

$$B_{ij} = \begin{cases} -N, & j = i+1 \\ 0, & j \neq i+1 \end{cases} \tag{6.11}$$

$$C_{ij} = \begin{cases} (j-i-1) l_c K_i, & j \leq i \\ 0, & j > i \end{cases} \tag{6.12}$$

$$D_{ij} = \begin{cases} (j+1)N, & i = 1 \\ 0, & i \neq 1 \end{cases} \tag{6.13}$$

For complexities greater than two, there is no absolute solution for critical load; therefore, to proceed with the design, one of the possible solutions is selected. Then, at the end of the calculations, the selected solution is checked to determine if any of the other solutions are less than the value of the selected one. This process is then repeated until the correct critical load is found.

After a critical load is selected, the equivalent spring constants can be applied based on a linear elastic behavior model for the cables, shown in Equation 6.14. Based on the geometry of the design as defined in Figure 6.3, the cable lengths can be represented as seen in Equation 6.15. Combining Equations 6.14 and 6.15 leads to Equation 6.16—a representation dependent on the geometry of the design and material properties. For these instances and those prior, the section lengths can be denoted as seen in Equation 6.17.

$$k_i = \frac{E_s A_{s,i}}{s_i} \tag{6.14}$$

$$s_i = \frac{l_i}{\cos(\alpha_i)} \tag{6.15}$$

$$k_i = \frac{E_s A_{s,i} \cos(\alpha_i)}{l_i} \tag{6.16}$$

$$l_i = \frac{l_0}{2^i} \tag{6.17}$$

From here, Equation 6.16 can be substituted into the previously found critical load and setting said critical load equal to the total compressive load on the structure. The resulting formula can then be solved for the cable areas and the areas subsequently minimized to result in a minimal mass solution. As such, there are two definitions for mass of the cables in the structure, the maximum of which must be selected in order to avoid failure of the structure prior to the prescribed load. By combining Equations 6.2, 6.3, and 6.15 in addition to the spring constant based representation for cable area, the cable mass can be represented as denoted in Equation 6.18.

$$m_{s,i} = \max\left\{\frac{f_0 l_i \rho_s}{2\cos^2(\alpha_i)\,\sigma_s}, \frac{A_{s,i} l_i \rho_s}{\cos(\alpha_i)}\right\} \tag{6.18}$$

The final mass of the structure can be represented as in Equation 6.19, as the sum of the masses of the beams and the masses of the cables. In total, the mass of the beams in the structure can be represented using Equation 6.20, which is the sum of the required mass for each beam in the structure as determined by the load supported by the beam and the beam's length, as well as material properties. Similarly, the total mass for the structure's cables is represented by Equation

6.21, equivalent to the sum of the cables' masses as determined by the tensile load requirement and the spring constant driven requirement, as well as their respective lengths, angles, and material properties.

$$m_{total} = m_{b,total} + m_{s,total} \tag{6.19}$$

$$\begin{aligned} m_{b,total} &= 2^c \sqrt{f_0} l_c^2 \frac{2\rho_b}{\sqrt{\pi E_b}} + 4 \sum_{i=1}^{c} 2^{i-1} \sqrt{f_{v,i}} l_{v,i}^2 \frac{2\rho_b}{\sqrt{\pi E_b}} \\ &= \sqrt{f_0} l_0^2 \frac{2\rho_b}{\sqrt{\pi E_b}} \left[\frac{1}{2^c} + 4 \sum_{i=1}^{c} \frac{\tan^{5/2}(\alpha_i)}{2^{i+1}} \right] \end{aligned} \tag{6.20}$$

$$\begin{aligned} m_{s,total} &= 8 \left[\sum_{i=1}^{c} 2^{i-1} \max \left\{ \frac{f_0 l_i \rho_s}{2\cos^2(\alpha_i)\,\sigma_s}, \frac{A_{s,i} l_i \rho_s}{\cos(\alpha_i)} \right\} \right] + 4 \left[\sum_{i=1}^{c} 2^{i-1} \frac{A_{s,i} l_i \rho_s}{\cos(\alpha_i)} \right] \\ &= 2 l_0 \rho_s \sum_{i=1}^{c} \sec(\alpha_i) \left[A_{s,i} + 2 \max \left\{ \frac{f_0}{2\cos(\alpha_i)\,\sigma_s}, A_{s,i} \right\} \right] \end{aligned} \tag{6.21}$$

A minimal mass structure for supporting the prescribed load with a given length can be obtained by minimizing Equation 6.19 with respect to the applicable angles α_i. There may be physical constraints to be considered for this minimization, such as intersection of the cables, but these may be dealt with by a rotation of the supporting beams depending on angle. Alternatively, physical constraints may be dealt with by imposing an additional constraint on the angles to avoid intersection.

This design approach greatly benefits the evacuated airship design. However, more reduction in mass is still necessary for the design to be plausible. Moreover, the mass of the membrane for the airship is still neglected, which further necessitates greater reduction in structural mass.

Thin Walled Cylindrical Beams

A thin walled cylindrical beam cross section is far more efficient in buckling than a solid circular cross section for a given area due to the increased second moment of area, and consequently increased stiffness. Therefore, the mass of the beam can be far less than that of a solid beam for a given loading and length. However, introducing a thin walled cylindrical cross section necessitates local buckling analysis to account for any possible buckling of the beam walls.

The second moment of area for an annular (ring) cross section can be seen in Equation 7.1. For simplification, the ratio between the inner and outer diameter of the ring is defined as shown in Equation 7.2. Applying the definition from Equation 7.2 to Equation 7.1 results in Equation 7.3, providing better insight into the properties of the annular cross section over a solid circular cross section. Similarly, the area of the annular cross section can be represented with Equation 7.4.

$$I = \frac{\pi}{4}(r_2^4 - r_1^4) \tag{7.1}$$

$$r = r_2 = \frac{r_1}{\psi} \tag{7.2}$$

$$I = \frac{\pi}{4} r^4 (1 - \psi^4) \tag{7.3}$$

$$A = \pi r^2 (1 - \psi^2) \tag{7.4}$$

The benefits of an annular cross section can be seen by observing that a reduction in cross sectional area by half can be achieved while the second moment of area is only reduced by one quarter. For the previous example, the ratio of the radii is the reciprocal of root two. Additionally, the relation between outer radii can be obtained by setting the area of an annular cross section equal to the area of a solid circular cross section, which yields Equation 7.5. Using this relation, the improvement in second moment of area for equivalent area cross sections can be observed in Figure 7.1 and the accompanying Equation 7.6.

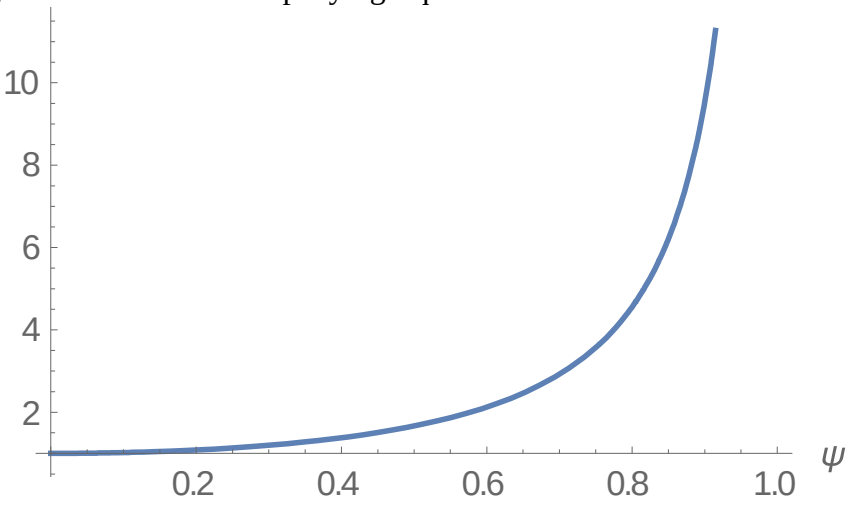


Figure 7.1

$$r_s = r_a \sqrt{1 - \psi^2} \tag{7.5}$$

$$\frac{I_a}{I_s} = \frac{r_a^4 (1 - \psi^4)}{r_s^4} = \frac{1 + \psi^2}{1 - \psi^2} \tag{7.6}$$

Implementation of annular cross section beams in the previously discussed tensegrity design is fairly trivial. Substituting Equation 7.3 into the buckling equation (5.6) and solving for the cross sectional area of the beam as defined by Equation 7.4 results in Equation 7.7, which gives the minimal mass of the beam for a given length and load. This mass formula can be substituted back into the same method discussed above in place of Equation 6.1.

$$m = \rho \cdot A \cdot l = \frac{2\rho(1-\psi^2)}{\sqrt{E\pi(1-\psi^4)}} l^2\sqrt{f} \tag{7.7}$$

Application of this cross section provides even further mass saving capability for the vacuum airship. However, as previously stated, this cross section warrants further analysis because of the additional buckling mode possibilities which accompany the hollow beam. Consideration for these new buckling modes is addressed below.

Applicable Buckling Modes and Analysis

Buckling, or elastic instability, is the primary source of failure for this design because of the pressure and mass constraints discussed previously. Due to the nature of the vacuum airship concept, the structural loads being supported are mostly compressive, which tends to lead to an elastic instability mode of failure. The types of elastic instability encountered with this design are global buckling, local beam buckling, and local buckling of the wall. Both the global buckling of the tensegrity beams and the local beam buckling have been addressed in the above sections. As such, the local buckling of the hollow beam walls is the only instability mode to be examined.

Analysis for the stability of the beam walls was conducted using the approach from Bruhn[1]. The approach detailed therein is based on a small-deflection theoretical solution to an eighth-order differential equation from Donnell[3], which is shown here in Equation 8.1 for reference.

$$\frac{Et^3}{12(1-\nu^2)}\nabla^8 w \frac{Et}{r^2}\frac{\partial^4 w}{\partial x^4} + 2s \cdot t \ \nabla^4\left(\frac{\partial^2 w}{\partial x \partial s}\right) = 0 \tag{8.1}$$

There are two possible representations of buckling for the beams in the vacuum airship, depending on the geometry of the beam. The distinction between beam classifications is shown in Inequality 8.2. Beams with geometries which satisfy Inequality 8.2 have their buckling load defined by Equation 8.3 and are considered, by Bruhn[1], as long cylinders. If a beam does not satisfy Inequality 8.2, the beam is likewise considered a transition cylinder and the buckling load of the beam can be represented by Equation 8.4.

$$\begin{cases} \dfrac{l^2}{r \cdot t} > 100 \quad \Rightarrow \quad Long\ Cylinder \\ \dfrac{l^2}{r \cdot t} \leq 100 \quad \Rightarrow \quad Transition\ Cylinder \end{cases} \tag{8.2}$$

$$f = \frac{E \cdot t}{r\sqrt{3(1-\nu^2)}} \tag{8.3}$$

$$f = \frac{E\pi^2 t^2}{12\ l^2(1-\nu^2)}\left[1 + \frac{12\ l^4(1-\nu^2)}{\pi^4 r^2 t^2}\right] \tag{8.4}$$

As long as the force on each of the beams in the design does not exceed the buckling load as defined by Equation 8.3 or 8.4, then the structure is not in danger of failure by local buckling of the walls. If the design load does exceed the determined buckling load, however, the thickness of the walls must be increased, or other constraints imposed on the design to ensure no local buckling occurs in the beam walls. Implementation of the entire theoretical design, as a compilation of all the aforementioned analysis, is implemented using Mathematica. Said code used to generate the tensegrity beams is included in Appendix 1.

Theoretical Results

The following results were obtained using the theoretical methods described above. These results use general material properties of carbon fiber composites currently available. Carbon fiber composite was chosen for the material's low density and high rigidity in order to keep mass to a minimum while supplying as much support against buckling as possible.

Considering both the tensegrity design as a whole as well as the implementation of tensegrity based beams, Figure 9.1 shows the mass reduction factor associated with each level of tensegrity complexity at various beam lengths. From this graph, one can see the minimum mass reduction from implementation of a tensegrity beam is independent of the length of the beam span. Therefore, the beam length will not limit the possible mass reduction from implementation of the tensegrity beam method. Additionally, from Figure 9.1, the trend for mass savings in relation to the complexity of the tensegrity beam starts to become apparent. The aforementioned trend can be empirically estimated as slightly greater than half of the previous complexity's mass. This trend is further reinforced when taking into consideration a beam of complexity zero would have a mass reduction value of one, since said beam would be the standard beam to which the other tensegrity structures are compared.

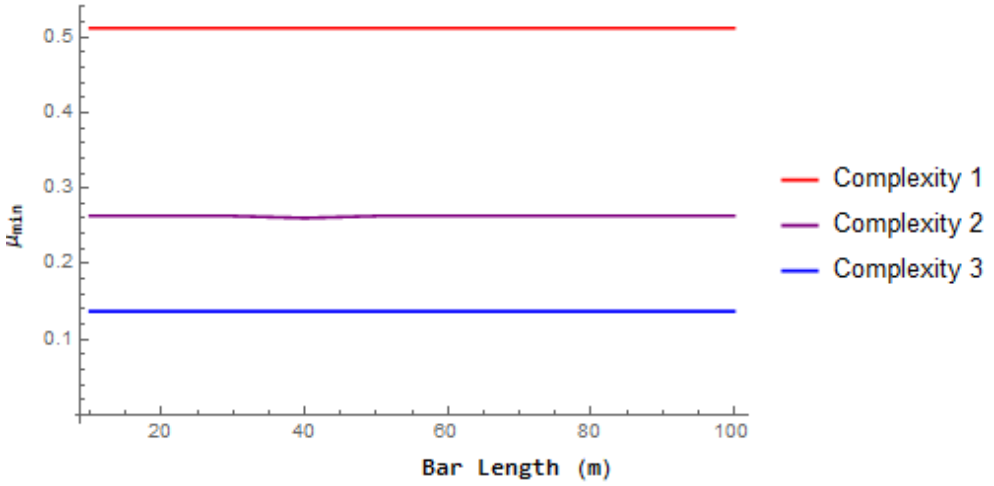


Figure 9.1

Results from implementing the second mass reduction method, using a hollow tube in place of a solid cylindrical beam, can be seen for a tensegrity beam of complexity three in Figure 9.2. As was the case with the previous graph, this graph shows see the mass reduction from use of hollow beams in the tensegrity design is independent of the length of the beam span. Similar to the complexity, the beam length will not limit the possible mass reduction from using a thin-walled cylindrical cross section. Instead, the mass savings will be constrained by the local wall buckling as well as manufacturing constraints on the minimum thickness of the tube walls.

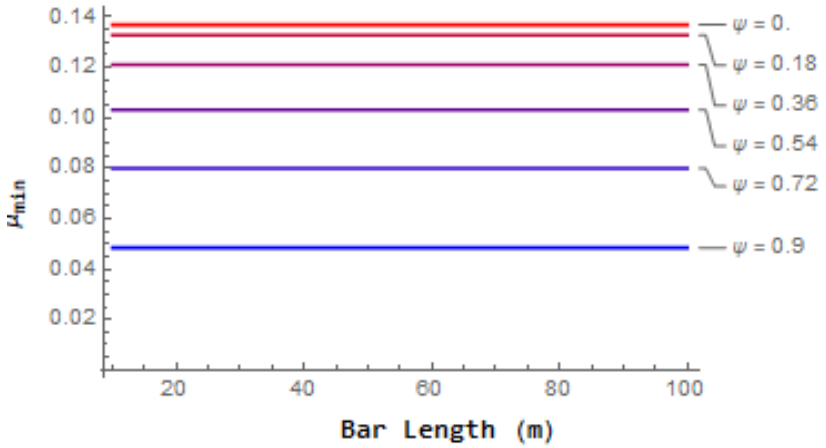


Figure 9.2

Since the mass reduction due to both the wall thickness and the complexity of the tensegrity structure have been shown to be independent of the beam span, the mass reduction can be viewed without consideration to the beam span. The analysis for mass reduction in relation to proportional wall thickness for three levels of complexity is shown in Figure 9.3 for an eighty meter beam span. Empirically, the weight reduction accomplished by the change in cross section is approximately the same, proportionally for each complexity. As such, the mass reduction is compounded by the implementation of both methods. Additionally, this graph shows the associated benefits of reducing the wall thickness increase as the ratio of radii approaches unity. However, physical constraints from minimum thickness of the material begin to play a part as the ratio approaches unity.

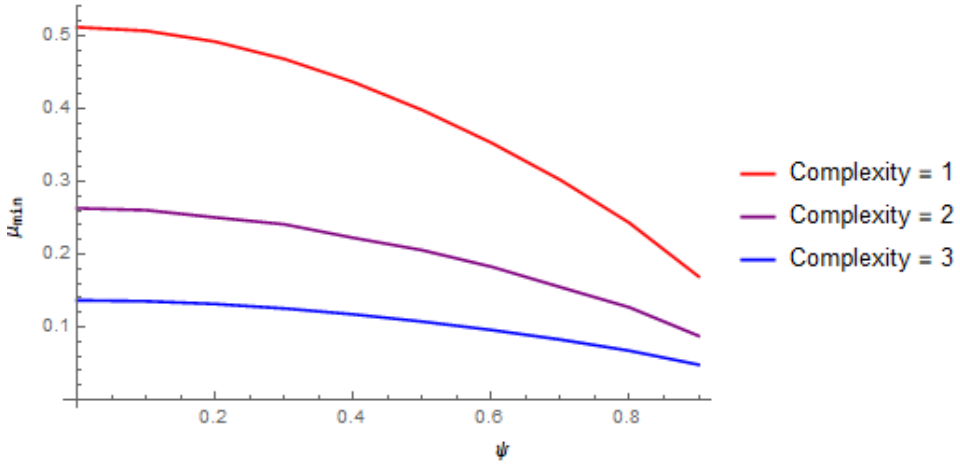


Figure 9.3

Assigning the ratio of radii to be 0.99, the resulting mass reductions can be seen for complexities one through three in Figure 9.4. The ratio of radii was chosen to be 0.99 because this is approximately the closest to unity the ratio can be set, for a structure of complexity three, while still being unaffected by local wall buckling at all levels, as well as being within manufacturing constraints of the material thickness for all members. Similar results can be seen in this graph as in Figure 9.1. However, the mass ratio represented in this graph is close to a

tenth of that shown in Figure 9.1, which emphasizes the benefits of implementing both mass reducing methods.

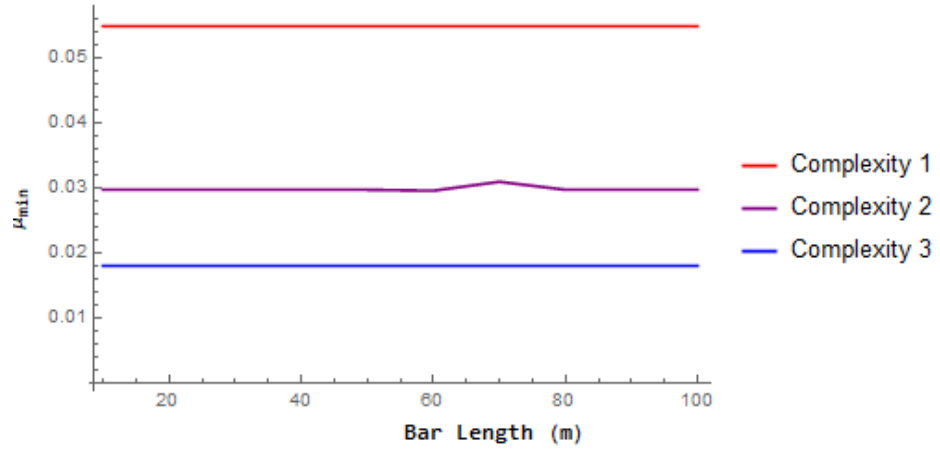


Figure 9.4

In order to observe the theoretical effect of scale on the lift provided by the airship, the skinless payload of the airship is defined as the lift the vacuum airship would provide if the vacuum could be held without a membrane, or with an infinitely thin membrane. The results of this measure can be seen in Figure 9.5. Here, the benefits of the airship's scale become apparent; excess lift provided by the airship increases as the scale of the airship increases for the higher complexity designs. These results can also be interpreted as the mass available to devote to the membrane for a neutral buoyancy or feasibility threshold case.

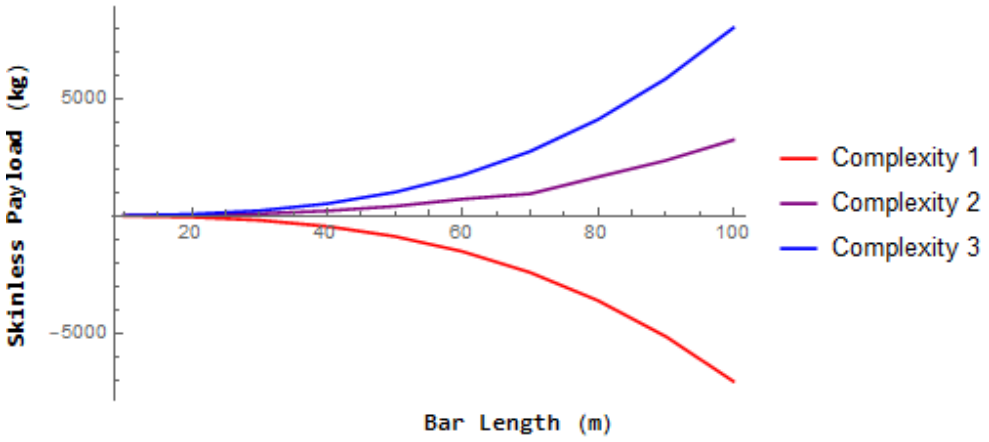


Figure 9.5

For the more interesting case where the evacuated airship has payload capacity, the available mass for the membrane is of note. The graph shown in Figure 9.6 shows the maximum thickness available for the membrane while providing enough surplus lift to transport five hundred kilograms of payload. Notably, the thickness for a complexity three beam structure of eighty meter span is within the physical manufacturing constraints of the carbon fiber composite, which marks the point where all physical constraints on the evacuated airship design have been satisfied theoretically. More detail will be given to the design of the membrane in later sections from a computational standpoint.

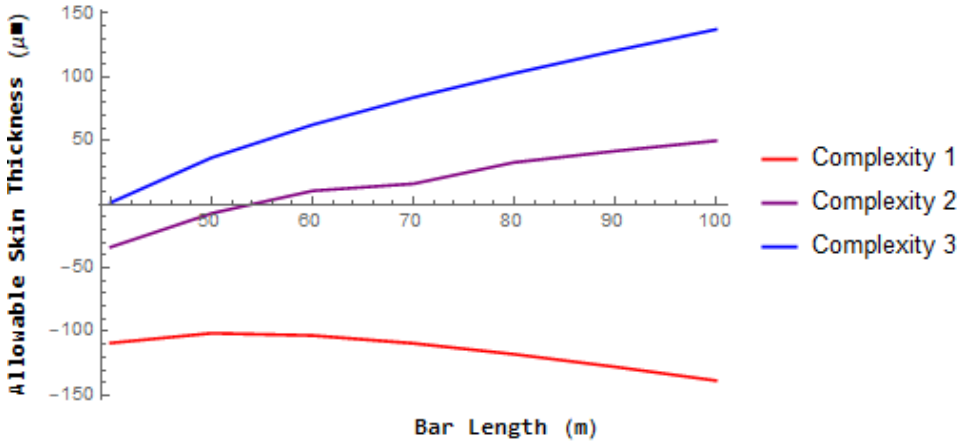


Figure 9.6

From the theoretical analysis of the evacuated airship design as a tensegrity structure as described, the concept is feasible given current materials and implementing the design methods detailed herein. This design, when using a bar length of 80 meters, which corresponds to a rough diameter of 100 meters, results in a total displaced mass of 7882 kilograms. The internal structure in this case has a mass of 3771.5 kilograms. Using FEA software, as described below, a membrane with a mass of 1600 kilograms was attached to the design and simulated under atmospheric loading on Mars. This simulation resulted in a loss in volume of the displaced atmosphere roughly equating to 1493 kilograms. This leaves approximately 500 kilograms for payload of the evacuated airship on site. If the trends seen in Figures 9.1 and 9.4 continue, then by extension of the tensegrity beam design to complexity 4, the internal structure mass should be reduced by a little less than half. Therefore, accounting for error and further mass distribution in the membrane, this results in an expected payload between 1 and 1.5 tons.

Unlike the previously described designs utilizing a monocoque shell, this tensegrity approach meets elastic stability constraints on all levels of the design as well as meeting manufacturing constraints based on the material. As such, this analysis has shown this design to be feasible in theory. Moving forward, the theory will be verified using detailed, non-linear finite element simulations, accounting for non-linearities in displacement, global and local buckling, and membrane failure criteria.

Additional Discussion

The mathematical theory of the tensegrity based design for the evacuated airship, though interesting, does not fully encapsulate the concept behind the design. In order to better describe the tensegrity structure based beam's design, three-dimensional renderings for beams of complexity one through three are shown in Figure 10.1. These renderings are produced from the theoretical calculations described prior, with dimensions which correspond to the minimal mass solutions to the theoretical model. In future research, higher complexity beam designs may be explored within the manufacturing constraints of the materials, and plausibly provide even greater mass reduction, leading to better payload capacity.

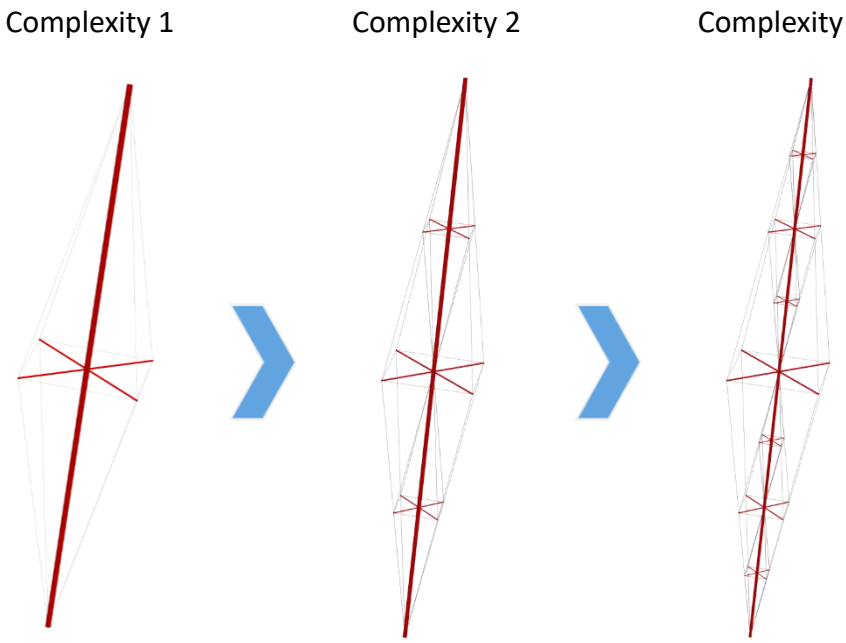


Figure 10.1

The entirety of the structure for this design of the evacuated airship, as a combination of a twelve beam tensegrity design utilizing tensegrity based beams, is fairly complex. Said design can be seen as a three-dimensional rendering in Figure 10.2. Due to the complexity of the design, and the resulting crowded nature of the full rendering, one may find using Figure 10.3 to be helpful. Figure 10.3 represents the twelve beam tensegrity design; the final design is then obtained by replacing each of these twelve beams with the tensegrity structure of complexity three seen in Figure 10.1. This may be a better way to understand the concept instead of interpreting Figure 10.2 directly.

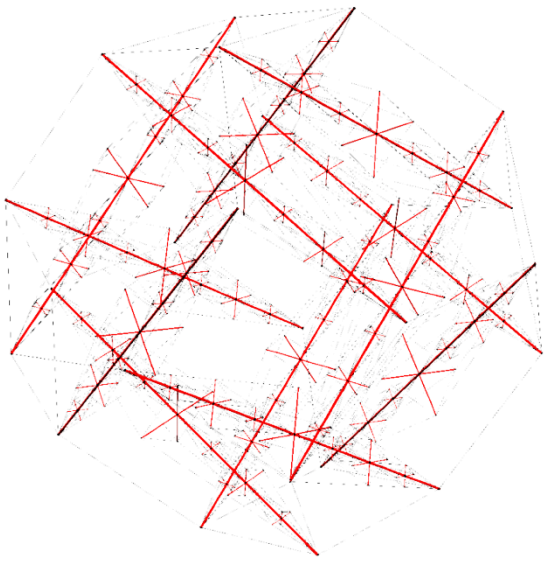


Figure 10.2

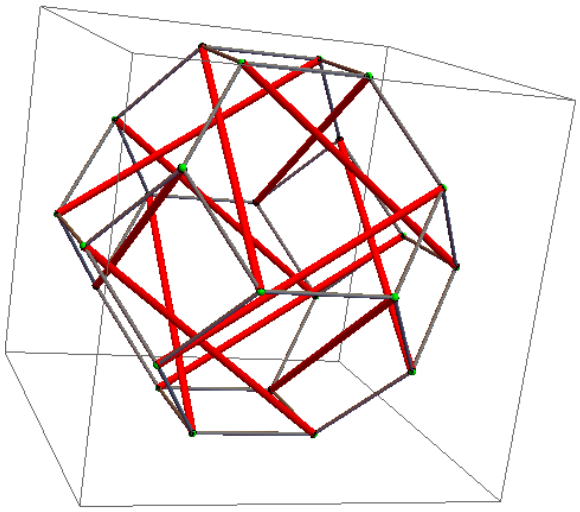


Figure 10.3

Even though the biggest impedance to the vacuum airship is the structural viability of the design, there are other factors to consider for a successful mission. The highest priority mission factors for the evacuated airship will be discussed further below. Additionally, the analysis and considerations for those factors as conducted in this research are entailed below.

Mission Considerations

There are several factors to consider when looking at the evacuated airship in a mission context; the foremost of these is the journey of the evacuated airship to Mars. This task has two major aspects: first, transport of the evacuated airship, and second, the planetary insertion of the vehicle. Transport of the vehicle is highly non-trivial because the design is on the order of a hundred meters in diameter, which is vastly too large to be contained within a launch vehicle fairing. The vehicle is also too large to be assembled in Earth orbit, then sent to Mars because this would have an exceedingly high probability of damage to the vehicle en route. Planetary insertion, or entry into the atmosphere, of the vehicle is necessary because the goal of the mission is to have the evacuated airship perform tasks in the Martian atmosphere and at the surface of Mars.

Deployment

As stated above, the design of the evacuated airship is exceedingly large in order to fully take advantage of the mission capabilities of the vehicle. Consequently, the airship will need to be transported in a smaller form, which drives a collapsible and deployable design. Fortunately, the tensegrity design lends itself well to this task.

Beside the mass reduction provided by the tensegrity structure, additional benefit from the tensegrity structure design of the beams is their aptitude for implementation in a deployable manner. Since the joints of the tensegrity structure do not require any geometric constraints for the structural stability of the system to be maintained, the entire structure can be folded in on itself by disconnecting a few of the cables in the design. The deployment of a complexity three tensegrity design is shown in Figure 11.1, with the structure being deployed to the full eighty meter design from a ten meter transportation configuration.

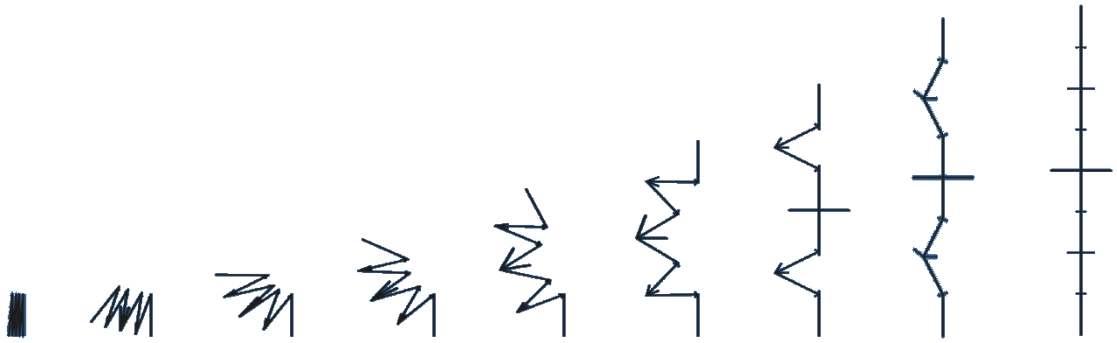


Figure 11.1

Planetary Insertion

The atmospheric entry of the evacuated airship was analyzed using a series of differential equations governing the dynamics and thermodynamics of the system. These differential equations are base off the analysis methods detailed by the FAA in reference 4 and Atmospheric Entry by Griffin[5]. As such, the dynamics of the airship during entry are governed by the effects of gravitational, drag, and buoyant forces, shown in Equation 11.1. The thermodynamics of the airship are governed by the energy exchange between the detached shock, through the boundary layer, and the vehicle as well as the blackbody radiative effects. This energy exchange can be modeled as shown in Equation 11.2.

$$m\vec{a} = \sum \vec{F} = \underbrace{\frac{-GMm}{r^2} \cdot \frac{\vec{r}}{|\vec{r}|}}_{gravitational} - \underbrace{\frac{\rho}{2} V^2 C_d \cdot A \frac{\vec{V}}{|\vec{V}|}}_{drag} + \underbrace{\rho \cdot Vol \cdot \frac{-GMm}{r^2} \cdot \frac{\vec{r}}{|\vec{r}|}}_{buoyant} \tag{11.1}$$

$$\dot{Q} = \underbrace{\frac{\rho \cdot V^3 \cdot A}{5}}_{\substack{shock\ energy \\ exchange}} - \underbrace{\sigma \cdot \varepsilon \cdot T^4}_{\substack{blackbody \\ radiation}} \tag{11.2}$$

From the atmospheric model for Mars in reference 4, Equation 11.3 represents the atmospheric density as a function of distance from the planet's surface. Subsequently, defining the entry in a Mars-centered, Cartesian reference frame leads to the definitions for terms shown in Equations 11.4, 11.5, and 11.6. These equations can be used in conjunction with those above to create a viable system of differential equations governing both the motion and the thermodynamics of the airship during entry.

$$\rho = \rho_s e^{-\beta \cdot h} \tag{11.3}$$

$$r = |\vec{r}| = \sqrt{x^2 + y^2} \tag{11.4}$$

$$h = |\vec{r}| - R = \sqrt{x^2 + y^2} - R \tag{11.5}$$

$$V = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{11.6}$$

Using the above equations, the atmospheric entry was modeled with the system of differential equations shown in Equations 11.7, 11.8, and 11.9. Since this system has two second order differential equations and one first order differential equation, five initial conditions are needed to solve the system. These initial conditions can be seen in Equations 11.10 as a function of initial distance from the surface of Mars, the true anomaly, velocity, and orbital flight path angle. For the purposes of this analysis, the true anomaly can be se to zero because of the radial symmetry of the problem, therefore the true anomaly will not have any bearing over the results. Additionally, the orbital flight path angle can be set to zero because the entirety of the design space can be covered with the other two parameters. The initial temperature of the airship was chosen to be fifty Kelvin as an estimation of the temperature of the airship components after the trip to Mars; if need be, additional cooling measures can be taken to further buffer the airship against heat prior to arrival.

$$\ddot{x} = \left[\rho_s e^{-\beta\left(\sqrt{x^2+y^2}-R\right)} \cdot Vol - 1\right] \cdot \frac{GM \cdot x}{(x^2+y^2)^{3/2}} - \rho_s e^{-\beta\left(\sqrt{x^2+y^2}-R\right)} \sqrt{\dot{x}^2 + \dot{y}^2} \cdot \frac{\dot{x} \cdot C_d \cdot A}{m} \tag{11.7}$$

$$\ddot{y} = \left[\rho_s e^{-\beta\left(\sqrt{x^2+y^2}-R\right)} \cdot Vol - 1\right] \cdot \frac{GM \cdot y}{(x^2+y^2)^{3/2}} - \rho_s e^{-\beta\left(\sqrt{x^2+y^2}-R\right)} \sqrt{\dot{x}^2 + \dot{y}^2} \cdot \frac{\dot{y} \cdot C_d \cdot A}{m} \tag{11.8}$$

$$\dot{T} = \left[\frac{\rho_s e^{-\beta\left(\sqrt{x^2+y^2}-R\right)} \cdot (x^2+y^2)^{3/2} \cdot A}{5} - \sigma \cdot \varepsilon \cdot T^4\right]\frac{1}{C_p m} \tag{11.9}$$

$$x(0) = (h_0 + R)\cos(\theta) \;\; ; \;\; y(0) = (h_0 + R)\sin(\theta) \;\; ;$$
$$\dot{x}(0) = V_0 \cos(\gamma) \;\; ; \;\; \dot{y}(0) = V_0 \cos(\gamma) \;\; ; \;\; T(0) = T_0 \tag{11.10}$$

This analysis assumes the various components of the airship have been deployed from the launch vehicle fairing and are ready for the assembly process to begin. The assembly process does not take place during travel to Mars because doing so would increase the likelihood of damage from debris in transit. Therefore, the assembly process occurs during the time period between deployment and entry into the atmosphere. This can be seen illustrated in Figure 11.1.

Results from the analysis can be seen in Figure 11.2 for various initial heights with initial velocities corresponding to circular orbit velocity at that height. This figure shows the trajectories associated with the different heights with respect to Mars. These simulations have been run until the lift of the evacuated airship fully takes over, which in the mission is the point where operations can begin. Furthermore, other important properties of the entry can be seen in Figure 11.3, which gives the trajectory in addition to the velocity, temperature, and acceleration profiles with time.

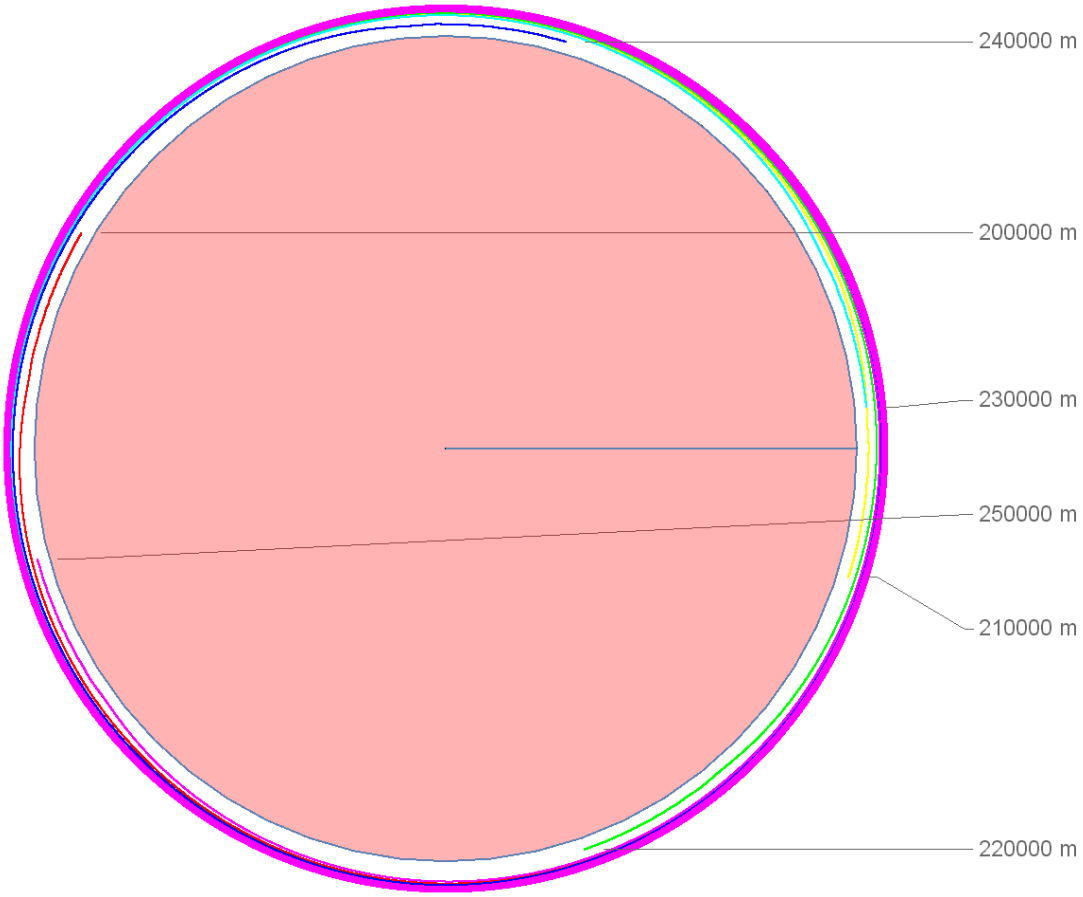


Figure 11.2

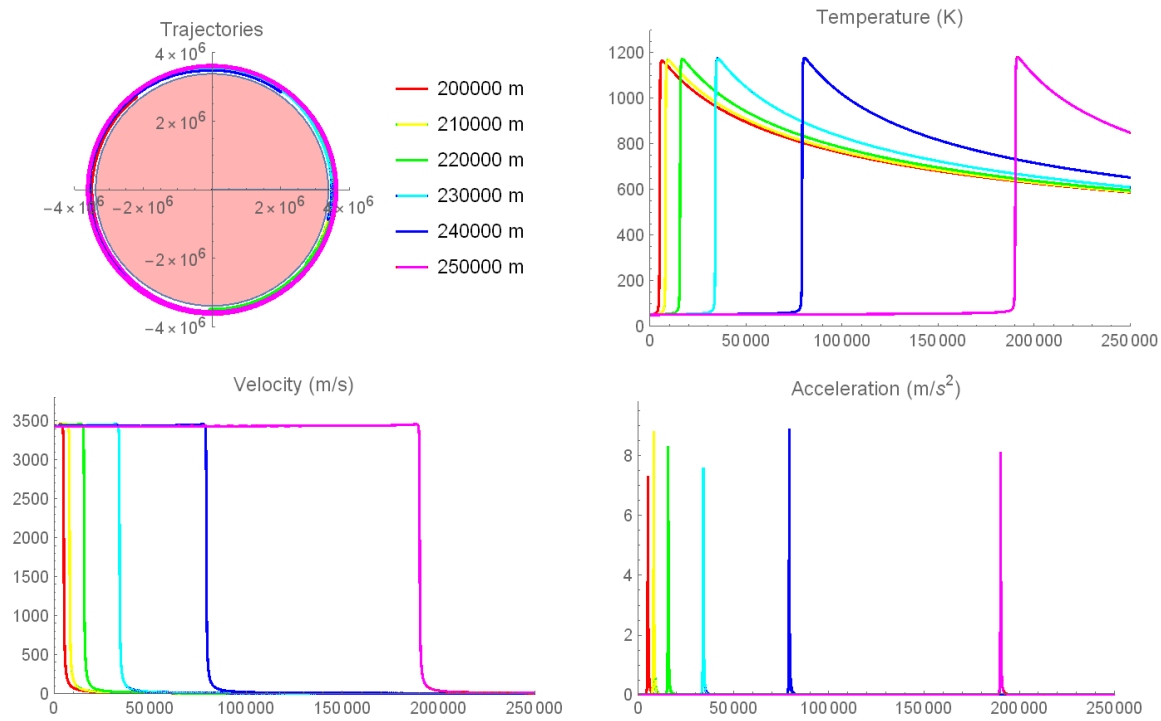


Figure 11.3

In addition to the data shown in Figure 11.3, the maximum values for the important aspects of the entry can be observed as they relate to the initial altitude. Said aspects being the time available for assembly of the airship from the deployed state and the maximum temperature experienced by the airship. These values are plotted in Figure 11.4 for a larger sampling of initial altitudes than shown in Figure 11.3, so as to result in more accurate interpolation. The variance observed in the maximum temperature depends on the point in the vehicle's orbit at which the atmosphere is first significantly encountered, which explains the sinusoidal nature of the variance. These figures show that the assembly time and the maximum temperature both decrease as the orbit is initiated with lower altitudes, however, the temperature decreases only slightly, whereas the assembly time has an exponential behavior.

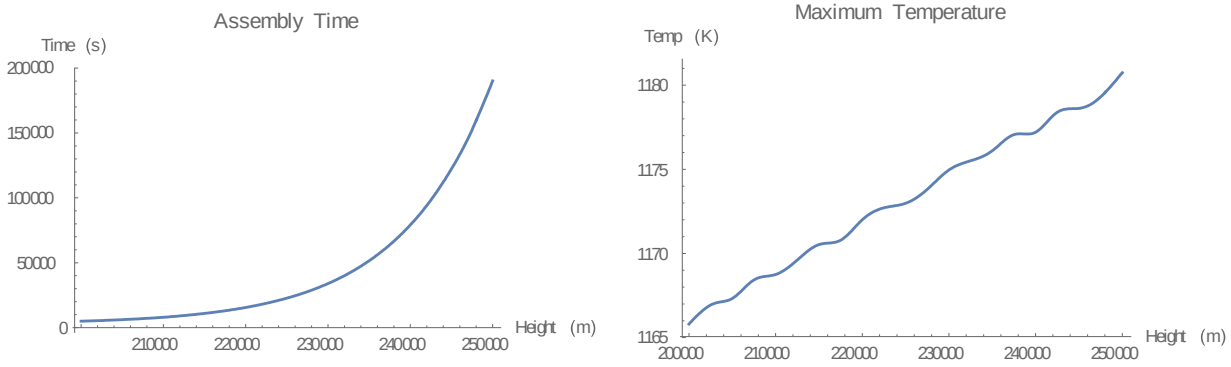


Figure 11.4

The analysis was also conducted for various perturbations of the initial velocity: the other design parameter available. Results for these simulations can be seen in Figure 11.5, which

shows the trajectory along with temperature, velocity, and acceleration profiles for each case. Again, the additional data associated with a larger sampling is seen in Figure 11.6. These figures show that the maximum temperature decreases much more significantly with a decrease in velocity from circular than that seen with change in the initial altitude. However, the same exponential trend in assembly time is observable for these cases as in the altitude cases.

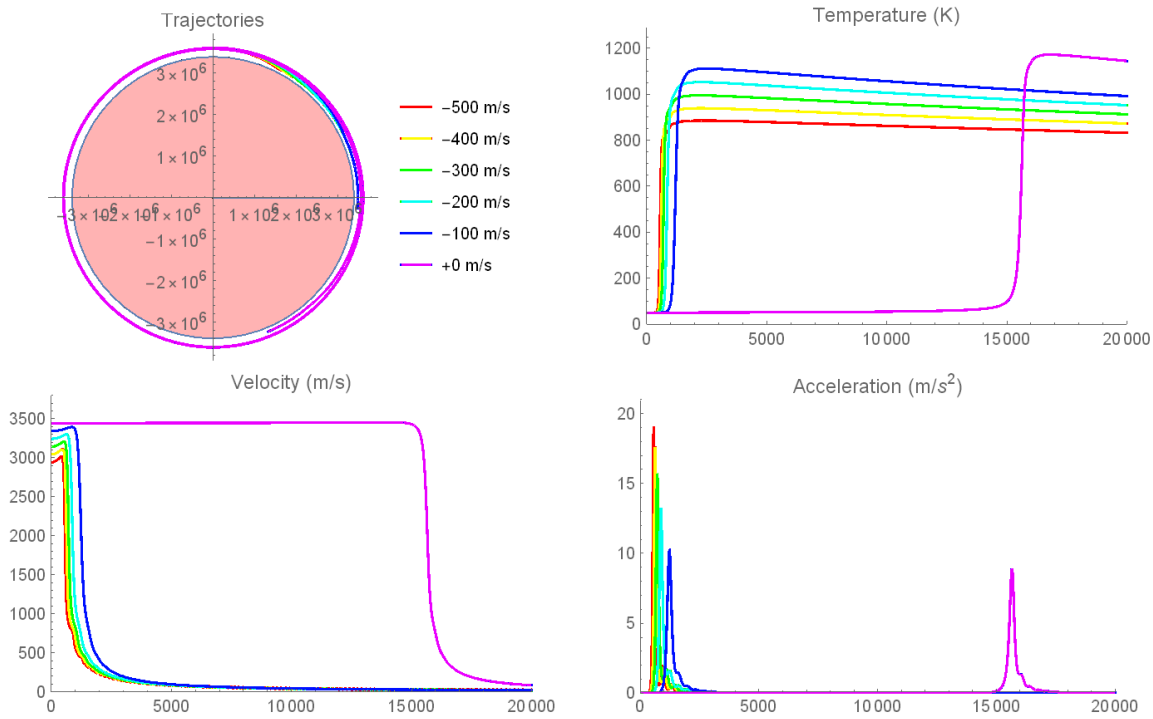


Figure 11.5

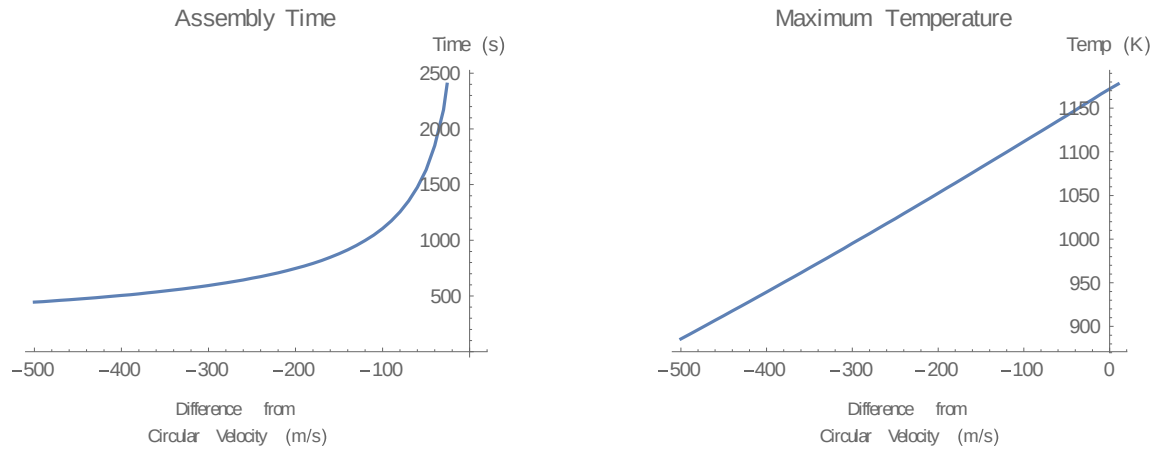


Figure 11.6

Of note are the temperatures experienced by the vehicle for both sets of cases above, for both of these simulations, the temperature is five hundred or more Kelvin lower than that of the orbiter shuttle upon Earth re-entry. However, having an even lower temperature would still be desirable. Since having a low temperature of the airship is desirable and having a relatively long time for assembly is also desirable, combinations of decreased velocity with increased height

were tested to observe the results of combining the two. As such, results from setting the difference in velocity to 200 meters per second off circular are shown in Figure 11.7, which has some fairly interesting properties observed in the acceleration profile. The peak acceleration for this simulation has a maximum for 600,000 meters initial altitude, both lower and higher than that, the acceleration has less of a peak value. However, this is far more acceleration in general than the cases seen with nominal initial circular orbital velocity. Therefore, the maximum acceleration, maximum temperature, and assembly time are all of interest for the analysis, which for this case are plotted in Figure 11.8.

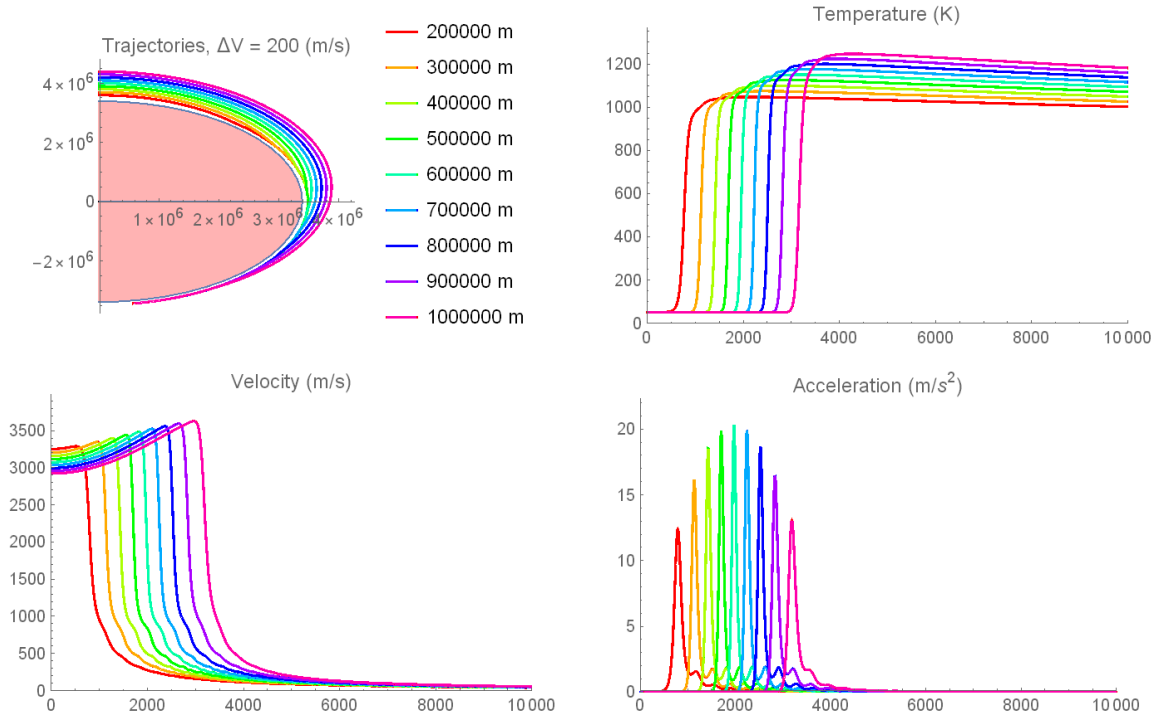


Figure 11.7

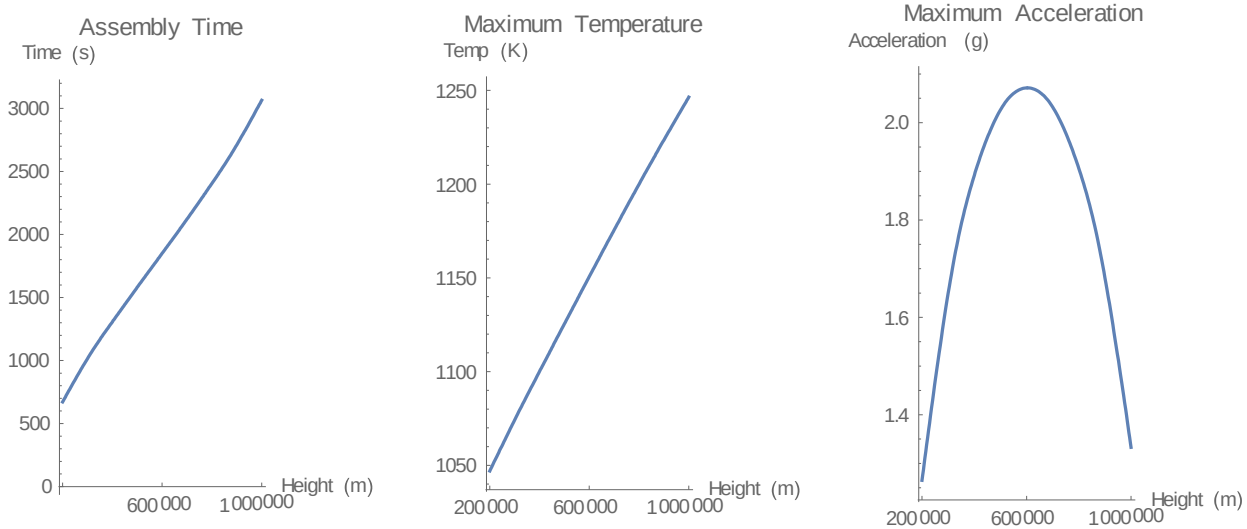


Figure 11.8

These results encourage further study of the relation between the design variables, initial altitude and initial velocity, and the assembly time, maximum temperature and acceleration. Consequently, simulations were done for various values of the altitude and velocity to obtain a full view of the design space as laid out by the three resulting values given above. The results of this analysis are shown in Figure 11.9, which gives a contour plot for each of the values of interest.

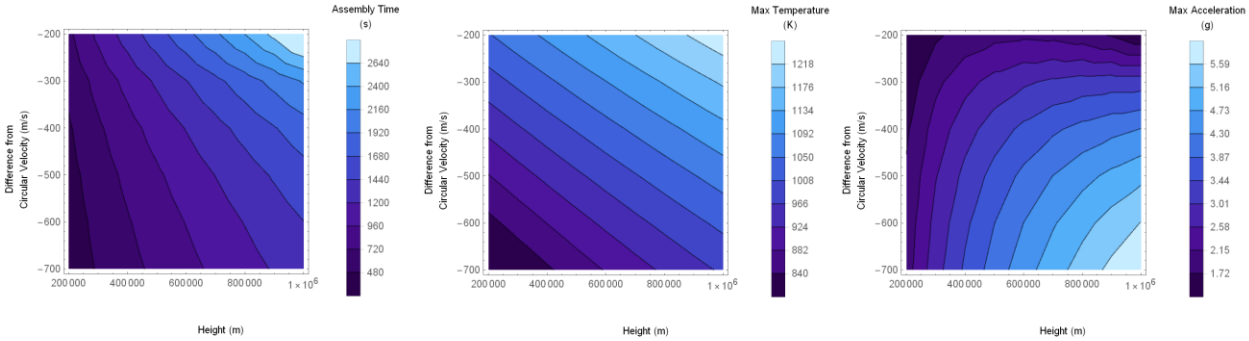


Figure 11.9

Simulations were also done for cases with 500 meters per second off circular velocity. Results for these simulations are shown in Figure 11.10, these cases can be seen to keep a relatively preferable balance between the resulting values of interest. However the final designed entry plan will be dependent on constraints imposed by the final design.

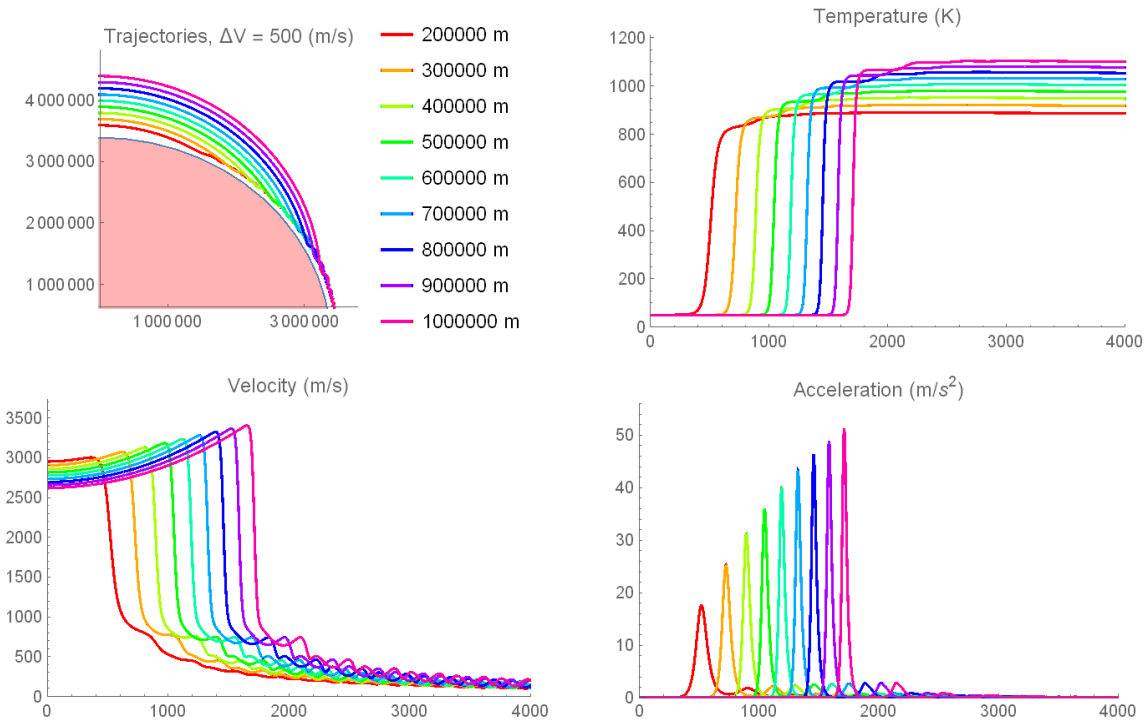


Figure 11.10

In summary, this analysis shows that using a decaying orbit for the evacuated airship's entry into the atmosphere provides significant benefits and may be a viable option for the mission architecture of the evacuated airship on Mars. Final values for the planetary insertion plan will need to be determined further down the line to conform to constraints based on the final material and structure of the airship. Therefore, this analysis should be viewed as preliminary and will be used and modified as the evacuated airship's design progresses. As such, the final trajectories will be based of meeting the three criteria of assembly time needed, maximum temperature, and maximum acceleration.

FEA Modeling and Verification

In order to verify the structural theory underlying the tensegrity based design of the evacuated airship, Finite Element Analysis software was used to simulate the structure and loads. The software chosen for this simulation was ABAQUS, which is a high fidelity numerical solver for the finite element system. Creation of the evacuated airship design for analysis by the program involved the production of a separate program written to be compatible with the Mathematica code responsible for optimizing the design. This code converts the design from pure mathematical values to a collection of nodes and the connectivity between nodes to form the final design, along with the properties of each member. Said program then converts the collection of nodes and the connectivity of the structure into an input file suitable for ABAQUS to interpret based on the programming conventions of the FEA software, e.g. using the ABAQUS keywords to define each of the members of the total structure.

Conversion of the theoretical design metrics to a truss-like form relies on two steps. The first step is to create a mathematic spatial description of the macro tensegrity structure, which can and was accomplished using the truncated octahedron simplification of the tensegrity design model. Next, a nodal description of the tensegrity based beam is created using the metrics from the theoretical design, e.g. the dimensions and geometric relations between each of the beam and cable members. These two components are then combined by translating the node placement of the tensegrity beams to replace the standard beams within the macro tensegrity structure using a series of translations and rotations. This process yields a full three dimensional description of the airship's internal structure which can then be converted into an FEA acceptable format.

In addition, the program developed to convert the theoretical design into a viable FEA model defines the connections between each of the members of the structure as pin joints (as dictated by the tensegrity beam theory). This is accomplished by constraining the positions of all coincident nodes in the design with a series of equations fixing their position relative to each other. Also dictated by the tensegrity beam theory is the inclusion of pre-stresses in the cable members, which is also taken care of by the conversion program. Pre-stressing the cables utilizes the initial conditions options within ABAQUS, and in doing so prescribes the required stress to each cable according to need. As such, the cable members never experience anything other than tension, and so need not be modeled for any buckling effects.

As stated before, the properties of each member are included in the creation of the input file. This includes the material properties, which are written into the input file in an acceptable format for the FEA solver, as well as the geometric properties of the member, such as the radius or cross sectional area of the member. All of these considerations are taken care of by the conversion program, which allows for rapid change of the structural design of the airship in contrast to the gargantuan task of creating the design from within an FEA solver. The code for this conversion process can be seen in Appendix 2.

Once the structural model is loaded into ABAQUS, the simulation can be created and evaluated. Obviously, the model is first constrained to be structurally determinate, that is unable to move in space without deformation while still being able to fully deform; this is to ensure numerical feasibility of the simulation. The membrane of the airship is then attached, with a prescribed thickness within the lifting capabilities of the design. This takes care of the entire evacuated airship structure, which is at this point ready for loading and simulation. Loads on the evacuated airship are prescribed by the atmospheric pressure on Mars, and modeled as distributed loads over the surface of the membrane, normal to said surface.

The simulation of the evacuated airship made use of geometric non-linearities, which allow the pressure distribution to conform to the deformation of the membrane. This is much more accurate to the real world instance because atmospheric pressure follows the curvature of a body, as opposed to a linear application of the pressure. Additionally, the simulation was conducted for non-linear strain states of the material to obtain a more accurate result. Due to the high complexity of the structural model, loading, and simulation scope, a dynamic analysis was performed. A dynamic analysis allows the structure to deform dynamically, which can be useful for many different cases, but here the dynamic analysis is used to find the steady state of the structure under deformation. The final stress state for the tensegrity beam is shown in Figure 12.1 as determined by the FEA simulation.

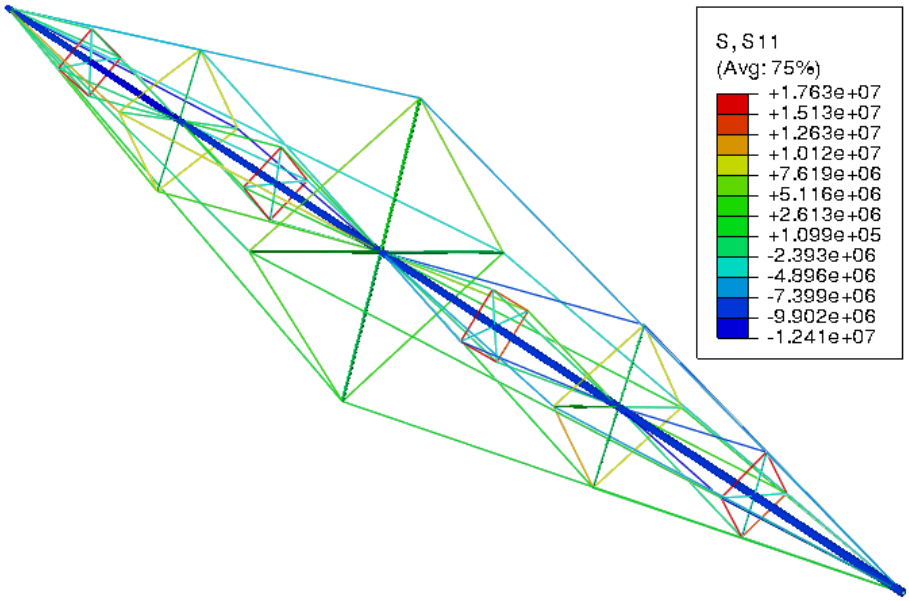


Figure 12.1

As such, the simulation resulted in the following deformation and stress fields shown in Figures 12.2 and 12.3 respectively for a membrane thickness of 80 microns of Miralon, a carbon fiber-like material developed through NASA research. The deformation seen in Figure 12.1 equates to 1493 kilograms in lost lift, which still leaves mass for payload when all is said and done. Furthermore, the effects of this deformation may be further reduced and the membrane made more resilient by non-uniform distribution of the thickness. This modulation of the thickness will allow the stress concentrations to be alleviated, increasing the reliability of the design as a whole. The aforementioned method can also reduce the deformation because the loss in lift is linked to key locations of deformation, which propagate through the membrane, leading to the loss in volume and consequent loss of lift.

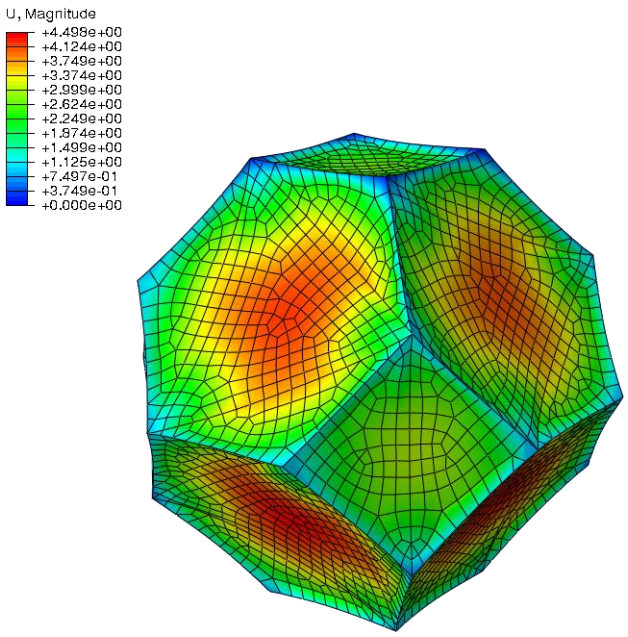


Figure 12.2

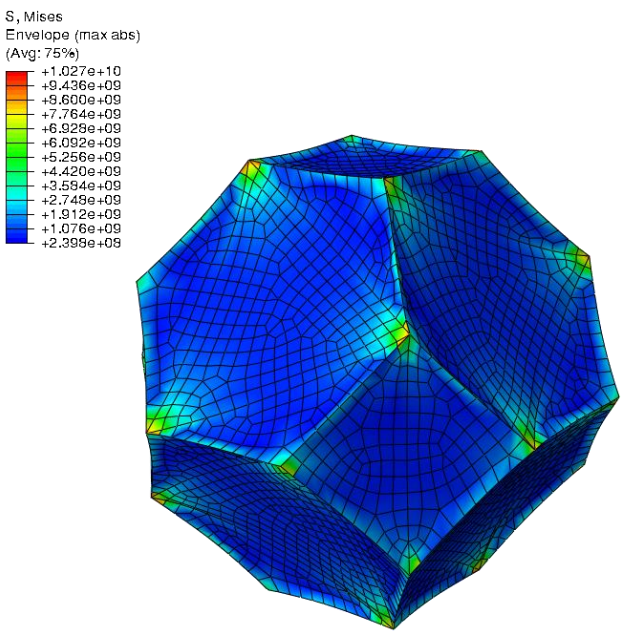


Figure 12.3

This FEA simulation validates the tensegrity beam theory used for design of the vacuum airship for Mars. As such, further application of the tensegrity beam theory may be useful and therefore warrants further investigation for both this and other applications requiring high strength, low mass structures. Additionally, this simulation shows that the current design for the evacuated airship is viable with consideration to all structural components, and as such demonstrates the feasibility of the concept and design.

Conclusion

This research has investigated the primary feasibility of the evacuated airship concept as a viable vehicle for Mars missions and exploration. At the conclusion of this investigation, there has been sufficient evidence to prove the feasibility of the evacuated airship as a tensegrity design using currently available materials. In addition, other designs have been analyzed and though found to be impossible with current materials, should materials technology continue to advance, these designs may prove feasible at a later date. Implementation of a monocoque shell utilizing a lattice based sandwich structure will be brought closer to feasibility with improvements to composite materials and their manufacturing capabilities. Likewise, implementation of the monocoque shell using a honeycomb sandwich structure will be made more feasible through advancements in ultra-low density materials, such as aerogels and the like.

The tensegrity based design using current materials, accounting for losses due to deflection of the membrane and other factors, is capable of 500 kilograms of payload. If trends in the mass reduction capabilities of the tensegrity beam design continue to higher complexities, which is probable, then implementation of a complexity 4 beam will result in 1 to 1.5 tons of payload capacity. In addition, the tensegrity design has been shown to offer full deployability and can be collapsed for transport to Mars. Consequently, this offers a feasible mission implementation path for the installment of the evacuated airship on Mars. Moreover, planetary insertion simulations have shown the design space of the atmospheric entry plan to be readily controllable to comply with the final needs of the evacuated airship design.

The structural aspects of the design have been fully verified using finite element analysis software. This computation involved detailed, non-linear finite element simulations, accounting for non-linearities in displacement, global and local buckling, and membrane failure criteria. These high fidelity tests utilize dynamic simulation to fully encapsulate the deformations and loadings on the structure in entirety. Additional validation for the mission aspects of the design comes from the numerical analysis associated with the atmospheric entry.

This effort should not be considered final in any respect. The investigation conducted here has been purposed to explore the most fundamental feasibility aspects of the evacuated airship concept as an installation on Mars. There are many other aspects to be explored to both further improve the capabilities of the evacuated airship as well as reinforce the mission within which the evacuated airship will be incorporated. As a result of this investigation, the foundational aspects of the evacuated airship have been established as they pertain to current technologies and the design has been shown to be feasible for implementation on Mars. Of course there will be more obstacles moving forward, but one should keep in mind there have been many obstacles in the path of arriving at this design point. Therefore, the most recent design iteration has plenty of avenues available for mitigation of difficulties, and is a strong foundation for future development.

References and Citations

1. Bruhn, E.F. (1973). Buckling Strength of Monocoque Cylinders. Analysis and Design of Flight Vehicle Structures 1973.

2. Skelton, R.E., Montuori, R. (2016, February 4). Globally stable minimal mass compressive tensegrity structures. Composite Structures 2016.

3. Donnell, L.H. (1935). Stability of Thin-Walled Tubes under Torsion. NACA Report No. 479, 1935.

4. FAA (N.D.). Returning from Space: Re-entry. Retrieved January 28, 2018, from https://www.faa.gov/about/office_org/headquarters_offices/avs/offices/aam/cami/library/online_libraries/aerospace_medicine/tutorial/media/III.4.1.7_Returning_from_Space.pdf

5. Griffin, M.D., French, J.R. (2004). Atmospheric Entry. Space Vehicle Design, Second Edition, 2004.

Appendix 1

```
Clear[ε, σs, Es, α, ρs, Eb, ρb, lo, f0, ψ]

"Physical Properties";
Eb = 200 * 10^9; (**)
ρb = 1150; (**)
ν = 0.1;
Patm = 700;
ρatm = 0.02;

(*αs = 100;*)
σs = 2000 * 10^6; (*3790*10^6;*)
Es = Eb;
ρs = ρb;

ψ = 0.99;
"Beam Length";
lo = 80;
a = lo / (6^0.5);

"Geometry and Force";
SArea = a^2 * (6 + 13 * 3^0.5);
f0 = Patm * SArea / 24;
V = a^3 * 8 * 2^0.5;
Lift = V * ρatm;

cl := ((*Clear[ε,αs,σs,Es,α,ρs,Eb,ρb,lo,f0,β];*)
   Clear[NcPossibilities];
   Clear[cs, cb, l, s, s3, k, kv, f, lv, t, ms, ms3, ml, mlv, ns, kvmax, kvmin];
   Clear[n, X, ik, R, M, i, R0, matrix, DisplacementList, kRules, Ncrit];
   Clear[k, AlphaRules, BetaRules, AgList, AreaRules, AreaList];
   Clear[A, A3, mbeams, mstrings, μb, μs, α, β];);
cl

complexity = 3;
minIndices = {0, 0, 2};
minInd = minIndices[[complexity]];

l[i_] := lo / (2^(i));
s[i_] := l[i] / Cos[α[i]];
f[i_] := f0 * Tan[α[i]];
lv[i_] := l[i] * Tan[α[i]];
t[i_] := f0 / (4 * complexity * Cos[α[i]]);
s3[i_] := lv[i] * Sqrt[2];
A3[i_] := A[i]; (*(f[i]/4)/σs;*)

ns[i_] := 2^(2 + i);
kvmax[i_] := Es * ml[l[i - 1]] / (l[i]^2 * ρs * 16);
kvmin[i_] := Es * f0 * Sin[1 * Degree]^2 / (2 * Sqrt[2] * l[i] * complexity * σs);

cb = 2 * ρb * (1 - ψ^2) / Sqrt[π * Eb * (1 - ψ^4)];
```

```
ms[i_] := A[i] * ρs * s[i];
ms3[i_] := A3[i] * ρs * s3[i];
ml[l_] := cb * l^2 * Sqrt[f0];
mlv[i_] := cb * lv[i]^2 * Sqrt[f[i]];

n = 2^complexity;
X[n] := 0;
For[j = 1, j ≤ n, j += 2, ik[j] = complexity;];
For[j = 2, j ≤ n, j += 2, ik[j] = ik[j / 2] - 1;];
R[j_] := X[j] * kv[ik[j]];
M[i_] := -Nc * X[i] + R0 * i * l[complexity] - Sum[R[j] * (i - j) * l[complexity], {j, 1, i}];
R0 = R0 /. Solve[M[1] == 0, R0][[1]];

DisplacementList = {};
For[i = 1, i ≤ n - 1, i += 1, DisplacementList = FlattenAt[{DisplacementList, X[i]}, 1]];
matrix = {};
For[i = 2, i ≤ n, i += 1, matrix =
   FlattenAt[{matrix, Normal[CoefficientArrays[M[i] == 0, DisplacementList]][[2]]}, 1];];
kRules =
 {lo >
   0};
For[i = 1, i ≤ complexity, i += 1, kRules =
  FlattenAt[{kRules, kv[i] > 0, kv[i] < kvmax[i]}, 1]];
NcPossibilities = Nc /. Solve[Det[matrix] == 0, Nc];
If[minInd == 0, Ncrit = Simplify[Min[NcPossibilities], kRules],
  Ncrit = Min[NcPossibilities][[minInd]];];
Print["N_crit = ", Ncrit];
MatrixForm[matrix];

"Apply Spring Equations and Create Rule Lists";
k[i_] := Es * Ag[i] / s[i];
kv[i_] := k[i] * (2 * Sin[α[i]]^2);
AlphaRules = {Es > 0, Eb > 0, f0 > 0, Ag[1] > 0, lo > 0};
For[i = 1, i ≤ complexity, i += 1,
 AlphaRules = FlattenAt[{AlphaRules, 45 * Degree > α[i] > 0}, 1]];
BetaRules = AlphaRules;
For[i = 1, i ≤ complexity, i += 1, BetaRules = FlattenAt[{BetaRules, β[i] > 0}, 1]];
AgList = True;
For[i = 2, i ≤ complexity, i += 1, AgList = AgList && Ag[i] == β[i] * Ag[1];];
AreaRules = BetaRules;
For[i = 1, i ≤ complexity, i += 1, AreaRules = FlattenAt[{AreaRules, Ag[i] > 0}, 1];];
AreaList = {};
For[i = 1, i ≤ complexity, i += 1, AreaList = FlattenAt[{AreaList, Ag[i]}, 1];];

(*A[i_] := Max[ (t[i]/σs),
   (AreaList[[i]]/.(Simplify[Solve[Ncrit==f0&&AgList,AreaList][[1,i]],AreaRules])) ];*)
A[i_] := Max[ (t[i] / σs), Simplify[
    Min[((Solve[Ncrit == f0 && AgList, AreaList][[All, i]]) /. Rule → List)[[All, 2]]],
    AreaRules]];
mbeams = n * ml[l[complexity]] + Sum[2^(i + 1) * mlv[i], {i, 1, complexity}];
```

```
mstrings =
  Sum[2^(i + 2) * ms[i], {i, 1, complexity}] + Sum[2^(i + 1) * ms3[i], {i, 1, complexity}];
μb = Simplify[mbeams / ml[lo], {f0 > 0}];
μs = mstrings / ml[lo];

μ = μb + μs;

Print["Non-Tensegrity Weight = ", BarSysWeight = 12 * ml[lo]];
(*Lift
 BarSysWeight/Lift*)
(*Print["Compressive Force = ",f0];*)
(*kvmax[1]
 kvmin[1]*)
(*a
 Sqrt[3]*a*)
Print[""];


Clear[minRes]
(*ε=ρs*Sqrt[f0*π*Eb]/(2*σs*ρb*lo);*)
ε = N[ρs * Sqrt[f0 * π * Eb] / (2 * σs * ρb * lo)];

AlphaList = {};
For[i = 1, i ≤ complexity, i += 1, AlphaList = FlattenAt[{AlphaList, α[i]}, 1];];
BetaList = {};
For[i = 2, i ≤ complexity, i += 1, BetaList = FlattenAt[{BetaList, β[i]}, 1];];
VarList = Flatten[{AlphaList, BetaList}];
α[0] = 45 Degree; β[1] = 1;
MinRules = True;
For[i = 1, i ≤ complexity, i += 1,
 MinRules = MinRules && α[0] > α[i] > 1 * Degree && α[i - 1] > (α[i] + 0 * Degree) && β[i] > 0];

minRes = Minimize[{μ, MinRules}, VarList];
For[i = 1, i ≤ complexity, i += 1,
  Print["α[", i, "] = ", (α[i] /. minRes[[2, i]]) / Degree]];
For[i = 2, i ≤ complexity, i += 1, β[i] = β[i] /. minRes[[2, complexity + i - 1]];
  Print["β[", i, "] = ", β[i] /. minRes[[2, complexity + i - 1]]]];
Print["μmin = ", minRes[[1]]]
Print["f0 = ", f0]
Clear[AlphaList, BetaList, VarList, MinRules]
Print[""]

LocalBuckling[L_, r_, t_, f_] := If[L^2 / (r * t) > 100,
   If[Eb t / (Sqrt[3] r Sqrt[1 - ν^2]) < f, "Local Buckling", "Locally Stable"] ": Long Cylinder",
   If[Eb π^2 t^2 (1 + 12 L^4 (1 - ν^2) / (π^4 r^2 t^2)) / (12 L^2 (1 - ν^2)) < f, "Local Buckling", "Locally Stable"]
    ": Transition Cylinder"];
```

```
l[i_] := lo / (2^(i));
r[i_] := ((f0 * 4 * l[i]^2) / (π^3 * Eb * (1 - ψ^4)))^0.25;
f[i_] := f0 * Tan[(α[i] /. minRes[[2, i]])];
lv[i_] := l[i] * Tan[(α[i] /. minRes[[2, i]])];
rv[i_] := ((f[i] * 4 * lv[i]^2) / (π^3 * Eb * (1 - ψ^4)))^0.25;
th[r_] := r * (1 - ψ);
t[i_] := f0 / (4 * complexity * Cos[α[i]]);
rs[A_] := Sqrt[A / π];
Print["          Horizontal Beams:          (only beam ", complexity, " is used)"]
For[i = 0, i ≤ complexity, i += 1,
  Print["l[", i, "] = ", N[l[i]]];
  Print["r[", i, "] = ", r[i]];
  Print["thickness of r[", i, "] = ", th[r[i]]];
  Print["     ", LocalBuckling[l[i], r[i], th[r[i]], f0]];];
Print[""]
Print["          Vertical Beams:"]
For[i = 1, i ≤ complexity, i += 1,
  Print["lv[", i, "] = ", lv[i]];
  Print["rv[", i, "] = ", rv[i]];
  Print["thickness of rv[", i, "] = ", th[rv[i]]];
  Print["     ", LocalBuckling[lv[i], rv[i], th[rv[i]], f[i]]];];

Print[""]
Print["VehicleWeight = ", (BarSysWeight * minRes[[1]])]
Print["Payload = ", Lift - (BarSysWeight * minRes[[1]])]
Print[""]
Print["Weight/Lift = ", (BarSysWeight * minRes[[1]]) / Lift]
compareProp = Flatten[{r[complexity], Evaluate[Array[rv, complexity]]}];

(*For[i=3,i≤complexity,i+=1,α[i]=(α[i]/.minRes[[2,i]]);];
Plot3D[μ,{α[1],1Degree,α[0]},{α[2],1Degree,α[0]},AxesLabel→{"α[1]","α[2]","μ"}]*)

For[i = 1, i ≤ complexity, i += 1, α[i] = (α[i] /. minRes[[2, i]]);];


"Node Calculations:";
NNodes = 2 + 5 * (2^complexity - 1);
(*Levels = 2^complexity+1;*)
Levels = 2^complexity + 1;
Nodes = {{0, 0, 0}};
For[i = 1, i < Levels - 1, i += 1,
  x = lv[ik[i]] (*/ModV[ik[i]]*); y = x; z = i * l[complexity];
  Nodes =
   FlattenAt[{Nodes, {x, y, z}, {-x, y, z}, {-x, -y, z}, {x, -y, z}, {0, 0, z}}, 1];];
Nodes = FlattenAt[{Nodes, {0, 0, l[0]}}, 1];
(*MatrixForm[Nodes]*)

σ0[i_] := -f0 / (π * r[i]^2 * (1 - ψ^2));
```

```
σv[i_] := -f0 * Tan[α[ik[i]]] / (π * rv[ik[i]]^2 * (1 - ψ^2));

BeamConnectivity = {}; BeamProperties = {}; StringRadList = {};
"Beam Connectivity:";
For[i = 1, i < Levels - 1, i += 1,
  CN = 1 + 5 * i;
  vprop = {rv[ik[i]] * ψ, rv[ik[i]], -1, N[σv[i]], Eb};
  BeamConnectivity = FlattenAt[{BeamConnectivity,
     {CN - 5, CN}, {CN, CN - 1}, {CN, CN - 2}, {CN, CN - 3}, {CN, CN - 4}}, 1];
  BeamProperties = FlattenAt[{BeamProperties,
     {r[complexity] * ψ, r[complexity], -1, N[σ0[complexity]], Eb},
     vprop, vprop, vprop, vprop}, 1];];
BeamConnectivity = FlattenAt[{BeamConnectivity, {CN, CN + 1}}, 1]; Clear[CN];
BeamProperties =
  FlattenAt[{BeamProperties, {r[complexity] * ψ, r[complexity], -1, 0, Eb}}, 1];
"String Connectivity:";
Connectivity = BeamConnectivity; ConnectorProperties = BeamProperties;
CableConnectivity = {}; CableProperties = {};
For[i = 1, i < Levels - 1, i += 1,
  BN = {1, 1, 11, 1, 21, 21, 31}[[i]]; CN = 1 + 5 * i;
  EN = Min[BN + 5 * 2^(ik[1] - ik[i] + 1), NNodes];
  Connectivity = FlattenAt[{Connectivity,
     {BN, CN - 1}, {BN, CN - 2}, {BN, CN - 3}, {BN, CN - 4},
     {CN - 1, CN - 2}, {CN - 2, CN - 3}, {CN - 3, CN - 4}, {CN - 4, CN - 1},
     {EN, CN - 1}, {EN, CN - 2}, {EN, CN - 3}, {EN, CN - 4}}, 1];
  CableConnectivity = FlattenAt[{CableConnectivity,
     {BN, CN - 1}, {BN, CN - 2}, {BN, CN - 3}, {BN, CN - 4},
     {CN - 1, CN - 2}, {CN - 2, CN - 3}, {CN - 3, CN - 4}, {CN - 4, CN - 1},
     {EN, CN - 1}, {EN, CN - 2}, {EN, CN - 3}, {EN, CN - 4}}, 1];
  sprop = {0, rs[A[ik[i]]], σs, t[ik[i]], Es};
  prop3 = {0, rs[A3[ik[i]]], σs, 0, Es};
  StringRadList = Flatten[{StringRadList, rs[A[ik[i]]]}];
  ConnectorProperties = FlattenAt[{ConnectorProperties,
     sprop, sprop, sprop, sprop,
     prop3, prop3, prop3, prop3,
     sprop, sprop, sprop, sprop}, 1];
  CableProperties = FlattenAt[{CableProperties,
     sprop, sprop, sprop, sprop,
     prop3, prop3, prop3, prop3,
     sprop, sprop, sprop, sprop}, 1];
 ];
(*Print[MatrixForm[Connectivity],MatrixForm[ConnectorProperties]]*)

Export["nodes.xls", Nodes];
Export["connectivity.xls", Connectivity];
Export["properties.xls", ConnectorProperties];


"ABAQUS data export";
currentFolder = NotebookDirectory[];
Export[currentFolder <> "\\nodes.txt", Nodes];
```

```
Export[currentFolder <> "\\connectivity.txt", Connectivity];
Export[currentFolder <> "\\properties.txt", ConnectorProperties];

kvList = {};
For[i = 1, i ≤ complexity, i += 1, kvList = Flatten[{kvList, A[i] * kv[i] / Ag[i]}];];
For[i = 1, i ≤ complexity, i += 1, kv[i] = kvList[[i]];];
NcList = DeleteCases[_Complex][NcPossibilities];
CGBC = minInd = Position[NcPossibilities, Min[NcList]][[1, 1]];
Print["Complex Global Buckling Value Assumption is ", CGBC];
If[CGBC, "";, MinInd = Position[NcPossibilities, Min[NcList]][[1, 1]];
  Print["Minimum Index for CGB is: ", MinInd];];
Print["Minimum Thickness: ", N[22 * 10^-6]];

Clear[conLines]; showLines = {};
travel[i_] := Max[Nodes[[All, i]]] - Min[Nodes[[All, i]]];
ratios = {travel[1], travel[2], travel[3]};
For[i = 1, i ≤ Length[BeamConnectivity], i += 1,
  conLines[i] = Graphics3D[
    {Red, Cylinder[{Nodes[[Connectivity[[i, 1]]]], Nodes[[Connectivity[[i, 2]]]]},
      ConnectorProperties[[i, 2]]]}, BoxRatios → ratios];
  showLines = Flatten[{showLines, conLines[i]}]];
For[i = Length[BeamConnectivity] + 1, i ≤ Length[Connectivity], i += 1,
  conLines[i] = Graphics3D[
    {Gray, Cylinder[{Nodes[[Connectivity[[i, 1]]]], Nodes[[Connectivity[[i, 2]]]]},
      ConnectorProperties[[i, 2]]]}, BoxRatios → ratios];
  showLines = Flatten[{showLines, conLines[i]}]];
Print[exp = Show[showLines, Boxed → False]];

(*Export["TenBeamTest.stl",exp];*)

prescribedPayload = 1000;
ρskin = 500;
Print["Maximum Skin Thickness with ", prescribedPayload, " kg Payload is ",
  (Lift - (BarSysWeight * minRes[[1]] + prescribedPayload)) / (ρskin * SArea) / (10^-3),
  " mm."];
Print["Minimum Thickness: ", N[22 * 10^-6] / (10^-3)];

cl
```

$$N_{crit} = 5\left(kv[2] + kv[3] - \sqrt{kv[2]^2 + kv[3]^2}\right)$$

```
Non-Tensegrity Weight = 21042.1
```

$$\alpha[1] = 8.71565$$

$$\alpha[2] = 8.71565$$

$$\alpha[3] = 8.71565$$

$$\beta[2] = 1.08737$$

$\beta[3]$ = 2.54076

μ_{min} = 0.179226

f_0 = 887185.

Horizontal Beams: (only beam 3 is used)

l[0] = 80.

r[0] = 0.552152

thickness of r[0] = 0.00552152

Locally Stable : Long Cylinder

l[1] = 40.

r[1] = 0.39043

thickness of r[1] = 0.0039043

Locally Stable : Long Cylinder

l[2] = 20.

r[2] = 0.276076

thickness of r[2] = 0.00276076

Locally Stable : Long Cylinder

l[3] = 10.

r[3] = 0.195215

thickness of r[3] = 0.00195215

Locally Stable : Long Cylinder

Vertical Beams:

lv[1] = 6.13204

rv[1] = 0.0956538

thickness of rv[1] = 0.000956538

Locally Stable : Long Cylinder

lv[2] = 3.06602

rv[2] = 0.0676375

thickness of rv[2] = 0.000676375

Locally Stable : Long Cylinder

lv[3] = 1.53301

rv[3] = 0.0478269

thickness of rv[3] = 0.000478269

Locally Stable : Long Cylinder

```
VehicleWeight = 3771.29

Payload = 4111.47

Weight/Lift = 0.478423

Complex Global Buckling Value Assumption is True

Minimum Thickness: 0.000022

Maximum Skin Thickness with 1000 kg Payload is 0.204582 mm.

Minimum Thickness: 0.022

showLines = {}; Clear[x, y, z, x2, y2];

"Nodes";
TenNodes = {};
(*Nodes=FlattenAt[{Nodes,{0,0,0},{a,0,0},{a,a,0},{0,a,0}},1];*)
x = y = a / 2; z = -1 * a * 2^0.5;(*z=0;*)
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
x = y = a; z = -1 / 2 * a * 2^0.5;(*z=a*2^0.5;*)
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
x1 = y2 = 3 * a / 2; x2 = y1 = a / 2; z = 0;(*z=2*a*2^0.5;*)
TenNodes = FlattenAt[{TenNodes, {x2, y2, z}, {x1, y1, z}, {x1, -y1, z},
    {x2, -y2, z}, {-x2, -y2, z}, {-x1, -y1, z}, {-x1, y1, z}, {-x2, y2, z}}, 1];
x = y = a; z = 1 / 2 * a * 2^0.5;(*z=3*a*2^0.5;*)
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
x = y = a / 2; z = a * 2^0.5;(*z=4*a*2^0.5;*)
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
"Connectivity";
TenBeamConnectivity = {{1, 12}, {2, 14}, {3, 16}, {4, 10},
   {15, 21}, {9, 22}, {11, 23}, {13, 24},
   {5, 20}, {6, 17}, {7, 18}, {8, 19}};
TenStringConnectivity = {{1, 2}, {2, 3}, {3, 4}, {4, 1},
   {1, 5}, {2, 6}, {3, 7}, {4, 8},
   {5, 9}, {5, 10}, {6, 11}, {6, 12}, {7, 13}, {7, 14}, {8, 15}, {8, 16},
   {17, 9}, {17, 10}, {18, 11}, {18, 12}, {19, 13}, {19, 14}, {20, 15}, {20, 16},
   {17, 21}, {18, 22}, {19, 23}, {20, 24},
   {21, 22}, {22, 23}, {23, 24}, {24, 21},
   {10, 11}, {12, 13}, {14, 15}, {16, 9}};

StringRadius = Max[StringRadList];
OutStrProp = Table[{0, StringRadius, CableProperties[[1, 3]],
    0, CableProperties[[1, 5]]}, Length[TenStringConnectivity]];
Clear[conLines]; bshowLines = {};
travel[i_] := Max[TenNodes[[All, i]]] - Min[TenNodes[[All, i]]];
ratios = {travel[1], travel[2], travel[3]};
For[i = 1, i ≤ Length[TenBeamConnectivity], i += 1,
  conLines[i] = Graphics3D[{Red, Cylinder[{TenNodes[[TenBeamConnectivity[[i, 1]]]],
       TenNodes[[TenBeamConnectivity[[i, 2]]]]}, 1]}, BoxRatios → ratios];
  bshowLines = Flatten[{bshowLines, conLines[i]}]];
```

```
For[i = 1, i ≤ Length[TenStringConnectivity], i += 1,
  conLines[i] = Graphics3D[{Black, Cylinder[{TenNodes[[TenStringConnectivity[[i, 1]]]],
       TenNodes[[TenStringConnectivity[[i, 2]]]]}, StringRadius]}, BoxRatios → ratios];
  showLines = Flatten[{showLines, conLines[i]}]];

FullNodes = {}; FullConnectivity = {};
FullStringConnectivity = {}; FullBeamConnectivity = {};
FullStringProperties = {}; FullBeamProperties = {};
NodeConv = ConstantArray[0, Max[TenBeamConnectivity]];
For[j = 1, j ≤ Length[TenBeamConnectivity], j += 1,

  ModNodes = Nodes;
  NewOrigin = TenNodes[[TenBeamConnectivity[[j, 1]]]];
  NewEnd = TenNodes[[TenBeamConnectivity[[j, 2]]]];
  (*NewOrigin={16.32993161855452,16.32993161855452,-46.188021535170066};
  NewEnd={16.32993161855452,-48.98979485566356,0};*)
  eprime = (NewEnd - NewOrigin) / (Norm[NewEnd - NewOrigin]); e = {0, 0, 1};
  v = Cross[e, eprime]; s = Norm[v]; c = e.eprime;
  vx = {{0, -v[[3]], v[[2]]}, {v[[3]], 0, -v[[1]]}, {-v[[2]], v[[1]], 0}};
  R = IdentityMatrix[3] + vx + vx.vx * ((1 - c) / s^2);
  For[i = 1, i ≤ Length[Nodes], i += 1,
   ModNodes[[i]] = R.Nodes[[i]] + NewOrigin;];
  NodeConv[[TenBeamConnectivity[[j, 1]]]] = Length[FullNodes] + 1;
  NodeConv[[TenBeamConnectivity[[j, 2]]]] = Length[FullNodes] + Length[ModNodes];
  FullNodes = FlattenAt[{FullNodes, ModNodes}, {{1}, {2}}];
  Clear[NewOrigin, NewEnd, eprime, e, v, s, c, vx, R];

  Clear[conLines];
  travel[i_] := Max[ModNodes[[All, i]]] - Min[ModNodes[[All, i]]];
  ratios = {travel[1], travel[2], travel[3]};
  For[i = 1, i ≤ Length[BeamConnectivity], i += 1,
   conLines[i] = Graphics3D[
     {Red, Cylinder[{ModNodes[[Connectivity[[i, 1]]]], ModNodes[[Connectivity[[i, 2]]]]},
       ConnectorProperties[[i, 2]]]}, BoxRatios → ratios];
   showLines = Flatten[{showLines, conLines[i]}]];
  For[i = Length[BeamConnectivity] + 1, i ≤ Length[Connectivity], i += 1,
   conLines[i] = Graphics3D[{Black,
      Cylinder[{ModNodes[[Connectivity[[i, 1]]]], ModNodes[[Connectivity[[i, 2]]]]},
       ConnectorProperties[[i, 2]]]}, BoxRatios → ratios];
   showLines = Flatten[{showLines, conLines[i]}]];

  lev = (j - 1) * Max[Connectivity];
  For[i = 1, i ≤ Length[Connectivity], i += 1,
   FullConnectivity = FlattenAt[
      {FullConnectivity, {Connectivity[[i, 1]] + lev, Connectivity[[i, 2]] + lev}}, 1];
  ];
  For[i = 1, i ≤ Length[BeamConnectivity], i += 1,
   FullBeamConnectivity = FlattenAt[
     {FullBeamConnectivity, {Connectivity[[i, 1]] + lev, Connectivity[[i, 2]] + lev}}, 1];
   FullBeamProperties = FlattenAt[{FullBeamProperties, ConnectorProperties[[i]]}, 1];
  ];
```

```
  For[i = Length[BeamConnectivity] + 1, i ≤ Length[Connectivity], i += 1,
   FullStringConnectivity = FlattenAt[{FullStringConnectivity,
      {Connectivity[[i, 1]] + lev, Connectivity[[i, 2]] + lev}}, 1];
   FullStringProperties = FlattenAt[{FullStringProperties,
      ConnectorProperties[[i]]}, 1];
  ];
 ];

OutStrCon = TenStringConnectivity;
For[i = 1, i ≤ Length[TenStringConnectivity], i += 1,
  OutStrCon[[i, 1]] = NodeConv[[TenStringConnectivity[[i, 1]]]];
  OutStrCon[[i, 2]] = NodeConv[[TenStringConnectivity[[i, 2]]]];];
FullStringConnectivity = FlattenAt[{FullStringConnectivity, OutStrCon}, {{1}, {2}}];
FullStringProperties = Catenate[{FullStringProperties, OutStrProp}];


FullOutConnectivity = Catenate[{FullBeamConnectivity, FullStringConnectivity}];
FullOutProperties = Catenate[{FullBeamProperties, FullStringProperties}];
"ABAQUS data export";
currentFolder = NotebookDirectory[];
Export[currentFolder <> "\\nodes.txt", FullNodes];
Export[currentFolder <> "\\connectivity.txt", FullOutConnectivity];
Export[currentFolder <> "\\properties.txt", FullOutProperties];


(*fshowLines={};
travel[i_]:=Max[FullNodes[[All,i]]]-Min[FullNodes[[All,i]]];
ratios={travel[1],travel[2],travel[3]};
For[i=1,i≤Length[FullBeamConnectivity],i+=1,
 conLines=Graphics3D[{Red,Cylinder[{FullNodes[[FullBeamConnectivity[[i,1]]]],
       FullNodes[[FullBeamConnectivity[[i,2]]]]},0.05]},BoxRatios→ratios];
 fshowLines=Flatten[{fshowLines,conLines}];];
For[i=1,i≤Length[FullStringConnectivity],i+=1,
 conLines=Graphics3D[{Green,Cylinder[{FullNodes[[FullStringConnectivity[[i,1]]]],
       FullNodes[[FullStringConnectivity[[i,2]]]]},0.05]},BoxRatios→ratios];
 fshowLines=Flatten[{fshowLines,conLines}];];
(*For[i=1,i≤Length[TenStringConnectivity],i+=1,
  conLines=
   Graphics3D[{Gray,Cylinder[{FullNodes[[NodeConv[[TenStringConnectivity[[i,1]]]]]],
        FullNodes[[NodeConv[[TenStringConnectivity[[i,2]]]]]},1]},BoxRatios→ratios];
  fshowLines=Flatten[{fshowLines,conLines}];];*)
Print[exp1 = Show[fshowLines,Boxed→False]];*)

FullBeamConnectivity = FullBeamConnectivity - 1;
FullStringConnectivity = FullStringConnectivity - 1;
StrFormat[list_] := (str = "";
   For[k = 1, k ≤ Length[list], k += 1, strpls = If[Norm[list[[k]]] > 10^-5, list[[k]], 0];
    If[k == 1, str = str <> ToString[strpls], str = str <> " " <> ToString[strpls]];];
   Return[str]);
var = ToString[Length[FullNodes]] <> "\n";
For[i = 1, i ≤ Length[FullNodes], i += 1,
```

```
  var = var <> StrFormat[FullNodes[[i]]] <> "\n";];
var = var <> ToString[Length[FullBeamConnectivity]] <> "\n";
For[i = 1, i ≤ Length[FullBeamConnectivity], i += 1,
  var = var <> StrFormat[FullBeamConnectivity[[i]]] <> " 1 1 1 1\n";];
var = var <> ToString[Length[FullStringConnectivity]] <> "\n";
For[i = 1, i ≤ Length[FullStringConnectivity], i += 1,
  var = var <> StrFormat[FullStringConnectivity[[i]]] <> " 1 1 1 1 1\n";];
Export["FileForDrRimoli.txt", var];


Print[exp = Show[showLines, Boxed → False]];
```

Appendix 2

```
"Inputs:";
JobName = "TensegrityAirship";
(*nodes = {{0,0,0},{0,0,10},{0,10,10},{0,10,0}};
(*{x,y,z}*)
connectivity = {
  {1,2},
  {2,3},
  {3,1},
  {3,4},
  {4,1},
  {2,4},
  {2,3}};
(*{node 1, node 2} with a beam connecting node 1 to node 2*)
properties = {
  {1.4,1.5,1000,100000,0.3,10,0, "tube"},
  {1.4,1.5,1000,100000,0.3,10,0, "tube"},
  {1.4,1.5,1000,100000,0.3,10,0, "tube"},
  {1.4,1.5,1000,100000,0.3,10,0, "tube"},
  {1.4,1.5,1000,100000,0.3,10,0, "tube"},
  {0.1,0.2,1000,100000,0.3,1,100,"cable"},
  {0.1,0.2,1000,100000,0.3,1,100,"cable"}};
(*{inner radius, outter radius, density, young's modulus,
 poisson ratio, number of elements, pre-tension value, part type}*)*)
destination = ToString[NotebookDirectory[]] <> JobName <> ".inp";

"*Initial Conditions, type=STRESS
element_number, stress_value, 0., 0.,"; (*Goes between step and materials*)

"Beam Length";
lo = 80;
a = lo / (6^0.5);
"Skin Nodes";
TenNodes = {};
x = y = a / 2; z = -1 * a * 2^0.5;
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
x = y = a; z = -1 / 2 * a * 2^0.5;
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
x1 = y2 = 3 * a / 2; x2 = y1 = a / 2; z = 0;
TenNodes = FlattenAt[{TenNodes, {x2, y2, z}, {x1, y1, z}, {x1, -y1, z},
     {x2, -y2, z}, {-x2, -y2, z}, {-x1, -y1, z}, {-x1, y1, z}, {-x2, y2, z}}, 1];
x = y = a; z = 1 / 2 * a * 2^0.5;
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];
x = y = a / 2; z = a * 2^0.5;
TenNodes = FlattenAt[{TenNodes, {x, y, z}, {x, -y, z}, {-x, -y, z}, {-x, y, z}}, 1];


"Initializations";
FileString = "*Heading\n";
FileString = FileString <> "";
FileString = FileString <> JobName <> "\n";
```

```
FileString =
  FileString <> "*Preprint, echo=NO, model=NO, history=NO, contact=NO\n**\n";
FileString = FileString <> "**\n**\n**\n";

majorNodes = Table[{}, Length[nodes]];
nodeCount = 0;
elementCount = 0;
materials = {};
pretensionSets = {};

elementDef[n_] := {n, n + 1};

TubeElements[elementCount_, elements_] := (ElementString = "";
   ElementString = ElementString <> "*Element, type=B31\n";
   For[j = 1, j ≤ Length[elements], j++, {subnode1, subnode2} = elements[[j]];
    ElementString = ElementString <> ToString[elementCount + j] <> ", ";
    ElementString = ElementString <> ToString[subnode1] <> ", ";
    ElementString = ElementString <> ToString[subnode2] <> "\n";];
   Return[ElementString];);
TubeSection[part_, material_, r2_, r1_] := (SectionString = "";
   SectionString = SectionString <> "*Beam Section, elset=Part-" <> ToString[part] <>
     "-Set-1, material=Material-" <> ToString[material] <> ", section=PIPE\n";
   SectionString = SectionString <> ToString[r2] <> ", " <> ToString[r2 - r1] <> "\n";
   Return[SectionString];);

CableElements[elementCount_, elements_] := (ElementString = "";
   ElementString = ElementString <> "*Element, type=T3D2\n";
   For[j = 1, j ≤ Length[elements], j++, {subnode1, subnode2} = elements[[j]];
    ElementString = ElementString <> ToString[elementCount + j] <> ", ";
    ElementString = ElementString <> ToString[subnode1] <> ", ";
    ElementString = ElementString <> ToString[subnode2] <> "\n";];
   Return[ElementString];);
CableSection[part_, material_, r_] := (SectionString = "";
   SectionString = SectionString <> "*Solid Section, elset=Part-" <>
     ToString[part] <> "-Set-1, material=Material-" <> ToString[material] <> "\n";
   SectionString = SectionString <> ToString[N[π * r^2]] <> ", " <> "\n";
   Return[SectionString];);



"Create Parts";
FileString = FileString <> "** PARTS\n**\n";
For[i = 1, i ≤ Length[connectivity], i++,

  "Start new part";
  FileString = FileString <> "*Part, name=Part-" <> ToString[i] <> "\n";


  "Create nodes for elements";
  {n1, n2} = connectivity[[i]];
  {x1, y1, z1} = nodes[[n1]];
```

```
{x2, y2, z2} = nodes[[n2]];
{r1, r2, ρ, Emat, ν, Nelements, pretension, type} = properties[[i]];
subNodes = Transpose[N[Round[
    {Array[ListInterpolation[{x1, x2}, InterpolationOrder -> 1], Nelements + 1, {1, 2}],
     Array[ListInterpolation[{y1, y2}, InterpolationOrder -> 1], Nelements + 1, {1, 2}],
     Array[ListInterpolation[{z1, z2}, InterpolationOrder -> 1], Nelements + 1, {1, 2}]},
    10^-8]]];
"Create elements";
elements = Array[elementDef, Nelements, nodeCount + 1];

"Add nodes to string";
FileString = FileString <> "*Node\n";
For[j = 1, j ≤ Length[subNodes], j++,
 {subx, suby, subz} = subNodes[[j]];
 FileString = FileString <> ToString[nodeCount + j] <> ", ";
 FileString = FileString <> ToString[subx] <> ", ";
 FileString = FileString <> ToString[suby] <> ", ";
 FileString = FileString <> ToString[subz] <> "\n";
];
majorNodes[[n1]] = FlattenAt[{majorNodes[[n1]], {nodeCount + 1, i, {x1, y1, z1}}}, 1];
nodeCount += Length[subNodes];
majorNodes[[n2]] = FlattenAt[{majorNodes[[n2]], {nodeCount, i, {x2, y2, z2}}}, 1];

"Add elements to string";
If[type == "tube", ElementStr = CableElements[elementCount, elements];,
 If[type == "cable",
   ElementStr = CableElements[elementCount, elements];, Print["error"];];];
FileString = FileString <> ElementStr;
elementCount += Length[elements];

"Build and add Nsets and Elsets";
FileString = FileString <> "*Nset, nset=Part-" <> ToString[i] <> "-Set-1, generate\n";
FileString = FileString <> ToString[nodeCount - Length[subNodes] + 1] <> ", ";
FileString = FileString <> ToString[nodeCount] <> ", 1\n";
If[Length[elements] > 1,
 FileString =
  FileString <> "*Elset, elset=Part-" <> ToString[i] <> "-Set-1, generate\n";
 FileString = FileString <> ToString[elementCount - Length[elements] + 1] <> ", ";
 FileString = FileString <> ToString[elementCount] <> ", 1\n",
 FileString = FileString <> "*Elset, elset=Part-" <> ToString[i] <> "-Set-1\n";
 FileString = FileString <> ToString[elementCount] <> "\n"
];

FileString =
 FileString <> "*Nset, nset=p_Set-" <> ToString[nodeCount - Length[subNodes] + 1] <> "\n";
FileString = FileString <> ToString[nodeCount - Length[subNodes] + 1] <> "\n";
FileString = FileString <> "*Nset, nset=p_Set-" <> ToString[nodeCount] <> "\n";
FileString = FileString <> ToString[nodeCount] <> "\n";
FileString = FileString <> "*Elset, elset=s_Set-0\n";
FileString = FileString <> ToString[elementCount - Length[elements] + 1] <> "\n";
```

```
  "Build and add beam section and material";
  If[MemberQ[materials, {ρ, Emat, ν}],
   material = Position[materials, {ρ, Emat, ν}][[1, 1]],
   materials = FlattenAt[{materials, {ρ, Emat, ν}}, 1];
   material = Length[materials]];
  If[type == "tube", SectionStr = CableSection[i, material, r2];,
   If[type == "cable", SectionStr = CableSection[i, material, r2];, Print["error"];];];
  FileString = FileString <> SectionStr;

  "End the part";
  FileString = FileString <> "*End Part\n**\n";
 ];
FileString = FileString <> "**\n**\n**\n";



"Create Assembly";
FileString = FileString <> "** ASSEMBLY\n**\n";
FileString = FileString <> "*Assembly, name=Assembly-1\n**\n";

"Create part instances";
For[i = 1, i ≤ Length[connectivity], i++,
  FileString = FileString <> "*Instance, name=Part-" <>
    ToString[i] <> "-1, part=Part-" <> ToString[i] <> "\n";
  FileString = FileString <> "*End Instance\n**\n";
 ];

"Create pins";
endNodeString = "";
For[i = 1, i ≤ Length[majorNodes], i++,
  pinList = majorNodes[[i]];
  If[Length[pinList] > 1,
   FileString = FileString <> "** Pin Joint at " <> ToString[i] <> "\n";
   For[j = 1, j ≤ (Length[pinList] - 1), j++,
    FileString = FileString <> "*EQUATION\n2\n";
    FileString = FileString <> "Part-" <>
      ToString[pinList[[j, 2]]] <> "-1." <> ToString[pinList[[j, 1]]] <> ", 1, 1, ";
    FileString = FileString <> "Part-" <> ToString[pinList[[j + 1, 2]]] <>
      "-1." <> ToString[pinList[[j + 1, 1]]] <> ", 1, -1\n";
    FileString = FileString <> "*EQUATION\n2\n";
    FileString = FileString <> "Part-" <>
      ToString[pinList[[j, 2]]] <> "-1." <> ToString[pinList[[j, 1]]] <> ", 2, 1, ";
    FileString = FileString <> "Part-" <> ToString[pinList[[j + 1, 2]]] <>
      "-1." <> ToString[pinList[[j + 1, 1]]] <> ", 2, -1\n";
    FileString = FileString <> "*EQUATION\n2\n";
    FileString = FileString <> "Part-" <>
      ToString[pinList[[j, 2]]] <> "-1." <> ToString[pinList[[j, 1]]] <> ", 3, 1, ";
    FileString = FileString <> "Part-" <> ToString[pinList[[j + 1, 2]]] <>
      "-1." <> ToString[pinList[[j + 1, 1]]] <> ", 3, -1\n";
   ];
   FileString = FileString <> "**\n";
```

```
    If[AnyTrue[Table[Sum[Abs[pinList[[j, 3, k]] - TenNodes[[m, k]]], {k, 3}],
       {m, 1, Length[TenNodes]}], # < 0.1 &],
     endNodeString = endNodeString <> "Part-" <> ToString[pinList[[j, 2]]] <> "-1." <>
        ToString[pinList[[j, 1]]] <> ", " <> ToString[pinList[[j, 3, 1]]] <> ", " <>
        ToString[pinList[[j, 3, 2]]] <> ", " <> ToString[pinList[[j, 3, 3]]] <> "\n";
    ];
   ];
  ];
Export[ToString[NotebookDirectory[]] <> "end_chain_nodes.txt", endNodeString, "Text"];
FileString = FileString <> "**\n";
FileString = FileString <> "*End Assembly\n**\n";
FileString = FileString <> "**\n**\n**\n";



"Create Materials";
FileString = FileString <> "** MATERIALS\n**\n";
For[i = 1, i ≤ Length[materials], i++,

  "Start new material";
  FileString = FileString <> "*Material, name=Material-" <> ToString[i] <> "\n";

  "Get material properties";
  {ρ, Emat, ν} = materials[[i]];

  "Write materials to string";
  FileString = FileString <> "*Density\n";
  FileString = FileString <> ToString[ρ] <> ",\n";
  FileString = FileString <> "*Elastic\n";
  FileString = FileString <> ToString[Emat] <> ", " <> ToString[ν] <> "\n";
  FileString = FileString <> "**\n";
 ];
FileString = FileString <> "**\n**\n**\n";


ABAQUSstring[number_] := (
   i = 0;
   While[Abs[number / (10^i)] > 1, i += 1];
   i -= 1;
   Return[ToString[number / (10^i)] <> "E" <> ToString[i]];);
pretensions = {};
For[i = 1, i ≤ Length[properties], i++,
  If[MemberQ[pretensions, properties[[i, 7]]], Null,
    pretensions = Join[pretensions, {properties[[i, 7]]}]];
 ];
pretensionStrings = Table[ABAQUSstring[pretensions[[i]]], {i, 1, Length[pretensions]}];

"*Initial Conditions, type=STRESS
element_number, stress_value, 0., 0.,"; (*Goes between step and materials*)
"Create Pre-Stresses";
```

```
FileString = FileString <> "** PRE-TENSIONS\n";
For[i = 1, i ≤ Length[properties], i++,
  If[properties[[i, 7]] ≠ 0,
    pretension = properties[[i, 7]];
    element = "Part-" <> ToString[i] <> "-1.s_Set-0";
    FileString = FileString <> "*Initial Conditions, type=STRESS\n";
    FileString = FileString <> element <> ", " <>
      pretensionStrings[[Position[pretensions, pretension][[1, 1]]]] <> ", 0., 0.,\n";
   ];
 ];


"Write File";
Export[destination, FileString, "Text"];
(*Print[FileString]*)



currentFolder = NotebookDirectory[];
string = OpenRead[currentFolder <> "\\nodes.txt"];
nodes = ReadList[string];
Close[string];
string = OpenRead[currentFolder <> "\\connectivity.txt"];
connectivity = ReadList[string];
Close[string];
string = OpenRead[currentFolder <> "\\properties.txt"];
inputproperties = ReadList[string];
Close[string];
Clear[string];

properties = Table[0, Length[inputproperties]];
ρ = 1150;
ν = 0.1;
For[i = 1, i ≤ Length[inputproperties], i++,
  {r1, r2, trash1, pretension, Emat} = inputproperties[[i]];
  If[trash1 == -1,
   Nelements = 1; type = "tube";,
   Nelements = 1; type = "cable";];
  properties[[i]] = {r1, r2, ρ, Emat, ν, Nelements, pretension, type};
 ];

Clear[r1, r2, ρ, Emat, ν, Nelements, pretension, type, trash1, i, inputproperties];
(*Print[MatrixForm[properties]]*)
```