



**DESIGN AND ANALYSIS OF AIR-STIFFENED VACUUM
LIGHTER-THAN-AIR STRUCTURES**

DISSERTATION

Ruben Adorno-Rodríguez, Captain, USAF
AFIT-ENY-DS-21-M-285

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-DS-21-M-285

DESIGN AND ANALYSIS OF AIR-STIFFENED VACUUM
LIGHTER-THAN-AIR STRUCTURES

DISSERTATION

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Ruben Adorno-Rodríguez, B.S., M.S.
Captain, USAF

March 2021

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENY-DS-21-M-285

DESIGN AND ANALYSIS OF AIR-STIFFENED VACUUM
LIGHTER-THAN-AIR STRUCTURES

Ruben Adorno-Rodríguez, B.S., M.S.
Captain, USAF

Committee Membership:

Anthony N. Palazotto, Ph.D.
Chairman

William P. Baker, Ph.D.
Member

Meir N. Pachter, Ph.D.
Member

Jaimie S. Tiley, Ph.D.
Member

Adedeji B. Badiru, Ph.D.
Dean, Graduate School of Engineering and Management

Abstract

Lighter-than-air (LTA) systems have been developed for numerous applications and have taken several forms. Airships, aerostats, blimps, and balloons are all part of this family of systems, which uses Archimedes principle to achieve neutral and positive buoyancy in air by replacing an air volume with LTA gases. These lifting gases stiffen the otherwise compliant envelope structures, allowing them to sustain the pressure difference brought by the displaced air. The compliance of these structures is a byproduct of the weight requirement, materials and geometrical arrangement of which these structures are built from, typically resulting in dimensionalities that exhibit low or virtually non-existent in-plane bending stiffness. The former has constrained the development of LTA structures that utilize an internal partial vacuum, rather than a lifting gas, to achieve positive buoyancy, where the structure would be subjected to a pressure differential near atmospheric pressure.

Given the above limitation, this research presents the development trajectory and structural characterization of air-stiffened designs, which utilize air to shape and serve as the core of a set of enclosing envelopes. The development trajectory established a simulation framework that enables the structural characterization of air-stiffened designs under a variety of geometric and loading conditions. Such framework allowed for the development of finite element solutions that included geometric, fluid-structure and contact nonlinearities, with capacity for further generalization. Given the developed framework, the structural characterization of the Helical Sphere and Icoron air-stiffened

designs demonstrated a reduction of material modulus and strength requirements compared to membrane-over-frame designs, and showed the capability of air-stiffened designs to be tailored for specific material strength limits.

AFIT-ENY-DS-21-M-285

*To my family, friends, teachers and mentors who have provided their time and energy to
help me grow and support my endeavors.*

Acknowledgements

I hereby express by deepest appreciation to my research advisor, Distinguished Professor Dr. Palazotto, for his invaluable commitment, advice and support throughout this journey. His expertise on nonlinear mechanics, enthusiasm for conveying knowledge, and guidance on its proper application energized the work presented here. I am extremely grateful for his guidance, support and confidence.

I would also like to extend my gratitude to Dr. Tiley, Dr. Pachter and Dr. Baker for their guidance, constructive feedback and support received throughout all the milestones that led to this work.

I would like to acknowledge my parents for their encouragement in the pursuit of higher education. From their support for my study of science and technology topics to their continued appreciation for my recent efforts, the tools they provided help me build the foundation that led to this milestone.

I can only try to express my deepest gratitude to my spouse. Her love, support, sacrifice, and ability to compensate for my shortcomings allowed me to complete my doctorate. I love her profoundly and am enormously grateful for all she has done. I can only hope that my efforts and continued journey inspires our children to build upon our teachings and experience for the betterment of their life.

Last but not least, I extend my gratitude to the Air Force Office of Scientific Research for funding this effort.

Ruben Adorno-Rodríguez

Table of Contents

	Page
List of Figures	xi
List of Tables	xvi
I. Introduction	1
1.1. Motivation	3
1.2. Research Scope.....	3
1.3. Dissertation Outline.....	7
II. Background and Related Work	10
2.1. Lighter-than-Air Systems	10
2.2. Structural Modeling and Simulation	30
2.3. Structural Stability	49
2.4. Surface Stress	52
2.5. Summary.....	54
III. Stability and Material Nonlinearity	57
3.1. Problem Description.....	58
3.2. Methodology.....	59
3.3. Equilibrium Paths	67
3.4. Summary and Conclusions	75
IV. Design Paths Investigation	78
4.1. Coated Spherical Shell	78
4.2. Topology Optimization	82
4.3. Stress Control Initial Validation.....	90
4.4. Summary and Conclusions	97

V. Helical Sphere Air-Stiffened Design.....	98
5.1. Design Description	99
5.2. Weight-to-Buoyancy Ratio Estimation	100
5.3. Preliminary Analysis	103
5.4. Structural Simulation Framework	106
5.5. Axisymmetric Representation of the Helical Sphere	116
5.6. Summary and Conclusions	136
VI. Icoron Air-Stiffened Design.....	139
6.1. Design Description	141
6.2. Geometry Effects on Expected Weight-to-Buoyancy Ratios and Stress.....	153
6.3. Case Study	157
6.4. Summary and Conclusions	173
VII. Conclusions	177
7.1. Summary of the Development Trajectory	177
7.2. Summary of Findings	178
7.3. Recommendations	181
Appendix A. Stability and Material Nonlinearity Supporting Information.....	184
Appendix B. Helical Sphere Air-Stiffened Design Supporting Information.....	197
Appendix C. Icoron Air-Stiffened Design Supporting Information.....	217
Appendix D. List of Abbreviations.....	243
Bibliography	244

List of Figures

	Page
Fig. I-1. Depiction of the lighter-than-air structure design space.	4
Fig. I-2. Generalized representation of air-stiffened designs.	6
Fig. I-3. Representations of air-stiffened designs considered.	7
Fig. I-4. Air-stiffened design development trajectory summary.	8
Fig. II-1. Example of lighter-than-air system applications for the intelligence, surveillance and reconnaissance mission set, 2012 [18].	11
Fig. II-2. Lighter-than-air system categorization, 2012 [18].	12
Fig. II-3. Average and seasonal wind speeds vs altitude and time of year over Kabul, Afghanistan and Baghdad, Iraq, 2011 [38].	16
Fig. II-4. Air density, pressure, temperature and speed of sound as a function of altitude [15,39].	17
Fig. II-5. Spherical diameter required for neutral buoyancy vs altitude for four masses [15,39].	18
Fig. II-6. Zero-pressure and super-pressure balloons comparison [41].	21
Fig. II-7. Super-pressure balloon inflation (left) and deployment, 2015 [40].	21
Fig. II-8. Pumpkin-shaped balloon rendition, 2007 [27].	22
Fig. II-9. SF420 LLDPE film stress vs strain, time, temperature and direction at strain rates of 0.1% s ⁻¹ and 1% s ⁻¹ , 2016 [46], 2018 [45].	23
Fig. II-10. Previous vacuum lighter-than-air structure designs.	28
Fig. II-11. Abaqus topology optimization and implicit analysis process diagram.	31
Fig. II-12. SF372 LLDPE film viscoelastic response at 4 MPa, 2004 [44].	32
Fig. II-13. Abaqus membrane elements, 2016 [59].	33

Fig. II-14. Abaqus topology optimization module life cycle, 2011 [74].	46
Fig. II-15. Optimal topology mesh dependence example, 2003 [73].	49
Fig. II-16. Shell bifurcation and snap-through examples [79,80].	50
Fig. III-1. Circular cylindrical shell dimensions diagram.	59
Fig. III-2. Model with imperfection from 17 buckling vectors and 0.1h amplitude.	61
Fig. III-3. Uniaxial stress vs strain representation of material models.	62
Fig. III-4. Mesh convergence study results.	66
Fig. III-5. Mesh convergence study process.	67
Fig. III-6. Equilibrium paths of elastic and elasto-plastic 1E-3 m thin shells.	69
Fig. III-7. Equilibrium paths of elastic and elasto-plastic 1E-4 m thin shells.	70
Fig. III-8. Equilibrium paths of elastic and elasto-plastic 1E-5 m thin shells.	70
Fig. III-9. Equilibrium paths of elastic and elasto-plastic 1E-6 m thin shells.	71
Fig. III-10. Equilibrium paths of an elastic 1E-3 m thin shell with two mesh sizes.	72
Fig. III-11. Uniaxial stress vs strain representation for two yield limits.	73
Fig. III-12. Equilibrium paths of an elastic 1E-3 m thin shell with two yield limits.	73
Fig. III-13. Fault location.	74
Fig. III-14. Collapsed shapes comparison for faulted models.	75
Fig. IV-1. Maximum material density required for buoyancy of spherical shell.	79
Fig. IV-2. Minimum material specific stiffness required for buoyancy of spherical shell.	81
Fig. IV-3. Example of minimum thickness required for buoyancy of coated spherical shell.	81
Fig. IV-4. Modeling representations for topology optimization.	82
Fig. IV-5. Sandwich shell topology optimization initial state and relative density plot.	86

Fig. IV-6. Solid sphere topology optimization relative density plot.....	87
Fig. IV-7. Geometrical development of semicircular section for topology optimization.	88
Fig. IV-8. Semicircular core section topology optimization relative density plots (showing elements with resulting relative densities at or above 0.6).	89
Fig. IV-9. Topology optimization findings driven design.	90
Fig. IV-10. Plane strain model diagram.....	92
Fig. IV-11. Pressure amplitude vs time applied on plane strain model.	92
Fig. IV-12. Plane strain model von Mises stress at four locations.	94
Fig. IV-13. Plane strain model displacement contour (in meters) at 2.004 seconds.....	96
Fig. IV-14. Plane strain model kinetic energy versus time.....	96
Fig. V-1. Geometric development of the Helical Sphere.....	99
Fig. V-2. Tube loading and dimensionality.	100
Fig. V-3. Hemispherical cut diagram.....	103
Fig. V-4. Tube diameter, tube length, weight-to-buoyancy ratio, and hoop stress vs number of revolutions.	105
Fig. V-5. Hydrostatic volume element [58].	108
Fig. V-6. Helical sphere contact diagram.	115
Fig. V-7. Axisymmetric representation of the Helical Sphere.....	117
Fig. V-8. Toroidal representation diagram ($n = 9$).	118
Fig. V-9. Weight-to-buoyancy ratio versus number of tori + 1.	120
Fig. V-10. Hoop stress versus number of tori + 1.....	121
Fig. V-11. Superimposed axisymmetric cuts of toroidal representation for $n = 9$ (black), 41 (red), and 161 (blue).....	122
Fig. V-12. Axisymmetric $n = 41$ model diagram.....	123

Fig. V-13. Smooth pressure load versus time.....	125
Fig. V-14. Von Mises stress contours at multiple loading magnitudes.	127
Fig. V-15. Cavities $T_0 - T_{20}$ absolute pressures versus applied pressure.	129
Fig. V-16. Cavities $T_0 - T_{20}$ volumes vs. applied pressure.....	130
Fig. V-17. Hemisphere visualization of von Mises stress contour at equilibrium state B.	131
Fig. V-18. Hemisphere visualization of von Mises stress contour at equilibrium state C.	132
Fig. V-19. Hemisphere visualization of von Mises stress contour at equilibrium state D.	132
Fig. V-20. Applied work, internal energy and kinetic energy versus applied pressure..	134
Fig. V-21. Kinetic to internal energies versus applied pressure.	134
Fig. V-22. Deformed shape at state C for weight-to-buoyancy estimate.....	136
Fig. VI-1. Diagram tracing the development of Icoron and its structural solutions as presented.	140
Fig. VI-2. Icosahedron and its subdivisions.....	141
Fig. VI-3. Icoron geometry development for frequency 4 (320 faces or unit cells).	142
Fig. VI-4. Two-dimensional representation of unit cell boundary and loading conditions	143
Fig. VI-5. Number of unit cells (blue) and unit cell radii (red) as a function of subdivision frequency.....	145
Fig. VI-6. Number of unit cells (blue) and number of unique unit cell radii (red) as a function of subdivision frequency.....	146
Fig. VI-7. Finite element model unit cell assembly.....	147

Fig. VI-8. Diagram of surface areas associated to radial component of pressure loading.	148
Fig. VI-9. Number of unit cells (blue) and radial components of internal-to-external loading area ratio (red).	150
Fig. VI-10. Number of unit cells (blue) and unit cell cavity-to-external pressure ratio (red).	151
Fig. VI-11. Weight-to-buoyancy (rigid) and spherical stress estimates.	154
Fig. VI-12. Unit cells air cavity and membrane total volumes as ratios of the displaced air mass across subdivision frequency.	156
Fig. VI-13. Unit cells air cavity and membrane total masses as ratios of the displaced air mass across subdivision frequency.	156
Fig. VI-14. Unit cell membrane mesh.	158
Fig. VI-15. Penalty contact enforcement diagram [61].	160
Fig. VI-16. Von Mises stress contours (left) and logarithmic maximum principal strain contours (right) overlaid for five loading states.	162
Fig. VI-17. Comparison of von Mises stress contours near a unit cell corner across loading states A-E.	164
Fig. VI-18. Comparison of von Mises stress contours near the unit cell corners at states A and B.	165
Fig. VI-19. Cavity pressure and area across the loading history.	166
Fig. VI-20. Energy distribution across load history.	169
Fig. VI-21. Weight-to-buoyancy estimate (deformed state) along the load history.	170
Fig. VI-22. Response surfaces representing the final deformed states as a function of modulus and preloaded cavity pressure.	171

List of Tables

	Page
Table II-1. Lighter-than-air structure design factors dependent on operational altitude. .	19
Table II-2. Polymeric films properties for lighter-than-air applications, 2004 [48].	27
Table II-3. Design responses supported by Abaqus general topology optimization, 2016 [59].	48
Table II-4. Finite element definitions used for solutions detailed in subsequent chapters.	56
Table III-1. Shell thicknesses considered.	68
Table IV-1. Specific stiffness and strength of materials considered [50,57,114].	80
Table IV-2. Axisymmetric model parameters for sandwich shell as initial representation.	85
Table IV-3. Axisymmetric model parameters for solid sphere as initial representation. .	87
Table IV-4. Three-dimensional model parameters for semicircular section as initial representation.	89
Table IV-5. Plane Strain Model Parameters	91
Table VI-1. Unit cell finite element model definitions.	159

DESIGN AND ANALYSIS OF AIR-STIFFENED VACUUM LIGHTER-THAN-AIR STRUCTURES

I. INTRODUCTION

“If a solid lighter than a fluid be forcibly immersed in it, the solid will be driven upwards by a force equal to the difference between its weight and the weight of the fluid displaced.”

Archimedes of Syracuse

LTA systems have been developed throughout history for a multitude of applications and have taken several forms and names. Airships, aerostats, blimps, and balloons are all part of this family of systems, which use Archimedes principle to achieve neutral and positive buoyancy in air by replacing an air volume with LTA gases, commonly known as *lifting gases*. Helium being the most common nowadays, these lifting gases are contained within an LTA system’s envelope (i.e., the outermost membrane acting as the barrier between the lifting gas and the atmosphere). These lifting gases stiffen the otherwise compliant structures, allowing them to sustain the pressure difference brought by the displaced air in order to achieve positive buoyancy. The compliance of these structures is a byproduct of the weight requirement, materials (and their air permeability), and the geometrical arrangement of which these structures are built from, resulting in dimensionalities that exhibit low or virtually non-existent in-plane bending stiffness [1].

The former has constrained the development of LTA systems that utilize an internal partial vacuum, rather than a lifting gas, to achieve positive buoyancy, where the envelope would be subjected to a pressure differential near atmospheric pressure. In other words, without a lifting gas to counteract the inward pressure exerted by the displaced air, a

vacuum lighter-than-air structure (LTAS) would rely solely on the structural configuration and material stiffness and strength to sustain such forces. Recent vacuum LTAS (VLTAS) research has therefore focused on using membrane-over-frame designs to provide the bending stiffness required to overcome the pressure differential. These efforts suggest that positively buoyant VLTAS are achievable with membrane-over-frame designs composed of thin hollow beams and membranes with stiffness moduli and strength requirements similar to those found in high end materials, such as graphene [2–6]. However, scaling such material properties and producing the required thicknesses has hindered the manufacturing of representative prototypes. In other words, the large density disparity between air and those materials exhibiting low air permeability often results in highly void VLTAS, with features containing dimensions in the order of 1/10,000 of their characteristic length [7]. VLTAS development is yet an attractive notion with the potential of yielding innovative structures and furthering LTA system utilization.

Two paths are envisioned to follow these efforts: {1} develop manufacturing processes that enable such complex designs; or {2} explore alternate designs that compensate the lack of material stiffness thru other means. The latter motivated the following question: can air be used to stiffen the otherwise gossamer envelope such that a LTAS achieves positive buoyancy without the use of an LTA gas? With the above in mind, this research aims at developing and characterizing *air-stiffened* designs as potential VLTAS, encompassing the following facets:

1. Geometrical characterization of air-stiffened designs
2. Development of structural simulation framework
3. Structural characterization of *air-stiffened* designs

1.1. Motivation

In the last three decades, several United States (U.S.) and international agencies have shown interest and/or supported the development of LTA systems for high-altitude, long-endurance sensorial applications [8–10]. The regained interest in these systems is primarily associated with their long endurance capability and potential economics, compared to current heavier-than-air (HTA) systems.

As sponsor of this research, the Air Force Office of Scientific Research supports innovative concepts that enable the Department of Defense (DoD) ability to counteract emerging threats. Design solutions of VLTAS with the potential for positive buoyancy in atmospheric air will not only be an innovative breakthrough, but more importantly, expand the usability of LTA systems. Additionally, the current depletion of Helium, a nonrenewable source and commonly-used lifting gas in LTAS [11,12], and the logistics associated with LTA gases, motivates the efforts of finding feasible VLTAS. Helium is in the List of Critical Minerals 2018, along with 34 other minerals the U.S. considers critical to our economy and defense [13]. Moreover, global demand has risen 100% over the past decade [12], where the U.S. accounted for 40% of the global consumption in 2010 [14].

1.2. Research Scope

The subject of VLTAS opens the door to many inquiries within air vehicles, including structures, aeroelasticity, flight dynamics, controls, and optimization, all underpinned by the capability of the structure to remain buoyant in quasi-static conditions. Such capability is dictated by its stability under the pressure differential as a result of displacing an amount of air weight larger than its own.

From a design perspective, the problem can be framed as follows: consider the weight of air displaced as a function of the system’s external geometry and operating altitude, given the standard atmosphere model [15]. Then consider the weight-to-buoyancy (W/B) ratio as a function of the air weight, air pressure loading, system’s weight and manufacturability. The requirement for neutral and positive buoyancy is a W/B ratio equal and less than one, respectively. Eq. 1 provides a generalized representation.

$$\frac{W}{B} = \frac{\sum V_c \rho_c}{V_a \rho_a} = f(\text{air weight, loading, envelope weight, manufacturability}) \leq 1 \quad (1)$$

where: V_c = system component volume V_a = displaced air volume
 ρ_c = system component density ρ_a = displaced air density

These four design factors are pictorially represented in Fig. I-1, with the W/B ratio at the center. Each factor is related to different design variables, such as geometry, altitude and atmosphere model in the case of displaced air weight.

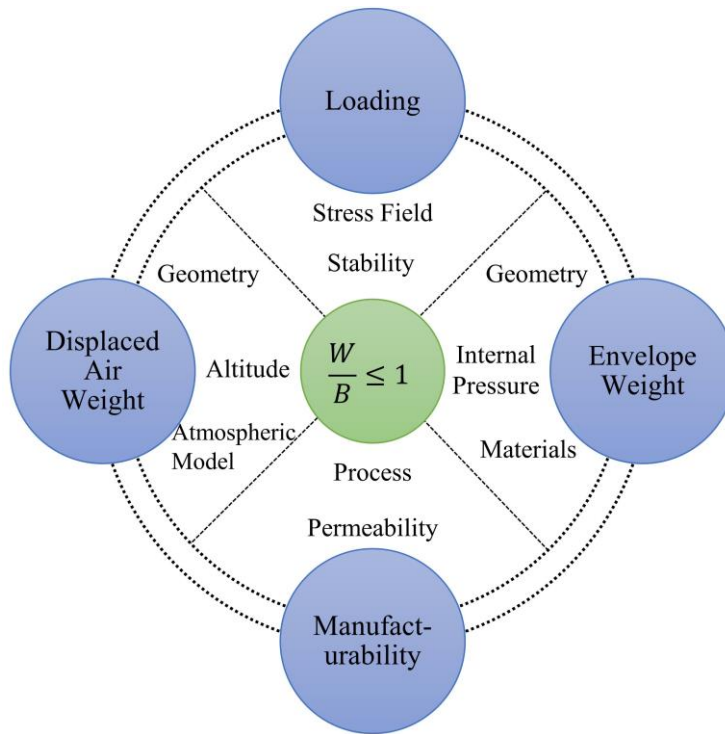


Fig. I-1. Depiction of the lighter-than-air structure design space.

For example, a specific structure and components therein dictate its weight and the volume of air displaced. At this point, the structure's existence implies manufacturability. A specific atmospheric location, or more generally an altitude, provides an air density and pressure, which dictates the weight of air volume displaced by the structure and the structural loading, respectively. It is important to note that structural responses are highly dependent on geometry and material. Out of these design factors, the development and characterization of *air-stiffened* designs presented in this dissertation focus on the {1} displaced air weight and loading resulting from sea level altitude air properties and {2} envelope weight given material properties associated with material known to be commercially available today and of prior use in LTA applications.

The terms *air-stiffened* designs and *air-stiffening* stated through this dissertation intend to represent a subset of designs that utilizes air to shape and serve as the core of a set of enclosing envelopes, as such providing structural support to the VLTAS upon loading. Specifically, the subset of designs considered are represented by two membranes enclosing a low-pressure region with pressurized air in-between and membrane-to-membrane connectivity. Fig. I-2 shows a generalized representation of such designs with the cross-section cut representing the connected membranes sandwiched between the air 'core'; the latter of which encloses a low-pressure region.

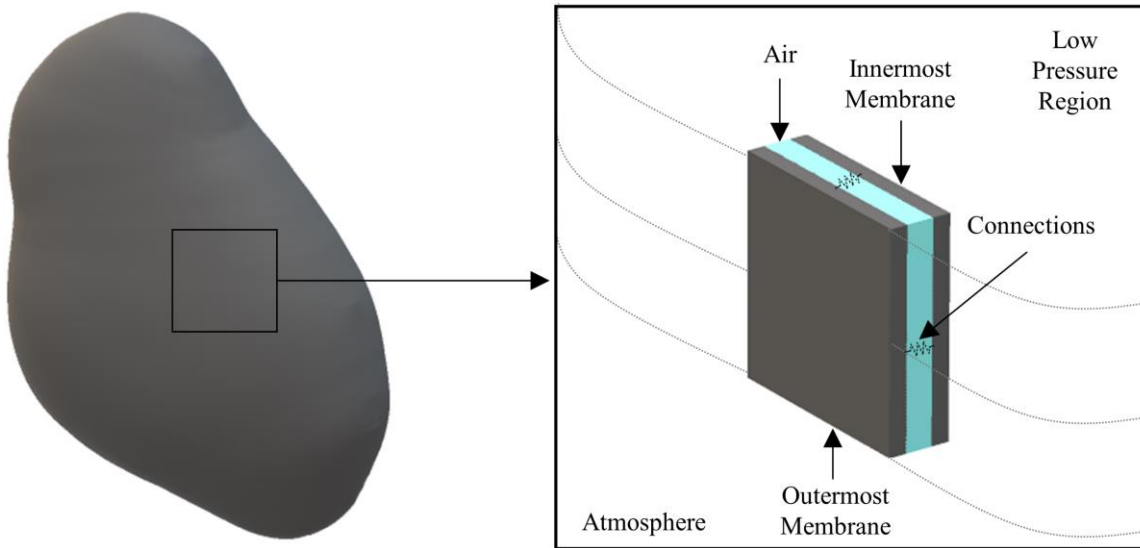


Fig. I-2. Generalized representation of air-stiffened designs.

Membrane-to-membrane connectivity is based on the premise that membrane stresses are reduced by adjusting the distance between connections such that the membrane post-loading local radius is small enough for the stresses away from the connections to fall below allowable stress levels. Connection-to-connection distance is premised on membrane stresses following Eq. 2, where r is the local membrane radius, h is the membrane thickness and ΔP is the differential pressure. Such stress control approach is the basis for the National Aeronautics and Space Administration’s (NASA) pumpkin-shaped super-pressure balloon design [16,17].

$$\sigma = \Delta P \left(\frac{r}{h} \right) \quad (2)$$

where: ΔP = pressure differential
 h = membrane, or film, thickness
 r = local radius

Two specific air-stiffened designs are considered in this dissertation, namely the Helical Sphere and Icoron discussed in Chapters V and VI, respectively. Representations of both designs are shown in Fig. I-3, where the Icoron can be seen as an incremental design

development that displays improved symmetry characteristics over the Helical Sphere. Both serve as geometric bases to study the characterization of air-stiffened VLTAS, including the utilization of membrane connectivity distance to target stress levels below material limits. The latter enables the consideration of material properties resembling commercially-available polymer films, specifically Polyethylene Terephthalate (PET).

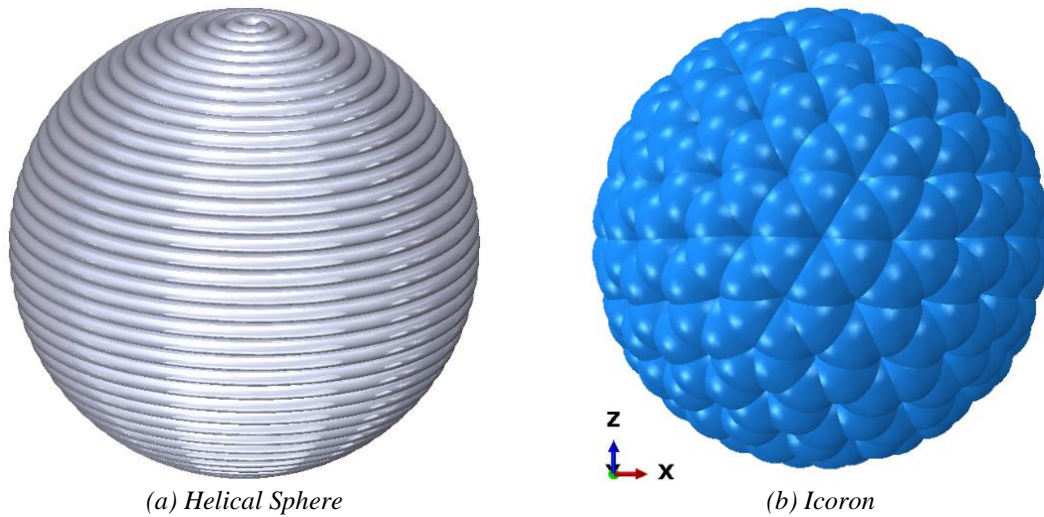


Fig. I-3. Representations of air-stiffened designs considered.

1.3. Dissertation Outline

This dissertation is organized following the trajectory that led to the development and characterization of air-stiffened designs. This trajectory is summarized in Fig. I-4, starting from the development of collapse solutions of the classical cylindrical shell of various thicknesses to the Icoron air-stiffened design proposed as an improvement over the Helical Sphere. Each phase served to progressively answer questions that produced the simulation methodology and structural characterizations.

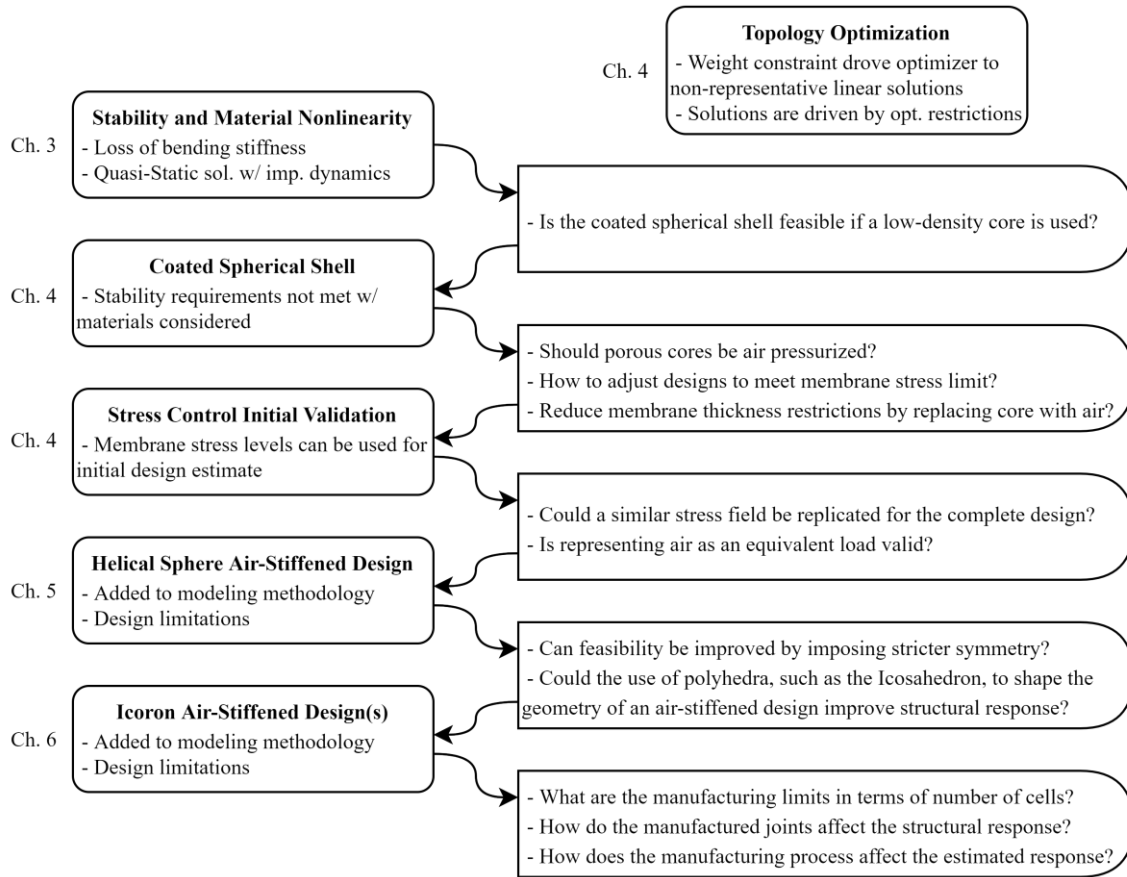


Fig. I-4. Air-stiffened design development trajectory summary.

This dissertation contributes the following towards the goal of developing feasible VLTAS:

- **Chapter II** provides background on LTA systems, including recent efforts involving the use of LTA gases and those efforts that focused on VLTAS potential. Furthermore, it presents structural modeling and simulation (M&S) methods and structural stability, which served as pillars to the simulation framework developed;
- **Chapter III** reports on collapse solutions of the circular cylindrical shell under a compressive force in order to characterize effects on the collapse path associated with increasingly small thicknesses, as well as shell-to-membrane transition;

- **Chapter IV** presents the consideration of a coated spherical shell, the use of topology optimization to search the solution space and the validation of the hoop, or circumferential, stress (Eq. 2) as a design parameter for stress control. Each of these design paths provided aspects that led to the consideration of air-stiffened designs and the simulation framework needed for such designs;
- **Chapter V** describes the Helical Sphere as an air-stiffened design, the simulation framework used to produce quasi-static structural solutions, and the simulation results that support the evaluation of its potential as a VLTAS;
- **Chapter VI** describes the Icoron as an air-stiffened design, detailing the modifications made to the simulation framework in order to characterize its structural response, and discussing its potential as VLTAS;
- **Chapter VII** summarizes the development of air-stiffened designs and the simulation framework developed to evaluate their potential as VLTAS. This chapter concludes with contributions and future work.

II. BACKGROUND AND RELATED WORK

Chapter I introduced the air-stiffened design concept and summarized the path that led to its development, stating the importance of structural characterization for VLTAS as a driver for the quantification of their potential as VLTAS. Such path was influenced by the background and related work presented in this chapter. As such, this chapter begins with the types, current use, and development of LTA systems to understand the basic principles behind LTA flight, as well as the motivation behind the regained interest. The development of NASA's balloons follows, contrasting the benefits and challenges of two balloon design types. Recent VLTAS research findings are then briefly discussed. Structural M&S techniques and procedures are described thereafter, which are used as foundation for the simulation framework shaped through Chapters III-VI. The chapter closes with a discussion on structural stability, and a brief surface stress discussion.

2.1. Lighter-than-Air Systems

Airships, aerostats, blimps, and balloons are all part of this family of systems, which uses Archimedes principle¹ to achieve neutral and positive buoyancy in air by replacing an air volume with LTA gases, notably helium, contained within the system's envelope². In the last three decades, many U.S. and international agencies have shown interest and/or supported the development of these systems for high-altitude, long-endurance sensorial applications, such as communications relay, weather forecasting, and intelligence,

¹ Archimedes principle states that a body is acted upon by a 'buoyant' force with direction opposite to gravity, and magnitude equal to the weight of the fluid displaced by and in which said body is submerged in. Said force is negative, neutral or positive if its magnitude is lesser, equal or larger than that of gravity [147].

² Envelope in this context, and throughout this dissertation, refers to the outermost membrane acting as the barrier between the lifting gas, or enclosed volume, and ambient air [10].

surveillance and reconnaissance (ISR) [8–10]. The regained interest on these systems is primarily associated with their long endurance capability and potential economics, compared to current HTA systems. Fig. II-1 shows how LTA systems can serve as a relay among satellite communications and employed systems, to include aircraft, ground, and personnel-carried systems.

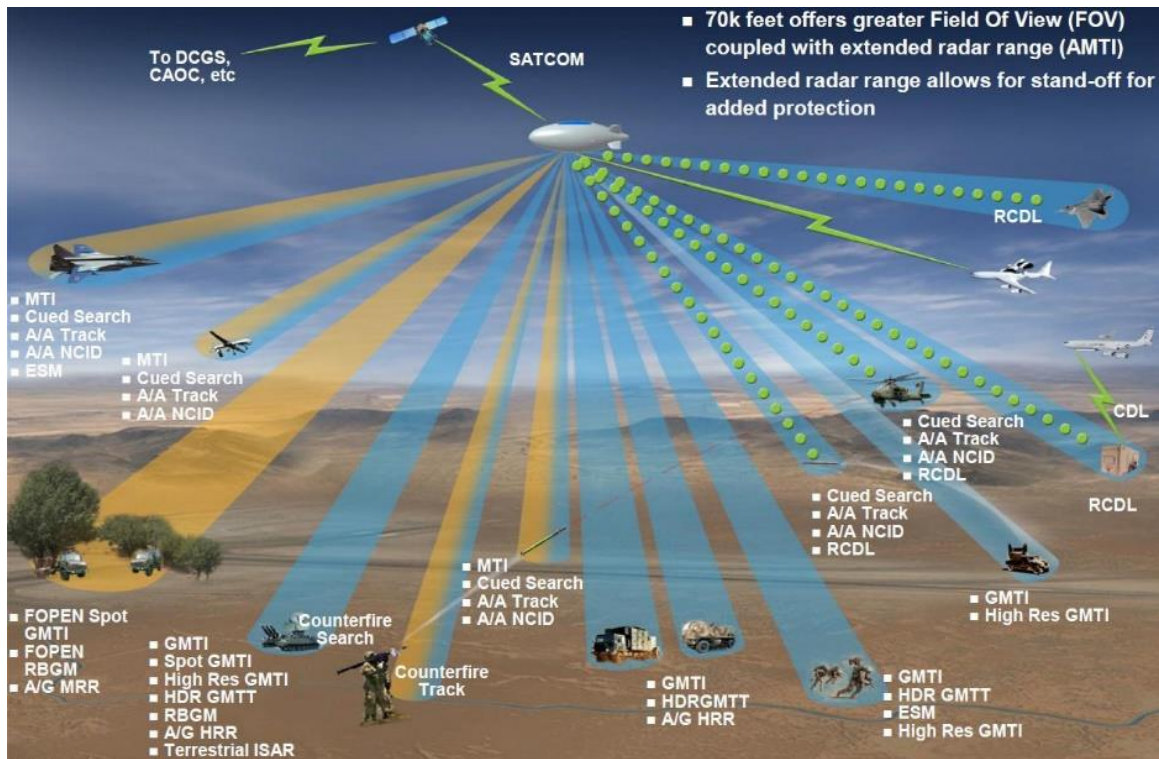


Fig. II-1. Example of lighter-than-air system applications for the intelligence, surveillance and reconnaissance mission set, 2012 [18].

LTA systems can be categorized as unpowered and powered, as shown in Fig. II-2. Unpowered systems, such as balloons (untethered) and aerostats (tethered), rely exclusively on buoyant forces for lift, and are typically comprised of an envelope and detachable payloads. Balloons tend to be free-floating with disposable envelopes, while aerostats operate from a fixed location with reusable envelopes. On the latter, the tether

acts as a continuous connection between the base and the system, providing power, altitude control, communications, etc. In contrast, powered systems, or airships, are aerodynamically-controlled vehicles, subcategorized into conventional and hybrid. Conventional systems rely exclusively on buoyant forces for lift, while hybrid systems supplement them with aerodynamic forces [10,18]. The VLTAS designs proposed here resemble the untethered balloons envelopes, which exclude (the weight associated with) support systems such as control surfaces and power.

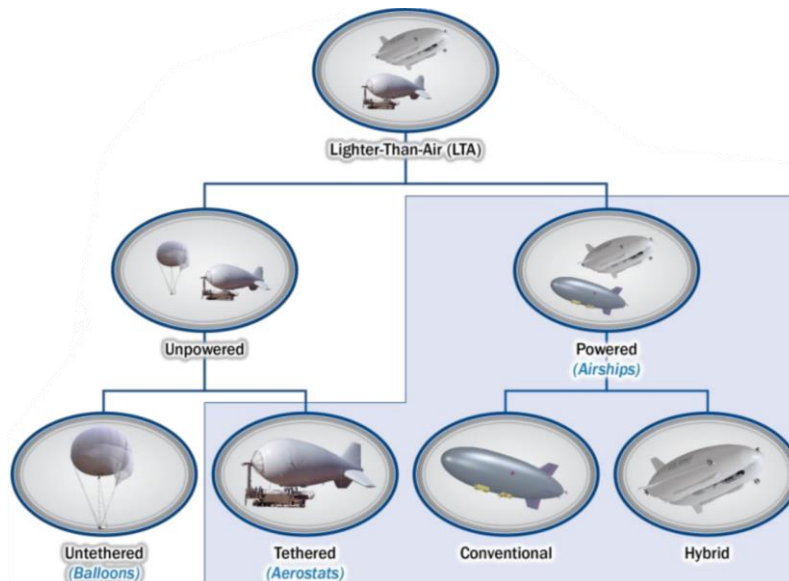


Fig. II-2. Lighter-than-air system categorization, 2012 [18].

Aerostats and airships are currently utilized or under development by, or in support of, the DoD and Department of Homeland Security [19]. As of 2012, the DoD was utilizing or actively developing nine aerostat and eight airship systems³. While many aerostats have been fully developed and operated, the Navy’s MZ-3A was the only fully developed airship then, which operated as a flying laboratory until 2017 [20]. These 17 systems were being

³ Refer to pages 13-60 of Ref. [18] for details on each of these 17 aerostat and airship systems.

developed to operate at altitudes outside 6-18 km (20,000-60,000 ft), with payloads between 8-1247 kg (18-2750 lbf), and endurances up to 30 days; 10 years for the Integrated Sensor Is Structure airship system being developed then by the Defense Advanced Research Projects Agency and U.S. Air Force [18]. In contrast, the MQ-9 Reaper unmanned air vehicle (UAV) has a 15 km (50,000 ft) ceiling, maximum takeoff of 4,763 kg (10,500 lbf), and an endurance of approximately 26 hours [21,22]. The development of these systems arises from the search of solutions that can extend the endurance of current UAVs, primarily for ISR missions.

Interestingly, a significant amount of insight into the design of VLTAS is gained from balloons. Arguably the simplest of LTAS categories from a system of systems perspective, high-altitude balloons are being developed and deployed for scientific and commercial purposes by organizations such as NASA and Loon LLC (traced back to a Google® project) [23,24]. The NASA efforts have been extensively documented in journals and NASA websites. Particularly, balloon envelope design, optimization, material characterization, modeling at multiple complexity levels, and testing have been published over the last two to three decades [16,23,33–35,25–32]. To build on this aspect of LTA systems, the thought process behind the operational altitude selection is discussed next, followed by a summary of the NASA's ultra-long duration balloons (ULDB) relevant findings, as well as recent VLTAS efforts. Takeaways important to the scope of this research are amassed lastly.

2.1.1. Operational Altitude Ranges

While long endurance enables the economics of LTA systems, size, weight, and speed bound the operational altitudes to ranges currently unexploited by HTA systems and space systems. From an aircraft dynamics point of view, the buoyancy-driven size and weight make *added mass* significant, and skin-friction the primary drag contributor [36].

As an object moves through a fluid, the fluid is forcibly displaced by the volume of the object since they cannot coexist in the same physical space simultaneously. The mass of fluid displaced adds to the object's inertia, hence *added mass*. From an energy perspective, added mass is the work done by the object to accelerate the fluid displaced. From an aerodynamic perspective, it can be seen as the drag due to fluid acceleration [36].

For example, consider a sphere moving through an inviscid and incompressible fluid. Using kinetic energy (KE), continuity, and work, Crowe, et al. [37]⁴ showed that the added mass force, F_{AM} , or force required to accelerate the fluid, can be represented by Eq. 3. Note that the force is dependent on the fluid's density, as well as the relative acceleration.

$$F_{AM} = \frac{\rho_f V_s}{2} \left(\frac{Dv_f}{Dt} - \frac{dv_s}{dt} \right) \quad (3)$$

where: ρ_f = density of the fluid v_s = velocity of the sphere
 v_f = velocity of the fluid $\frac{D}{Dt}$ = material derivative
 V_s = volume of the sphere

⁴ The authors defined F_{AM} as a function of the work rate required to change the kinetic energy of the fluid being displaced by the sphere. Given kinetic energy, $KE = \frac{1}{2} \rho_f \int_{V_s} v_f^2 dV_s = \dots = \frac{\rho_f V_s v_f^2}{4}$, the work rate is given by $v_f F_{AM} = \frac{dKE}{dt} = \frac{\rho_f V_s}{2} \left(v_f \frac{dv_f}{dt} \right)$, such that $F_{AM} = \frac{\rho_f V_s}{2} \left(\frac{dv_f}{dt} \right)$. The material derivative then comes into play when considering that the relative acceleration of the fluid with respect to the acceleration of the sphere can generally be described as $\dot{v}_f - \dot{v}_s$ ($:= \frac{dv_f}{dt}$), such that $\dot{v}_f = \frac{Dv_f}{Dt}$. The material derivative therefore describes the Eulerian velocity v_f in Lagrangian description. Further details found in section 4.3.4 of Ref. [37].

Consequently, the momentum equation for the sphere is expressed in Eq. 4. The term $\left(m_s + \frac{\rho_f V_s}{2}\right)$ represents the total mass of the systems, from a dynamic point of view. The term $\left(\frac{\rho_f V_s}{2} \frac{Dv_f}{Dt}\right)$ is the force contribution due to the added mass. In other words, the sphere moves as if it was carrying half of fluid displaced, $\frac{\rho_f V_s}{2}$, with a force contribution due to the resulting acceleration, $\frac{\rho_f V_s}{2} \frac{Dv_f}{Dt}$.

$$m_s \frac{dv_s}{dt} = \Sigma F + \frac{\rho_f V_s}{2} \left(\frac{Dv_f}{Dt} - \frac{dv_s}{dt} \right) \quad \rightarrow \quad \left(m_s + \frac{\rho_f V_s}{2} \right) \frac{dv_s}{dt} = \Sigma F + \frac{\rho_f V_s}{2} \frac{Dv_f}{Dt} \quad (4)$$

where: m_s = mass of the sphere

ΣF = summation of other force terms, such as gravity, drag, lift and Basset force⁵

Added mass is typically not considered in HTA system dynamics since it does not contribute significantly to the equations of motion. In contrast, added mass is significant in LTA system dynamics since its mass has the same order of magnitude as the air mass it displaces. This and their typically low speeds make mean wind speed and its deviations, such as gusts and turbulence, a significant component in the equations of motion [36].

The wind velocity field vs altitude and location can be described by mean values and time-dependent variations. As an example, Fig. II-3 shows the average wind speeds over Kabul, Afghanistan (left), and the seasonal wind speed over Baghdad, Iraq (right). The 6-18 km (20,000-60,000 ft) range is identified as inoperable for LTA system due to the high speeds encountered. Nonetheless, it is important to realize that there are seasonal deviations from the mean, where for example, winds remain below 21 m/s (40 knots or 46

⁵ Basset force results from the relative velocity between the object and the boundary layer of the fluid. In other words, it results from “the temporal delay in boundary layer development as the relative velocity changes with time” [37].

mph) for several altitudes up to 21 km (68,000 ft) during part of July above Baghdad [18,38]. These wind effects are one reason why recent aerostat and balloon programs have operational altitudes outside 6-18 km (20,000-60,000 ft). There are several incentives for untethered systems to have operational altitudes above 18 km (60,000 ft), such as: constituting unregulated airspace, remaining outside many air defense systems, and providing increasingly larger coverage areas and improved lines of sight [16,18]. However, high altitudes have its challenges.

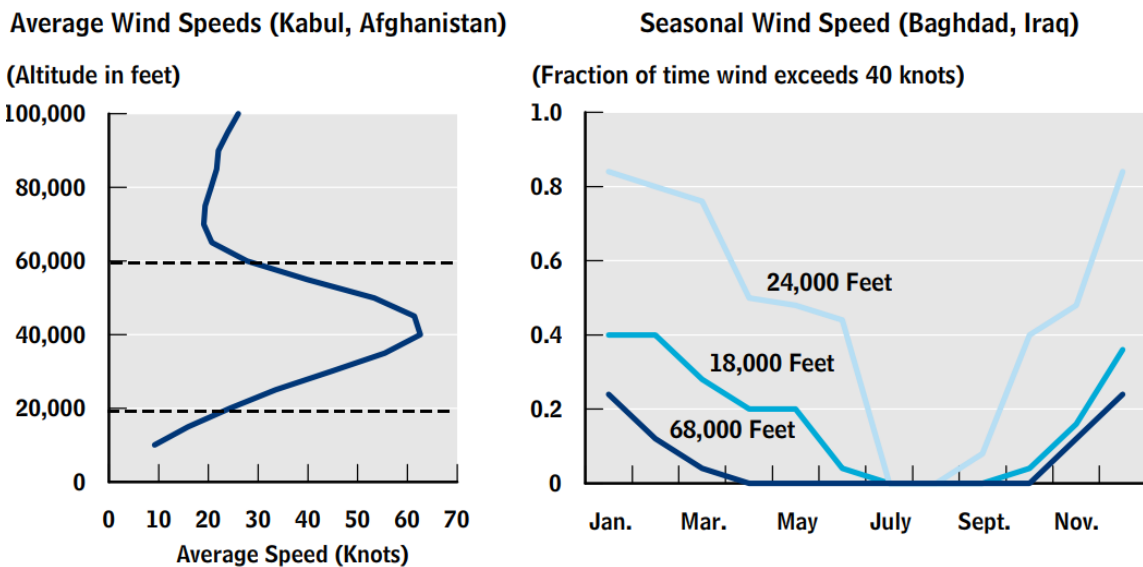


Fig. II-3. Average and seasonal wind speeds vs altitude and time of year over Kabul, Afghanistan and Baghdad, Iraq, 2011 [38].

It is commonly known that the atmospheric pressure, density, and temperature generally decrease as altitude increases. Fig. II-4a and Fig. II-4b show the geopotential altitude⁶ vs these, as well as speed of sound, up to 37 km (120,000 ft.). At 18.3 km (60,000 ft.), the pressure and density are 7.1% and 9.4% of sea level values, respectively.

⁶ The geopotential altitude is the atmospheric altitude measured from, and perpendicular to, the mean sea level; the latter loosely referred here as sea level [15].

The temperature moves from 15°C (59°F) down to -56.5°C (-69.7°F) between 11-20 km (36,000-65,600 ft.). The temperature (and ergo speed of sound) plateau and reversion above 10 km (Fig. II-4b) is associated with the absorption of sun ultraviolet radiation by the ozone layer; the later located within the stratosphere (formally the second atmospheric layer above the troposphere) [15].

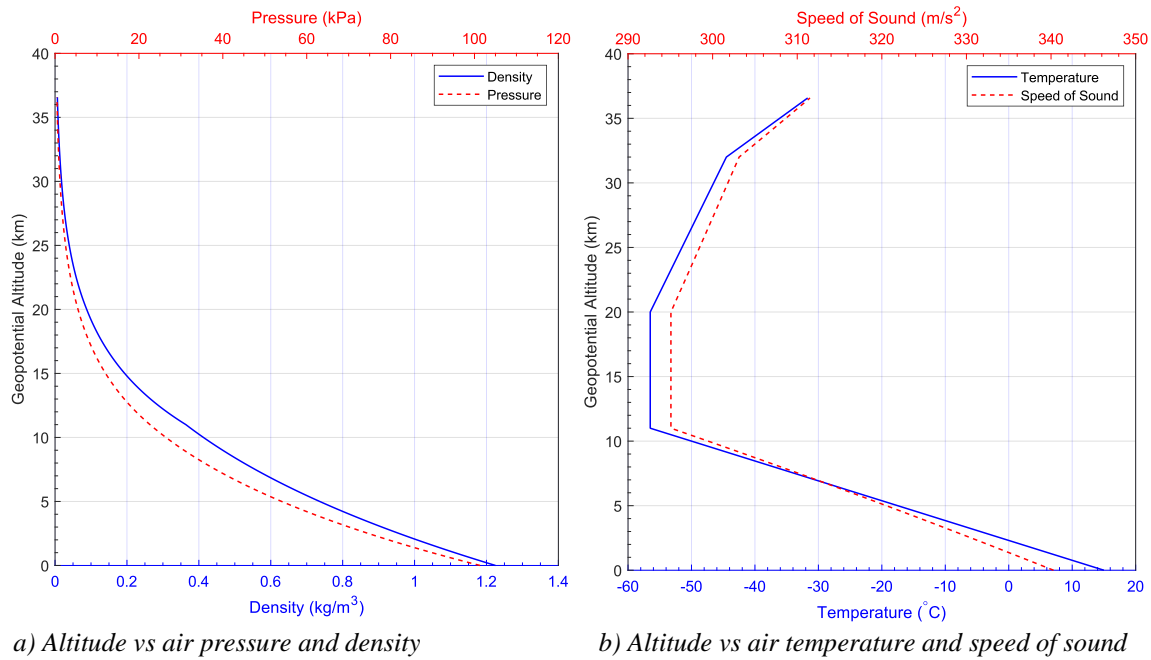


Fig. II-4. Air density, pressure, temperature and speed of sound as a function of altitude [15,39].

An LTA system, particularly the structural material thereof, needs to accommodate these drastic atmospheric changes as it moves from deployment to operational altitudes, and while pressure reductions alleviate loading, the resulting density and temperature changes constrain buoyancy and limit material selection. These pressure, density and temperature changes need to be considered during VLTA structure design, particularly when predicting W/B ratio and loading changes with altitude.

The tradeoff between altitude and buoyancy can be represented by the capability of an object to achieve neutral buoyancy. Eq. 5 expresses Newton’s second law on a spherical object with gross weight, W_s , displacing air mass $V_s\rho_a$. Solving for the spherical diameter results in Eq. 6. Fig. II-5 then shows the resulting diameters required for neutral buoyancy as altitude increases, where each curve represents a gross mass. Increases in diameter from those required at sea level are driven by the exponential changes in air density with altitude. This is yet another representation of the altitude effects in LTA applications.

$$W_s = V_s\rho_a g = \frac{4}{3}\pi r_s^3 \rho_a g \quad (5)$$

$$d_s = 2 \left(\frac{m_s}{\frac{4}{3}\pi\rho_a} \right)^{1/3} \quad (6)$$

where: ρ_a = altitude-dependent air density g = gravitational acceleration
 V_s = volume of sphere W_s = gross weight of sphere
 r_s, d_s = radius, diameter of sphere m_s = gross mass of sphere

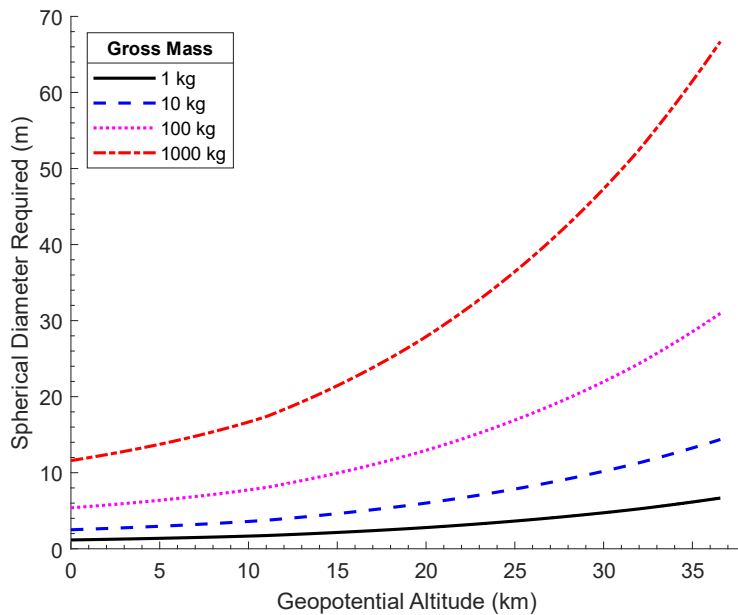


Fig. II-5. Spherical diameter required for neutral buoyancy vs altitude for four masses [15,39].

Therefore, operational altitude selection is a tradeoff among several LTAS design factors. Table III-1 summarizes the factors considered through this subsection. Low altitudes can result in smaller-sized systems with lesser temperature and environmental constraints, at the cost of potential higher material stiffness and strength requirements. On the other hand, high altitudes allow for operational benefits not previously available with low structural loading. At these high altitudes however, the low temperatures, high ozone concentration and intense ultraviolet light can cause materials, particularly polymeric, to become brittle, and lose strength and permeability⁷ due partly to the material reaching their embrittlement (i.e., glass transition temperature⁸) [10].

Table II-1. Lighter-than-air structure design factors dependent on operational altitude.

Factors ⁹ \Altitude	0 to 6,096 m (0 to 20,000 ft)	18,288+ m (60,000+ ft)
<i>Air Temperature</i>	15 to -24.6 °C (59 to -12.3 °F)	≥ -56.5°C (-69.7°F)
<i>Air Pressure</i>	101.3 to 46.6 kPa (14.7 to 6.8 psi)	≤ 7.2 kPa (1.04 psi)
<i>Air Density</i>	1.2 to 0.65 kg/m ³ (0.077 to 0.041 lbm/ft ³) (i.e., high weight-to-envelope size)	≤ 0.1153 kg/m ³ (0.0072 lbm/ft ³) (i.e., low weight-to-envelope size)
<i>Wind Speed</i>	Low, ≤ 12.9 m/s (28.9 mph)	Low, 10.3-15.4 m/s (23-34.4 mph)
<i>Added Mass</i>	Significant	Significant
<i>Others</i>	High probability of detection, regulated airspace.	Low probability of detection, unregulated airspace, larger sensorial coverage area, harsh stratospheric environment [10].

⁷ ASTM D4850-13 defines air permeability as “the rate of air flow passing perpendicular through a known area under a prescribed air pressure differential between the two surfaces of a material” [148]. Standard air density at sea level is 1.225 kg/m³, while density of polymers, carbon, and typical metals ranges between 800 kg/m³ and 7700 kg/m³ [149,150]. Air permeability is a function of material composition, manufacturing process and dimensionality [151].

⁸ Glass transition temperature, T_g , dictates when a material transitions from a glassy to a rubbery state; the latter being where the molecular freedom is enough to allow the material to sustain significant strain. Flexibility and toughness increase above T_g , while tensile strength and elastic modulus decrease [10].

⁹ Temperature and pressure follow Ref. [15,39]; wind speed follows averages on Fig. II-3.

2.1.2. Ultra-Long Duration Balloons

The NASA ULDB project, established in 1998, has been developing pumpkin-shaped super-pressure balloons (SPB) with the ultimate goal of lofting 900 kg (1,984 lbm) in instruments and developing technologies at near-constant altitudes greater than 33 km (108,268 ft) for 100 days [34,35,40]. Publications discussing SPB designs, modeling, materials and testing as part of the ULDB project show similarities with the VLTAS designs proposed in this dissertation.

Balloons are categorized as zero-pressure and super-pressure. *Zero-pressure* balloons (ZPB) maintain pressure equilibrium at the operating altitude via vent ducts that release *excess* gas to the atmosphere. Temperature changes through the daily cycles result in altitude cycles of as much as 9-15 km (30,000-50,000 ft). when ballast is not released. As they maintain *zero-pressure*, the envelope requires enough thickness to support the payload, and unsteady pressure differentials. On the other hand, SPB maintain a positive internal pressure differential, typically of 150-200 Pa (0.022-0.029 psi), by conserving the *excess* gas at the operating altitude, which allows near-constant altitudes during daily cycles at the expense of a higher strength requirement. At deployment, both are partially filled with enough helium for positive buoyancy, and gas expansion proceeds as it rises until it fills the envelope at the operating altitude, and pressurizes, in the case of the SPB. Fig. II-6 shows a pictorial representation of both designs. Fig. II-7 shows the inflation of a SPB at the Wanaka, New Zealand airport (left), and a fully deployed 532,200 m³ (18,794,466 ft³) SPB (right). The improved endurance and altitude stability, as well as the lack of ballast, have made the SPB the preferred choice for ULDB [40,41].



Fig. II-6. Zero-pressure and super-pressure balloons comparison [41].

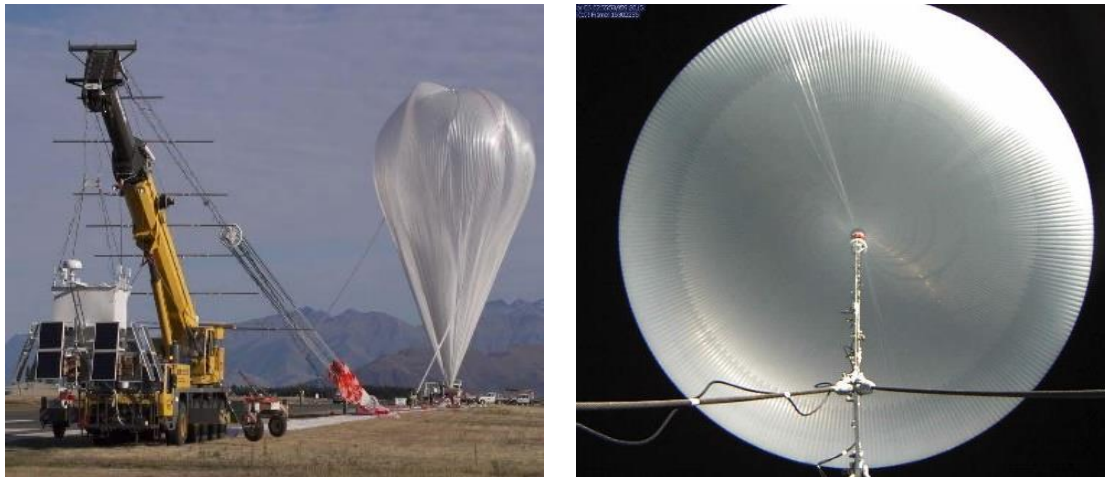


Fig. II-7. Super-pressure balloon inflation (left) and deployment, 2015 [40].

Even though other designs have been explored [42,43], the pumpkin-shaped design, shown in Fig. II-8, is the current choice for NASA's ULDB due to its ability to increasingly meet the high altitudes requirements [16]. The pumpkin-shaped SPB design is an oblate spheroid, where its height is approximately 60% of its diameter, made up of two

components: separate panels, or gores, running pole-to-pole and tendons running in between. The gores are lobed¹⁰ to reduce their local radius of curvature, significantly reducing the hoop stress (Eq. 2 on page 6)¹¹. The tendons sustain the longitudinal stresses. This reduction in hoop stress allows the use of film thicknesses and materials, such as linear low-density Polyethylene (LLDPE), that would otherwise yield at similar conditions [41].

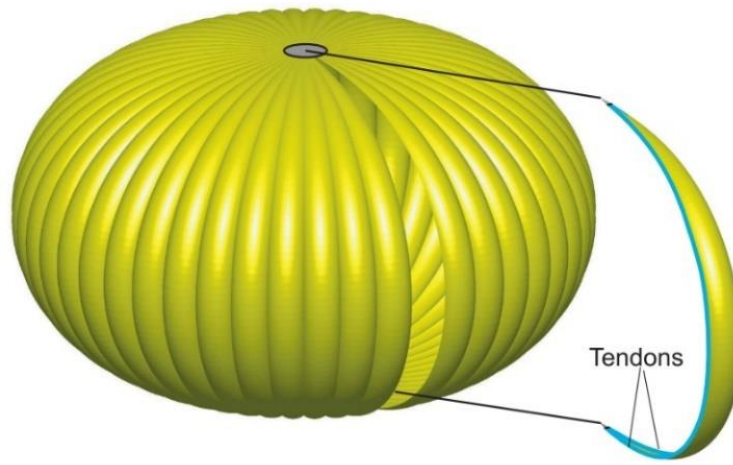


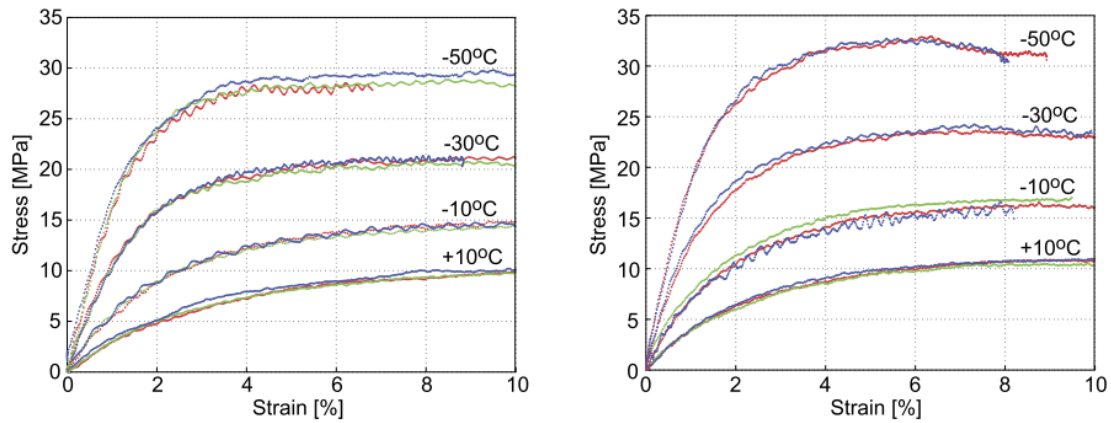
Fig. II-8. Pumpkin-shaped balloon rendition, 2007 [27].

NASA has utilized StratoFilm (SF) 372, 420 and 430 throughout the development of the pumpkin-shaped SPB. SF is produced for NASA by Charter NEX Films using Dow® DOWLEX™ LLDPE resin. The SF420 and SF430 films are 38 μm (1.5 mil) thick, three-layered, coextruded films with a 20/60/20 composition¹², where 60% of the center is LLDPE, and the rest is composed of an ultraviolet inhibitor (UVI) for radiation damage at high altitudes. Of the three, SF420 is further characterized in literature [41,44–46].

¹⁰ Each lobe is a membrane portion divided along the axis of revolution, with radii smaller than the spherical.
¹¹ Hoop stress refers to the circumferential stress historically describing small deformation observed circular cylindrical pressure vessels, and it is associated with the force perpendicular to the axis of symmetry and radius. It represents a component of the first Piola-Kirchhoff stress since the stress area follows the initial configuration as the reference configuration. Any ‘hoop stress’ in this dissertation follows this definition.

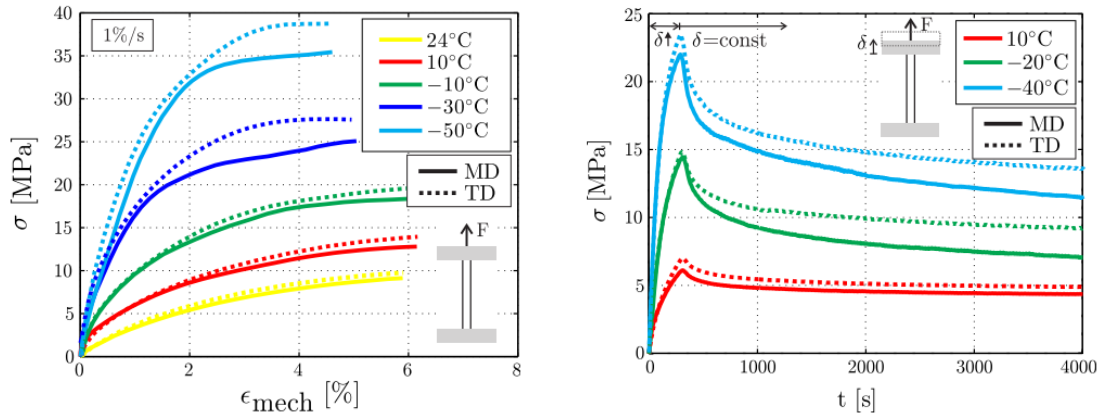
¹² SF430 data was found in Refs. [28,44,152]. While [28,44] report a 38 μm thickness, [152] reported 36.6 μm [13.2 (UVI) + 10.2 (LLDPE) + 13.2 (UVI), or 36/28/36 composition]. Furthermore, Ref. [44] reports that SF430 “is essentially the same as SF420, but without an additive for protection from ultraviolet light”.

NASA chose LLDPE because of its toughness and tear resistance at low temperatures (e.g., $T_g = -95^\circ\text{C}$ [-139°F]), as well as its low density (i.e., $900\text{-}1040\text{ kg/m}^3$ [$56.2\text{-}64.9\text{ lbm/ft}^3$]) and heat-sealing capacity [28,46–48]. Nonetheless, this material choice opened the door to viscoelastic, strain rate, anisotropy, temperature, and nonlinearity considerations, some of which are observed on the SF420 plots in Fig. II-9.



(a) Nominal stress vs engineering strain (ϵ) during uniaxial tension tests at $\dot{\epsilon} = 0.1\% s^{-1}$ in machine direction [46].

(b) Nominal stress vs engineering strain during uniaxial tension tests at $\dot{\epsilon} = 0.1\% s^{-1}$ in transverse direction [46].



(c) Nominal stress vs engineering strain during uniaxial tension tests at $\dot{\epsilon} = 1\% s^{-1}$ [45].

(d) Nominal stress vs time for relaxation tests; elongated at $\dot{\epsilon} = 1\% s^{-1}$ for 300 s, then constant for 3700 s at 3% engineering strain [45].

Fig. II-9. SF420 LLDPE film stress vs strain, time, temperature and direction at strain rates of $0.1\% s^{-1}$ and $1\% s^{-1}$, 2016 [46], 2018 [45].

The LLDPE film temperature dependence is observed in the machine direction (MD) (Fig. II-9a) and transverse direction (TD) (Fig. II-9b) at strain rates of $\dot{\epsilon} = 0.1\% s^{-1}$, where nonlinearity is prominent with decreasing temperature and yielding occurs at approximately 7% strain for 10°C and 3% strain for -50°C [46]. Strain rate dependence is deduced by comparing Fig. II-9a and Fig. II-9b with Fig. II-9c, where the latter shows tests at $\dot{\epsilon} = 1\% s^{-1}$ with increased film stresses across the strain range compared to $0.1\% s^{-1}$ [45]. Furthermore, film stress relaxation (Fig. II-9a) showed significant relaxation, predominant at lower temperatures; the latter is based on a film sample elongated at $\dot{\epsilon} = 1\% s^{-1}$ for 300 s and kept constant at 3% engineering strain for 3700 s [45]. Strain rate and temperature dependence are important considerations for whole flight simulations¹³ and required for accurate ULDB modeling, from low strain rates ($\dot{\epsilon} \leq 0.1\% s^{-1}$) during inflation up to day-night loading cycles at operational altitudes [45].

A SF420 material model found in literature, by Li et al. [46] as well as Bosi and Pellegrino [45], describes the LLDPE material through a *plane stress orthotropic nonlinear thermoviscoelastic model* with strain rate and thermal anisotropy effects. Plane stress is a simplification resulting from the membrane's thicknesses. Orthotropy is a result of the extrusion direction during the film blowing process. Nonlinearities capture dilatational and distortional (i.e., shearing effects) with large strains up to where non-recoverable deformation starts. The model is based on the Boltzmann superposition integral (Eq. 7), where nonlinearities are included via the time shift factor (based on the free volume model).

¹³ Whole flight simulations intend to represent structural and flight behaviors during LTAS flight; an aspect not evaluated and presented in this dissertation but relevant, as polymeric membranes are the primary target material considered.

$$\epsilon(t) = \int_0^{t'} D(t' - s) \frac{d\sigma}{ds} ds + \int_0^{t'} \alpha \frac{dT}{ds} ds \quad (7)$$

where: ϵ = strain vector σ = stress vector (one-dimensional case)
 t' = reduced time α = coefficients of thermal expansion vector
 T = temperature D = time-dependent creep compliance tensor

The Boltzmann superposition integral is a representation of the Boltzmann superposition principle using hereditary integrals. The Boltzmann superposition principle states that the strain output due to a combination of two arbitrary but different stress inputs applied at different times is equal to the sum of strain outputs resulting from each stress input applied separately (e.g., $\epsilon(\sigma_1 + \sigma_2) = \epsilon(\sigma_1) + \epsilon(\sigma_2)$) [49]. In other words, the material response at t (or t') due to individual load cases is the sum of the responses generated individually by each load case. Additionally, hereditary integrals are history-dependent integrals such that the resulting solution is dependent on the history of any prior input variable. Lastly, the concept of reduced time, t' , relates strain responses at different temperatures by shifting the timescale in which the stress response is integrated over, or vice versa. Reduced time is a reflection of the *time-temperature superposition principle*, which states that strain responses associated with a temperature change corresponds to a time shift. In practice, reduced time allows the natural timescale of a process to be shortened or extended while predicting the same material response.

Nonetheless, simpler models, such as the time-hardening creep power law shown in Eq. 8, have been used to characterize the LLDPE time-dependent response. These can provide an accurate material representation at specific stresses and temperatures. The appeal of the power law lies in its capability of being fitted with a relatively simple curve fitting process to find parameters A , m , and n . Many of the viscoelastic phenomena

observed in LLDPE are shared by a variety of polymer films, and are therefore presented above to expose some of the material modeling aspects required once the Hookean approximations used for such films demonstrate potential in VLTAS [44].

$$\dot{\epsilon}^{cr} = A\bar{\sigma}^n t^m \quad (8)$$

where: $\dot{\epsilon}^{cr} = \sqrt{\frac{2}{3}\dot{\epsilon}^{cr}:\dot{\epsilon}^{cr}}$ = uniaxial equivalent creep strain rate
 A, m, n = parameters fitted to creep data at nominal stress and temperature
 $\bar{\sigma}$ = uniaxial equivalent deviatoric stress
 $": "$ = double tensor inner product
 t = time

Sterling and Fairbrother [48] tested six films that were commercially available in thickness of 15 μm (0.6 mil) or less for potential use in ULDB and Gossamer space structures. They tested LLDPE, PET, PEN, polyimide, and polyamide for their modulus, yield¹⁴ strength, toughness, and tear propagation at 190 K (-83°C, -118°F) and 300 K (26°C, 80°F). Table II-2 shows the specific film brands tested and their results. Polyamide films testing was not completed as they became brittle at 213 K (-60°C, -76°F), which was above their anticipated operating temperature. PET and Polyethylene Naphthalate (PEN) had the highest modulus and yield strength, but poor toughness and tear propagation. The opposite can be said about LLDPE. All exhibited modulus and yield strength increases when going from 300 K to 190 K. PET, PEN and polyimide had densities in the 1140-1450 kg/m^3 (71.2-90.5 lbm/ft^3) range [50,51]. Interestingly, they note that several PET balloons intended for Mars exploration were damaged during inflation tests and aerial deployments. An important distinction between a SPB and a VLTAS is the magnitude of the pressure differential. While NASA's ULDB are subjected to pressure differentials up to 150-200 Pa

¹⁴ Yield stress, in this case, corresponds to the point where the stress-strain curve slope decreases below 20% of the maximum slope [48].

(0.022-0.029 psi), VLTAS are likely to be subjected to considerably higher values to achieve positive buoyancy.

Table II-2. Polymeric films properties for lighter-than-air applications, 2004 [48].

	Temperature Kelvin	Orientation	Units	Single-Layer Film Type					
				LLDPE Dowlex 2256	PET Mylar C	PEN Teonex Q51	Polyimide Apical AV	Polyamides Dartek C917 Dartek C101	
Thickness	300	—	μm	12.4	11.0	11.6	11.8	15.1	14.7
Weight	300	—	g/m ²	12.2	19.4	19.3	18.5	15.1	14.8
CB	—	—	K	172	<163	<163	<163	>213	>213
Impact Tear	300	—	N	8.8	0.6	0.6	0.2	1.8	2.1
	300	MD	N	3.7	0.1	0.1	0.1	0.2	0.2
	300	TD	N	3.3	0.1	0.1	0.1	0.2	0.2
Modulus	300	MD	GPa	0.2	4.5	5.6	2.8	—	—
	300	TD	GPa	0.2	4.8	5.8	2.9	—	—
	190	MD	GPa	1.2	6.6	9.4	4.7	—	—
	190	TD	GPa	1.3	8.8	9.5	5.2	—	—
Yield Stress	300	MD	MPa	7.1	110	140	106	—	—
	300	TD	MPa	7.4	98	141	111	—	—
	190	MD	MPa	42	207	278	141	—	—
	190	TD	MPa	47	190	257	152	—	—
Toughness	300	MD	MPa	57	127	105	82	—	—
	300	TD	MPa	93	36	62	78	—	—
	190	MD	MPa	>150	15	16	15	—	—
	190	TD	MPa	>120	19	12	13	—	—

2.1.3. Recent Vacuum Lighter-Than-Air Structure Efforts

With the intent of overcoming the stringent strength and stiffness requirements that a “perfect” spherical shell imposes on the realization of a VLTAS, efforts have focused on evaluating unconventional geometric configurations. These geometries trade manufacturing ease with geometric stiffness, structurally optimal dimensionalities, and properties found in high-end materials, such as carbon nanotubes (CNT), all in order to

theoretically achieve feasibility. These efforts have yielded valuable findings concerning the stability and limitations of such geometries. This subsection, then, intends to briefly describe them, focusing on those found germane to the path presented in this document.

Several geometric configurations were studied throughout the theoretical development of a VLTAS; the three most studied shown in Fig. II-10. The icosahedron was first proposed for this application by Metlen and Palazotto [2,52] with the intent of overcoming the instability and high strength required by the spherical thin shell.

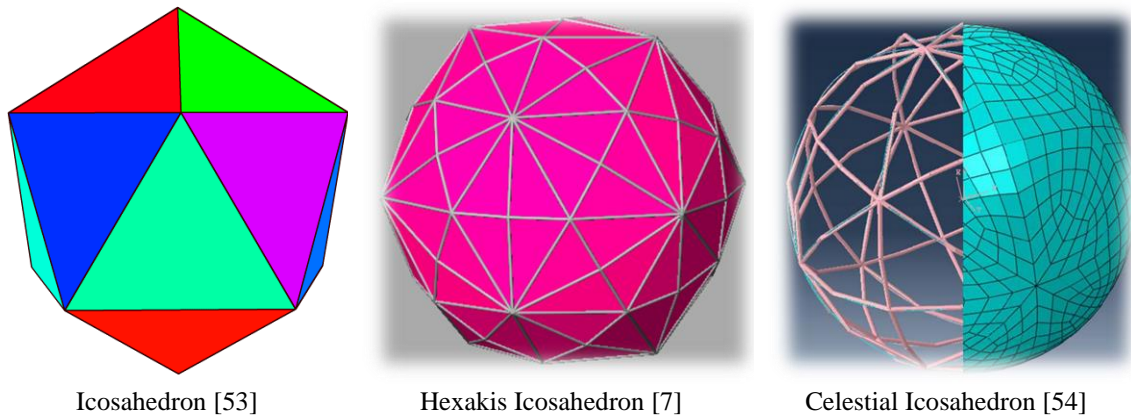


Fig. II-10. Previous vacuum lighter-than-air structure designs.

They posited that this geodesic frame of frequency¹⁵ one, composed of cylindrical beams with a membrane over it, could reduce the strength required and result in a feasible design. Their work and the nonlinear responses observed led to nonlinear quasi-static and dynamic analyses, as well as uniaxial tension and vibrational experimental tests of the frame, and dimensional studies of the full design. It was found, among many others, that {1} the static and dynamic stability is largely dependent on the frame's stiffness; {2} large membrane displacements do not significantly affect the predicted buoyancy; and {3} the

¹⁵ The icosahedron is a 20-sided polyhedron, also defined as a geodesic sphere of frequency one. Higher integer frequencies denote the number of times the sides, or beams in this case, are divided. The division points, referred as vertices, are placed in the circumscribing sphere along with previous vertices.

structural response is significantly affected by the beams thickness-to-radius ratio and the material properties of the membrane [7,53,55,56].

The icosahedron was followed by the hexakis icosahedron. From a structural perspective, the hexakis icosahedron takes advantage of the stiffness benefits seen in the icosahedron while reducing the strength requirement through shorter beams and hence smaller membrane sections. First proposed for this application by Cranston [2], and further analyzed by Schwemmer, Snyder, Castello, and Palazotto [3,57], the hexakis icosahedron showed a larger feasible design space over that of the icosahedron. Schwemmer proposed 1.2 m (4 ft)- and 4.6 m (15 ft)-diameter designs with membrane and beam thicknesses within 0.2-0.7 mm (8-28 mil), based on Hookean stiffness and strength approximations of CNT¹⁶, graphene¹⁷ and Spectra®¹⁸. Although both the icosahedron and the hexakis icosahedron exhibited chaotic behavior, the hexakis icosahedron's frame exhibited less chaos than the icosahedron's frame; attributed to the rigidity of the shorter beams.

The celestial icosahedron, first proposed by Cranston and studied by Moore [54] and Graves et al. [4], takes advantage of the stability that a frame provides as well as of the optimal surface area-to-volume ratio that a sphere provides¹⁹. They studied the quasi-static behavior as well as initial fluid-structure interactions via wind tunnel testing of additively manufactured models. They found buoyancy capability when a 0.85 μm (0.03 mil)-thick membrane is utilized. The challenge left ahead by these efforts is the development of

¹⁶ CNT approximations: modulus = 293 GPa, yield = 3.8 GPa, Poisson's ratio = 0.33 [114].

¹⁷ Graphene approximations: modulus = 500 GPa, yield = 50 GPa, Poisson's ratio = 0.1 [114].

¹⁸ Spectra® is an ultra-high molecular weight polyethylene fiber, typically used for their high strength [153]; approximations: modulus = 172 GPa, yield = 3.0 GPa, Poisson's ratio = 0.33 [114].

¹⁹ The sphere's surface area-to-volume ratio optimality is explained through the Isoperimetric Problem [154].

manufacturing processes that connect realized material properties with the Hookean assumptions made.

2.2. Structural Modeling and Simulation

Structural M&S is conducted using Abaqus, a general-purpose finite element (FE) package capable of modeling nonlinear structural problems considered static, quasi-static, and dynamic using both implicit and explicit integration techniques [58]. The review focuses on those techniques intended for the structural representation of LTAS subjected to differential pressures up to sea level pressure. Fig. II-11 shows the implicit and optimization processes that exemplify the solution methods used in this research. It is important to note that explicit solutions were also considered when convergence difficulties prevented a significant load history to be developed. Specific methodology is discussed in each chapter as associated to the solutions being presented.

2.2.1. Model Definitions

Starting on the top right side of Fig. II-11, model definitions are part of any preprocessing, and most of these are required for the three processes presented. Geometry is largely dependent on the specific study being conducted, but is associated with the general subset of stiffened designs presented in Chapter I. In terms of material definitions, there are several models that provide a useful representation. The first and simplest of all is the Hookean representation, shown in Eq. 9 [59]. Abaqus allows for material properties to be defined with anisotropic and temperature-dependent effects.

$$\boldsymbol{\sigma} = \mathbf{D}^{el} : \boldsymbol{\epsilon}^{el} \quad (9)$$

where: $\boldsymbol{\sigma}$ = total or Cauchy stress $\boldsymbol{\epsilon}^{el}$ = total elastic strain
 \mathbf{D}^{el} = fourth-order elasticity tensor

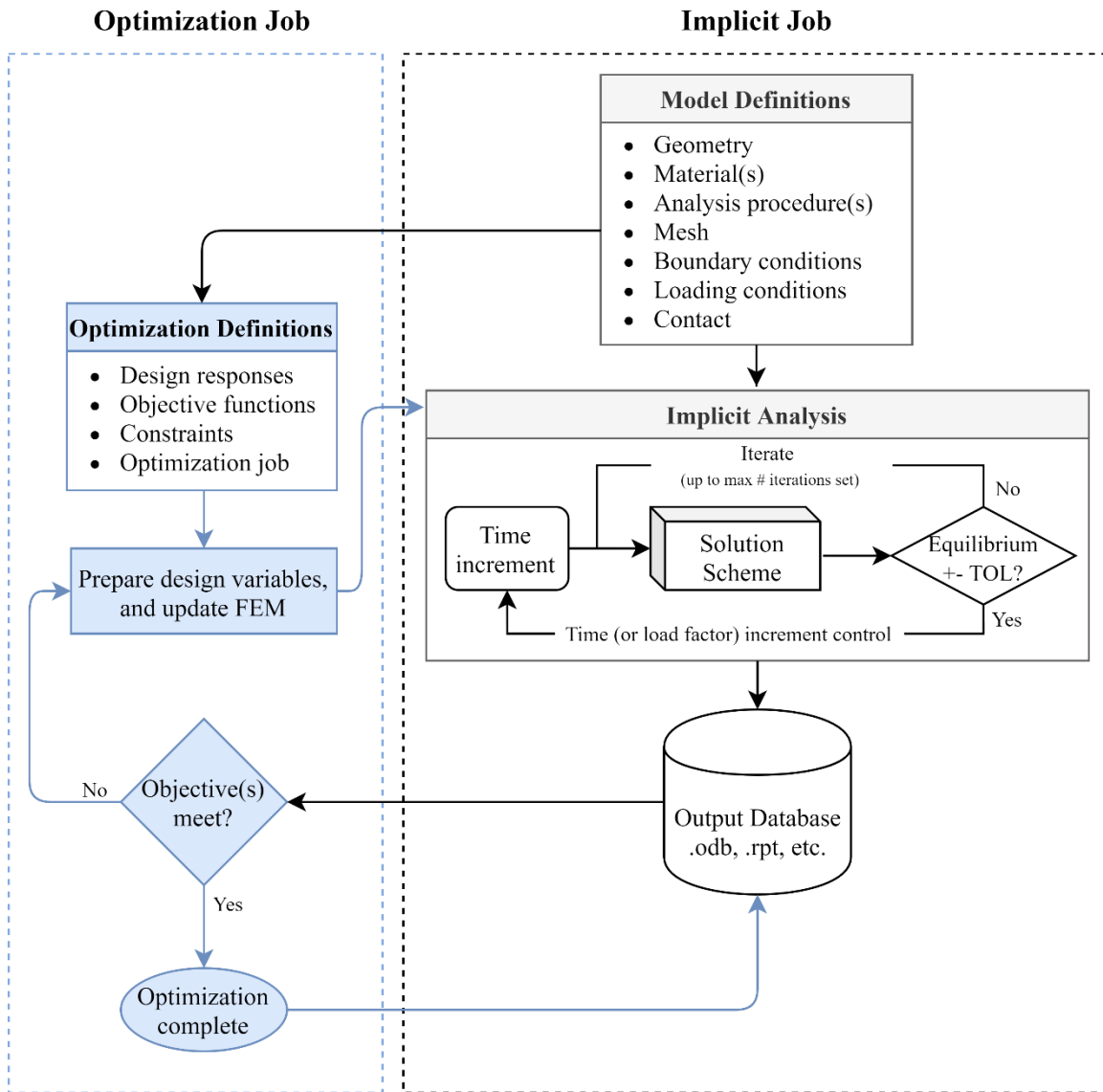


Fig. II-11. Abaqus topology optimization and implicit analysis process diagram.

Another model potentially useful for this application is the creep power law (Eq. 8 on page 26). Presented in the Abaqus documentation under the *CREEP routine, it was used by Gerngross et al. [44] to estimate the viscoelastic²⁰ response of pumpkin-shaped SPB lobes. To this effect, Fig. II-12 compares experimental results of SF372 LLDPE at

²⁰ While Ref [44] uses the *CREEP routine to represent a viscoelastic response (i.e., a rate-dependent elastic behavior), creep (i.e., strain accumulation at constant stress conditions over time) is typically observed in the inelastic range and associated with viscoplasticity (i.e., rate-dependent inelastic behavior) [49].

4 MPa with the Abaqus *CREEP²¹ and *VISCOELASTIC²² routines along with the Schapery–Rand model²³ [29]. They reported that *CREEP predicted strains up to 10% lower than the experiment.

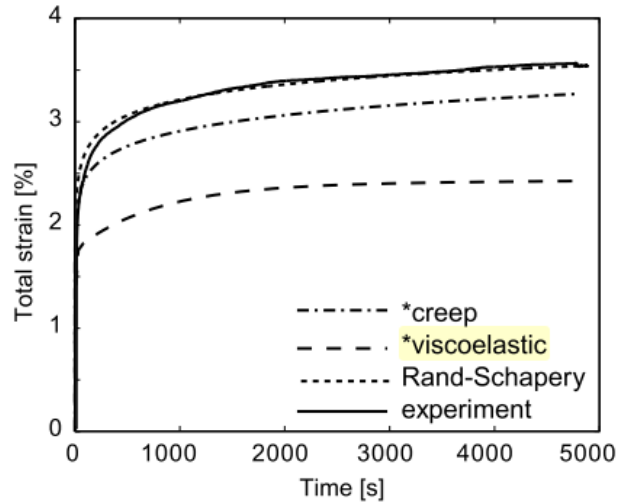


Fig. II-12. SF372 LLDPE film viscoelastic response at 4 MPa, 2004 [44].

Material models unavailable in Abaqus, such as the Schapery–Rand and the thermoviscoelastic models described by Li et al. [46], and Bosi and Pellegrino [45], are applied through the user-defined subroutine UMAT. The user-defined subroutine is called at every iteration of every increment for all material points associated with the material model, and provided with the increment size, Δt , material state (stress, temperature, etc.) and deformation gradients. UMAT then returns the new material state as well as the material Jacobian matrix. The accuracy of the Jacobian matrix depends on the integration

²¹ The *CREEP routine generally represents strain hardening, where its parameters are based on a constant stress condition representing equivalent uniaxial deviatoric stress states of elements in the FE model. As such, any stress-dependent variations of its parameters are not included across the time span of the solution, limiting its range [58].

²² The *VISCOELASTIC routine uses creep or relaxation (i.e., stress reduction at constant strain conditions over time) test data based on a linearized Schapery constitutive equation to represent linear viscoelastic behavior. Consequently, the latter neglects stress-dependent nonlinearities observed in LLDPE films [29].

²³ Schapery–Rand models nonlinear viscoelastic behavior by adding the elastic response to the transient response and capturing nonlinearities through stress- and temperature-dependent parameters [44].

process use inside UMAT, playing a key role in convergence during implicit time integration [28,44,59,60].

Another key model definition is the element selection. Two types of elements are primarily used in this research: membrane and solid, or their respective axisymmetric or plane strain representations. Membranes are thin sheets that carry tensile and compressive stresses parallel to the middle surface of the element, in a state of plane stress. In other words, they transmit in-plane forces with no bending stiffness. Abaqus offers membrane elements that can be used for two-dimensional (2D) and three-dimensional (3D) discretization, categorized as shown in Fig. II-13. General membrane elements include three- and six-node triangles, and four-, eight- and nine-node quadrilaterals. Reduced integration²⁴ is available on four- and eight-node quadrilaterals [58,59].

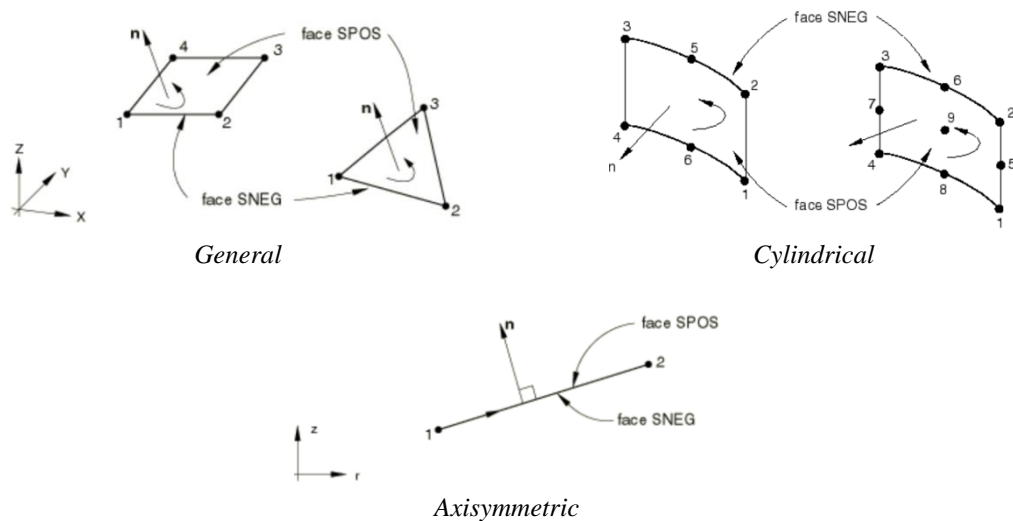


Fig. II-13. Abaqus membrane elements, 2016 [59].

²⁴ All Abaqus elements are integrated numerically (e.g., the virtual work integral is replaced with a summation, $\int_V \sigma : \delta D dV \rightarrow \sum_{i=1}^n \sigma_i : \delta D_i V_i$, where D is the deformation rate tensor, σ is the Cauchy's stress tensor, and n is the number of integration points). In quadrilateral and hexahedral elements, the integration points can be reduced such that the strain field is one order less than the interpolation order, called reduced integration. The zero-deformation modes that can result from reduced integration (i.e., hourglass modes) are counteracted by an artificial stiffness matrix that Abaqus adds to the reduced integration elements [58].

Similarly, Abaqus provides general, cylindrical and axisymmetric solid elements. General solid elements include isoparametric ‘brick’, triangular, tetrahedron, and wedge shapes. For efficient and accurate stress analysis of thin geometries subjected to bending, Abaqus recommends second-order (quadratic interpolation) elements to avoid: {1} shear-locking and constant (vs linear) strain fields of first-order triangular and tetrahedral elements, and {2} slow mesh convergence observed in first-order elements. Except for elements C3D27R and C3D27RH²⁵, second-order reduced integration brick elements generally avoid the volumetric and shear locking observed in fully integrated elements, while reducing computational time, and are therefore suggested for 3D solutions [58,59]. An exception is finite strains of second-order reduced integration brick elements, as they can lead to volumetric locking of near-incompressible materials; typically solved by mesh refinement in areas displaying finite strain [59]. Ultimately, the element types chosen depend on the intended analysis (e.g., 2D plane strain or axisymmetric, or 3D with membrane and/or solid elements), the computational time and mesh size (e.g., linear vs quadratic formulations), and whether implicit or explicit integration is used.

Another key model definition is contact surface interaction. Abaqus offers a variety of options to define surface pairings, interaction formulation, treatment of overclosures and thickness, and implementation or condition enforcement. The *general contact* algorithm provides the most robust capability often at a higher computational cost compared to specified contact pairs, while being somewhat restrictive when used in Abaqus/Explicit (e.g., kinematically enforced contact is only available using the contact pair algorithm). A

²⁵ Elements C3D27R and C3D27RH show three hourglass modes when all 27 nodes are unconstrained.

comprehensive description of the contact capabilities offered in Abaqus for both implicit and explicit analyses is presented in Chapter 7 of Ref. [61]. With respect to the research presented in this dissertation, general contact provided representative membrane-to-membrane interactions as deformation developed in the air-stiffened designs. Specific interaction definitions used are discussed in Chapters V and VI.

2.2.2. Implicit and Explicit Analyses

Solutions integrated implicitly, as exemplified by the implicit dynamic analysis process at the right center of Fig. II-11, are conducted in Abaqus under Abaqus/Standard. Abaqus/Standard uses the Newton's method to implicitly integrate the equations of motion to guarantee equilibrium within a tolerance (set by default to 0.5% of an average force in the structure, averaged over time [59]). On the other hand, solutions integrated explicitly are conducted in Abaqus under Abaqus/Explicit. Abaqus/Explicit uses the central difference integration rule to integrate the equations of motion utilizing diagonal lumped mass matrices for computational efficiency. The conditional stability of the central difference rule limits the time step size to the same order of the element size [62], which tends to increase the number of increments required to find the end state, compared to an implicit integration. However, the iterations required per increment to implicitly solve nonlinear equations can be computationally expensive and non-convergent at times. Both Abaqus/Standard and Abaqus/Explicit are used to find the solutions presented in this dissertation, prioritizing Abaqus/Standard as it {1} guarantees equilibrium within tolerance, {2} topology optimization is unavailable in Abaqus/Explicit, and {3} offers a wider range of mesh and material definitions. Nonetheless, lack of convergence in

Abaqus/Standard required the use of Abaqus/Explicit for the solutions presented in Chapter VI.

Geometric Nonlinearity and Strain Measures

Geometric nonlinearity, opposed to the linearity assumption, has a significant effect on large strain and rotation problems. This is simply observed thru the Green strain tensor, expressed in Eq. 10 [62]. The linearity assumption removes the quadratic components, $\frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j}$, undercutting strain and causing nonzero components in rigid body rotations.

$$E_{ij} = \frac{1}{2} (F_{ik}^T F_{kj} - \delta_{ij}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right) \quad (10)$$

where: E_{ij} = Green strain tensor components X = material coordinates
 $F_{ij} = \frac{\partial x_i}{\partial X_j}$ = deformation gradient x = Eulerian coordinates
 u_i = displacement δ_{ij} = Kronecker delta

Generally, constitutive definitions in Abaqus are formulated such that the stress and strain definitions form a conjugate pair, which allows the resulting strain energy to be a scalar-valued energy potential in the virtual work (“weak form”) statement of equilibrium. In other words, stress and strain (second-order) tensors are chosen such that their contraction becomes a scalar representing the internal energy of the system, formally referred to as being *conjugate in power* [62]. Such restriction, combined with the fact that some history-dependent constitutive expressions are strictly defined in strain and stress rates, leads Abaqus to identify “*the rate of Kirchhoff stress for use in constitutive equations as an appropriate stress measure for stress-sensitive materials*”; hyperelasticity is one of the exceptions where a total (rather than a rate) formulation is used [58].

Conjugate pairs, such as Kirchhoff stress and logarithmic strain tensors, replace the Cauchy stress²⁶ and deformation rate tensors (also conjugate pairs) in the virtual work statement of equilibrium, and their main purpose is simply to enable accurate and computationally-efficient formulations of constitutive models. Nonetheless, for some cases outside the scope of the problems considered in this dissertation, the accuracy of the latter statement has been challenged in terms of energy consistency violations in association to the stress and strain rates used to establish the rate form of the constitutive equations within the updated Lagrangian formulation²⁷ that Abaqus (and most commercially-available finite element (FE) codes) uses [63]. That is because there are structural cases in which the stress-strain rate pairs used by Abaqus (i.e., Jaumann and Green-Naghdi rates of the Kirchhoff stress [58]) combined with the constitutive definition can lead to large errors (e.g., buckling loads of thick orthotropic structures can be significantly overpredicted). Refs. [64–68] describe the different stress-strain formulations and exemplify where the errors lie, indicating errors can be avoided by switching to the Truesdell rate and its conjugate, the Green’s Lagrangian finite strain rate. These rare errors are associated with nonconjugate stress and strain increments, and become unimportant for thin-walled structures such as the ones considered in this dissertation [66]. It is relevant to note that Abaqus reports stress outputs as Cauchy stress regardless of the stress tensor used internally by the solver. Abaqus formulations and further details are found in Chapters 1 and 4 of Ref. [58].

²⁶ The Cauchy (“true”) stress is a symmetric tensor defined in the current configuration (i.e., deformed state, or in a spatial description) and represents actual traction on the material element. The Kirchhoff stress tensor is also described in the current configuration. Rate forms of the Kirchhoff stress and logarithmic strain tensors are used as they allow for the modeling of history dependent material behavior [62].

²⁷ An updated Lagrangian formulation refers a formulation where the variables are calculated with respect to the reference configuration, where, in incremental analyses, the latter is the state of the system at the previous increment, rather than the initial (user-defined) state.

Green strain is the default strain measure for small-strain formulations of shells and beam, in which small-strain but large rotations may be experienced. On the other hand, integrated strain (Eq. 11) is the default strain measure for most finite strain analyses in Abaqus/Standard. Logarithmic strain (Eq. 12) and nominal strain (Eq. 13) can be requested on geometrically nonlinear analyses within both Abaqus/Standard and Abaqus/Explicit. Logarithmic strain is the default on Abaqus/Explicit, and any hyperelastic analyses. Eqs. 11-13 are described from a material description and are detailed in Ref. [59].

$$\varepsilon^{n+1} = \Delta R \cdot \varepsilon^n \cdot \Delta R^T + \Delta \varepsilon \quad (11)$$

$$\varepsilon^N = \sqrt{F \cdot F^T} - I \quad (12)$$

$$\varepsilon^L = \ln \sqrt{F \cdot F^T} \quad (13)$$

where: ΔR = incremental rotation tensor (i.e., average rotation change over increment)

$\Delta \varepsilon = \int_{t^n}^{t^{n+1}} D dt$ = total strain increment, where D is the deformation rate

$\varepsilon^{n+1}, \varepsilon^n$ = total strain at increments $n + 1, n$

$\sqrt{F \cdot F^T}$ = left stretch tensor

Enabling geometric nonlinearity also enables thickness variation of shell and membrane elements by assuming plane stress conditions. The development is shown in Eq. 14 [58]. For incompressible materials (i.e., $\nu = 0.5$), the thickness becomes a linear function of the area ratio, $\frac{A}{A^0}$. The thickness remains constant for $\nu = 0$. This exponential dependency of thickness with element surface area becomes a significant contribution to the true stress in finite strain analyses as significant strain will reduce the elements' cross-sectional area.

$$\epsilon_{33} = -\frac{\nu}{1-\nu}(\epsilon_{11} + \epsilon_{22}) \rightarrow \ln\left(\frac{t}{t_0}\right) = -\frac{\nu}{1-\nu}\left[\ln\left(\frac{l_1}{l_1^0}\right) + \ln\left(\frac{l_2}{l_2^0}\right)\right] = -\frac{\nu}{1-\nu}\ln\left(\frac{A}{A^0}\right) \quad (14)$$

$$\rightarrow \boxed{t = t_0 \left(\frac{A}{A^0}\right)^{-\frac{\nu}{1-\nu}}}$$

where: ν = Poisson ratio t_0, t = initial, final thickness
 ϵ_{33} = thru-the-thickness strain l_i^0, l_i = initial, final length
 $\epsilon_{11}, \epsilon_{22}$ = strains in directions 1,2 A^0, A = initial, final area

Convergence of Nonlinear Solutions

Lack of convergence of static and quasi-static problems solved implicitly is largely associated with mesh distortion, global instabilities, and/or local instabilities. Mesh distortion is typically observed in solutions subjected to large strains and/or material loss. Several meshing techniques can minimize this, to include user-defined mesh definitions and convergence as well as adaptivity techniques. Global stabilities (i.e., load-displacement response with negative stiffness) can be representative of buckling and collapse, in which case displacement-controlled or Riks procedures can be employed; the latter is detailed in section 6.2.4 of Ref. [59]. Local instabilities, on the other hand, result in transfers of strain energy from the instability region to its neighboring elements. These problems can be solved either dynamically, which is discussed in the next subsection, or statically by using artificial damping to stabilize the solution. Abaqus provides *adaptive automatic stabilization* within its nonlinear static procedures. When the solution becomes unstable, *automatic stabilization* adds viscous forces to the equilibrium equations of the form described in Eq. 15, where the damping factor, c , is user-defined or calculated by Abaqus [59].

$$P - I - F_v = 0 \rightarrow P - I - cM^*v = 0 \quad (15)$$

where: P = external force vector c = damping factor
 I = internal force vector M^* = unity-density artificial mass matrix
 F_v = viscous force vector v = $\frac{\Delta u}{\Delta t}$ = nodal forces vector / time incr.

Abaqus calculates the damping factor based on a damping energy to strain energy ratio, referred to as *dissipated energy fraction* and defaulted to 2E-4, and assuming that the solution is stable at the first increment. *Automatic stabilization* by itself applies damping constantly throughout the model. *Adaptive automatic stabilization* tries to reduce its effect on the solution while preserving convergence by varying the damping factor both spatially and incrementally based on the convergence history and dissipated energy fraction. The dissipated energy fraction is then subject to a global accuracy tolerance, defaulted to 0.05, to ensure that inaccuracies resulting from incremental energy calculations do not produce unnecessary damping [59].

Implicit Dynamic Solutions

The Abaqus implicit dynamic procedure within Abaqus/Standard uses Hilber-Hughes-Taylor (HHT) and backward Euler methods to setup numerical integration frameworks, allowing the Newton's method to iterate until convergence at a time step. Both integration methods provide numerical damping to the solution of various degree; backward Euler providing the most. The implementation of the HHT into the equilibrium equation is shown in Eqs. 16-19. Eqs. 16 and 17 show the continuous and discrete representations of the equilibrium equation, respectively. Eqs. 18 and 19 show the displacement and acceleration, respectively.

$$M^{NM}\ddot{u}^M + I^N - P^N = 0 \quad (16)$$

$$M^{NM}\ddot{u}^M|_{t+\Delta t} + (1 + \alpha)(I^N|_{t+\Delta t} - P^N|_{t+\Delta t}) - \alpha(I^N|_t - P^N|_t) + L^N|_{t+\Delta t} = 0 \quad (17)$$

$$u|_{t+\Delta t} = u|_t + \Delta t\dot{u}|_t + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{u}|_t + \beta \ddot{u}|_{t+\Delta t} \right] \quad (18)$$

$$\dot{u}|_{t+\Delta t} = \dot{u}|_t + \Delta t[(1 - \gamma)\ddot{u}|_t + \gamma\ddot{u}|_{t+\Delta t}] \quad (19)$$

where: M^{NM}	= the mass matrix	N, M	= degrees of freedom
I^N	= internal force vector	$-\frac{1}{2} \leq \alpha \leq 0$	
P^N	= external force vector	$\beta = \frac{1}{4}(1 - \alpha)^2 > 0$	
L^N	= Lagrange multiplier forces sum	$\gamma = \frac{1}{2} - \alpha \geq \frac{1}{2}$	
t	= time	\dot{u}	= velocity
u	= displacement	\ddot{u}	= acceleration

The HHT method extends Newmark's method by adding the parameter α to Eq. 17, such that the numerical integration achieves second-order accuracy while allowing numerical damping. This is based on the fact that the Newmark's method either achieves second-order accuracy by becoming the trapezoidal rule at $\beta = 1/2$ and $\gamma = 1/4$ (which does not have numerical damping), or adds numerical damping outside these β and γ values [69]. Eqs. 18 and 19 are identical for both Newmark's and HHT methods. Note that an $\alpha = 0$ results in zero damping, reducing to the trapezoidal rule, while an $\alpha = -1/3$ results in maximum damping (e.g., 6% damping ratio when the time increment is 40% of the oscillation period of the mode considered).

Abaqus presents the user with three pre-set options: transient fidelity, moderate dissipation and quasi-static. Transient fidelity uses HHT, with an $\alpha = -0.05$, which results in minimal dissipation and ergo small time increments for maximum accuracy and convergence. Moderate dissipation also uses HHT, with an $\alpha = -0.41421$, and is recommended for dynamic events with nonlinearities, contact, and/or significant amount of energy dissipated through viscous means, such as plasticity and damping. Quasi-static

uses backward Euler, which has larger numerical damping than HHT, with the intent of modeling problems where inertia effects are present to regularize instabilities [58,59,70].

Explicit Dynamic Solutions

Abaqus/Explicit provides a platform for solving quasi-static and dynamic FE problems via the use of wave propagation analysis. The explicit procedure propagates the loads throughout the structure, using dynamic equilibrium and the conditionally stable central difference integration method to find nodal accelerations, and updates the velocities and displacements as the forces are applied. Specifically, rearranging the dynamic equilibrium stated in Eq. 16, the acceleration (Eq. 20) at the current time step is obtained using the diagonalized mass matrix. The velocity at half-step is then obtained (Eq. 21), followed by the displacement at the next time step (Eq. 22).

$$\ddot{u}^N|_t = [M^{NM}]^{-1}(P^M - I^M)|_t \quad (20)$$

$$\dot{u}^N|_{t+\frac{\Delta t}{2}} = \dot{u}^N|_{t-\frac{\Delta t}{2}} + \left(\frac{\Delta t|_{t+\Delta t} + \Delta t|_t}{2}\right)\ddot{u}^N|_t \quad (21)$$

$$u^N|_{t+\Delta t} = u^N|_t + \Delta t|_{t+\Delta t} \left(\dot{u}^N|_{t+\frac{\Delta t}{2}}\right) \quad (22)$$

where: M^{NM} = diagonalized mass matrix	u = displacement	t = time
I^N = internal force vector	\dot{u} = velocity	
P^N = external force vector	\ddot{u} = acceleration	
L^N = Lagrange multiplier forces sum	N, M = degrees of freedom	

The above explicit formulation allows for a large number of increments to occur efficiently (i.e., each time increment is computationally inexpensive compared to implicit integration) since the simultaneous solution of equations is not required, removing the need for iterations. This is partly attributed to the diagonalized, or lumped, mass matrix, which its inversion only requires $O(N)$ operations [71]. It is also attributed to the half-step

velocity increments, which allow the inclusion of stiffness-proportional damping without requiring the simultaneous solution of equations.

These benefits come at the cost of conditional stability through a maximum stable time increment. The stability limit is generally defined in terms of the highest frequency of the model, but, for practical purposes, Abaqus defines it (Eq. 23) as the minimum characteristic element length and material dilatational wave speed ratio (Eq. 24) for elastic materials, ignoring damping.

$$\Delta t_{max} \approx \frac{L_{min}}{c_d} \quad (23)$$

$$c_d = \sqrt{E/\rho} \quad (24)$$

where: L_{min} = smallest element dimension E = elastic modulus
 c_d = dilatational wave speed ratio ρ = material density

Actual time increments during Abaqus-controlled time incrementation are less than the stability limit by a factor between $1/\sqrt{3}$ and 1, accounting for stiffness associated with contact. Abaqus-controlled incrementation serves well for problems in which the total energy balance will change significantly, such as models with large nonlinear deformations and/or nonlinear material response. Furthermore, material, mesh, and other elements will directly influence stability. The material model itself will affect the stability through its relationship with the dilatational wave speed (e.g., a linear material has a constant dilatational wave speed, but a plastic material exhibits a variable wave speed with yielding and stiffness). Furthermore, the mesh will affect stability via the shortest element dimension, making a course mesh beneficial for stability (not necessarily for accuracy). Lastly, spring and dashpot elements have shown to become unstable during the course of

some analyses, and it is typically characterized by unbounded and nonphysical results, and by oscillatory solutions.

From a model accuracy perspective, there are other limitations when using Abaqus/Explicit. First-order, reduced-integration elements are typically used, which can deteriorate the solution with hour glassing if not properly controlled, and may require a denser mesh to achieve similar results to those found with second-order elements in Abaqus/Standard. From a computational cost perspective, deciding between using Abaqus/Standard or Abaqus/Explicit becomes a function of the model size as the cost of implicit solutions is comparatively low in Abaqus/Standard, but rises sharply with model size. The efficiency of Abaqus/Standard over Abaqus/Explicit is related to the number of iterations needed to achieve equilibrium in each time (load) step with potentially large time increments versus significantly smaller, but relatively inexpensive, time increments, respectively. Overall, Abaqus/Explicit excels at high-speed, short-duration dynamic events, large nonlinear quasi-static analyses, highly discontinuous post-buckling and collapse analyses, coupled temperature-displacement analyses, and structural-acoustics analyses. Therefore, the use of Abaqus/Explicit over Standard relies on the smoothness of the solution and/or the lack of convergence on large nonlinear static problems in Abaqus/Standard [58,59,72].

2.2.3. Structural Optimization

Structural optimization was introduced into the development trajectory (summarized in Fig. I-4) as a method of challenging current air-stiffened design ideas. Specifically, it was introduced to answer whether structural optimization was able to

provide a promising replacement to Helical Sphere air-stiffened design or could inform its development. With this in mind, the left side of Fig. II-11 shows the overarching Abaqus structural optimization process used during this research. Abaqus provides four types of structural optimization: topology, shape, sizing, and bead. Topology optimization essentially removes volume (elements) to achieve a particular objective, most commonly minimize strain energy. Shape optimization moves nodes typically to alleviate stresses concentrations. Sizing optimization adjusts shell thicknesses, element by element, typically to maximize stiffness-to-weight ratios. Bead optimization adds stiffening beads to a shell structure, to increase its moment of inertia, with the intent of increasing its stiffness and eigenfrequencies. All can follow the following optimization problem framework²⁸ [59], stated in Eq. 25: minimize the objective function f , built from the summation of the weighted, W_i , differences between design responses, φ_i , and reference values, φ_i^{ref} , subject to design responses, g_i , constrained by g_i^* , and expressions based on design variables, k_i , constrained by k_i^* . The responses are single scalar values, calculated by the optimization module from the output database (ODB) and model data, based on two state variables, U , operators within a region: {1} min or max value; and {2} sum of all values.

²⁸ Abaqus also allows maximization of the same objective function stated in equation (25), and the minmax problem of weight design responses [59].

$$f_{min} = \min[f(U(x), x)] = \min \left[\sum_{i=1}^N W_i (\varphi_i - \varphi_i^{ref}) \right] \quad (25)$$

subject to: $g_i(U(x), x) - g_i^* \leq 0$ and $k_i(x) - k_i^* \leq 0$

where: f = objective function
 U = model state variables
 N = number of design responses
 W_i = weight given to each design response
 g_i = design response constrained by g_i^*
 k_i = constraining expression in terms of x , constrained by k_i^*

φ_i = design response
 φ_i^{ref} = reference value
 x = design variables

Topology optimization, which is the one used during this research, is a material distribution method for finding an optimum (nonunique) layout. The optimization question can be framed as follows: given a topology, analysis procedure, boundary conditions (BC), loads, and constraints, which material distribution produces an optimized objective? The topology is represented by a discretization of the design domain, which parametrizes the stiffness tensor [73]. The topology is then optimized by scaling the relative material density(ies) within the design domain in an iterative fashion, such that elements whose densities are small enough are assumed to be voids [74]. Fig. II-14 shows a typical life cycle for a topology optimization of a brake pedal, starting from a computer-aided design produced outside Abaqus; the latter is not a necessary condition.

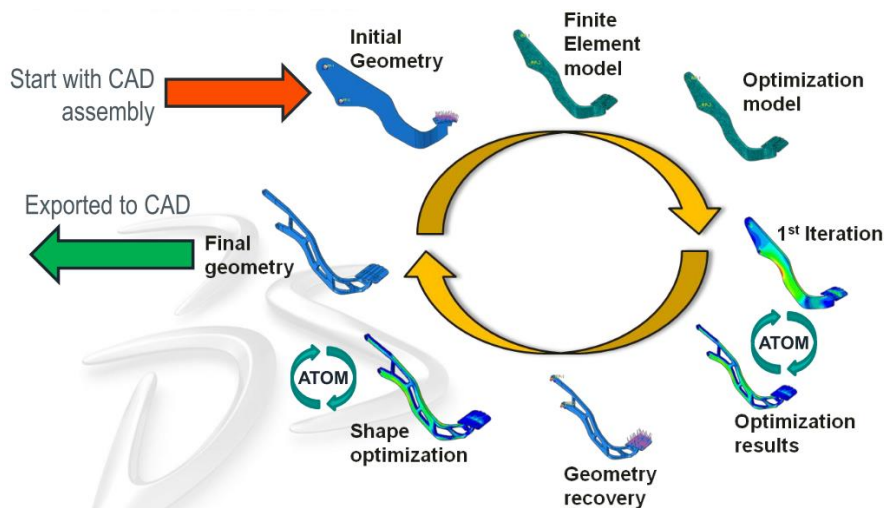


Fig. II-14. Abaqus topology optimization module life cycle, 2011 [74].

Generally, a topology optimization algorithm introduces element densities as variables, and an interpolation function to relate it to other design variables. Sensitivity analysis is then conducted using optimality criteria or gradient-based algorithms [75]. The iterative process then leads to designs that meet the objective and constraints.

Abaqus reports two algorithms for topology optimization: general and condition-based. The documentation states that the general algorithm is “partly described in Bendsoe and Sigmund” [59], which, although not explicitly stated, alludes to gradient-based algorithms, such as the Method of Moving Asymptotes (MMA), along with the Solid Isotropic Material with Penalization (SIMP) and/or Rational Approximations of Material Properties (RAMP) interpolations. MMA is a programming algorithm that use convex and separable approximating functions; the latter decoupling the design variables in the necessary optimality conditions. SIMP changes the elemental relative density continuously between zero (void element) and one (solid element), where stability issues near zero are solved by restricting the lower bound [73]. RAMP was developed to handle boundary-dependent loads (e.g., pressure) by defining a volume fraction of each void element as an incompressible fluid, allowing load transfer through them [75]. The general algorithm can use various design responses to compose the objective and constraints following Eq. 25. Table II-3 shows the design responses available on the general algorithm for objectives and constraints in Abaqus. There are other constraints, specifically geometric restrictions, that can be implemented to control the optimization region, such as ensuring planar symmetry or freezing an area such that its defined density (and stiffness) remains constant throughout the optimization. An example for VLTAS design is freezing the area interacting with the

atmosphere such that continuity is maintained, and the internal air can be eventually vacuumed.

Table II-3. Design responses supported by Abaqus general topology optimization, 2016 [59].

Design Response Available	Objective	Constraint
Center of gravity	Yes	Yes
Displacement and rotation	Yes	Yes
Eigenfrequency (from modal analysis)	Yes	Yes
Energy stiffness measure	Yes	Yes
Internal and reaction forces and moments	Yes	Yes
Moment of inertia	Yes	Yes
Scaled element-centroidal von Mises stress	Yes	Yes
Strain energy	Yes	No
Volume	No	Yes
Weight	Yes	Yes

The condition-based algorithm, described in [76], is the second and reportedly the most computationally efficient of the two topology optimization algorithms. Opposite to the general algorithm, it changes the relative density discretely (either void or solid). Nonetheless, total strain energy is the only design response used as the objective function, and volume fraction as the only constraint.

Abaqus allows topology optimization only based on static and Eigen analyses, supporting geometric nonlinearities and contact, as well as the following material definitions: linear elastic isotropic to anisotropic; elastic-plastic with Mises or Hill yield surfaces; isotropic hardening; and hyperelastic, except for the Marlow model and test-data based [59]. Temperature²⁹ and field variable dependencies can be included as well.

It is important to realize that the optimized solution is not unique and highly dependent on the discretization (i.e., mesh-dependent). Generally seen as a numerical instability, finer meshes tend to lead to scaled layouts instead of refined, qualitatively-equal

²⁹ Temperature dependence is supported by the general algorithm, only as constant temperature loading.

solutions [73]. Fig. II-15 shows the optimized solution of a beam as it is further discretized, resulting in a more detailed, or a scaled representation, of the previous.

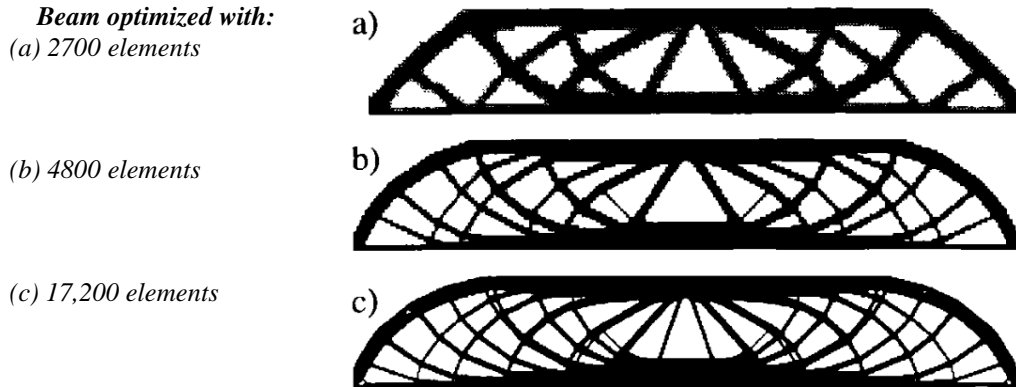


Fig. II-15. Optimal topology mesh dependence example, 2003 [73].

2.3. Structural Stability

Structural stability is a fundamental consideration in solid mechanics, particularly in thin structures subjected to large deformations. Its study dates back to Euler's elastic column solution in 1744 [77], and from there has extended tremendously into arches, frames, shells, etc., with solutions that include elastic, inelastic, viscoelastic, viscoplastic, thermal, fracture and chaotic effects³⁰. Stability can be framed as a question on whether the body in equilibrium departs from it when slightly disturbed.

The sudden departure from equilibrium is characterized nonlinearly in terms of *bifurcation*, *buckling*, *snap-through*, and/or *collapse*. Bifurcation is a mathematical term representing the point of sudden departure from equilibrium, observed by a bifurcation of the load-displacement curve; the load magnitude at which it occurs is commonly referred to as the buckling, critical, or bifurcation load. Snap-through describes a physical snap or

³⁰ Ref. [77] provides a broad overview on the developments made in this topic up to year 1999.

jump of a structure (e.g., the moment in which an arch subjected to an orthogonal load changes slope signs) represented in the load-displacement curvature. Fig. II-16 compares the load-displacement curves of these two phenomena. Lastly, collapse refers to the behavior of a structure, whether bifurcation or snap-through.

A classical case study is the circular cylindrical case under a compressive load or stress. Using the classical theory of infinitesimal deflections based on an infinitely-long shell and axisymmetric waves, a critical stress was first predicted by Lorenz, and revised by Timoshenko and Southwell between 1908-1914, given by Eq. (26) [78]. Experiments showed critical loads three to eight times smaller than what Eq. 26 predicted, later attributed to geometrical imperfections of the cylinders tested. The imperfection sensitivity was proven when experimental results of almost-perfect cylinders approached Eq. 26 [65].

$$\sigma_{cl} = \frac{E}{\sqrt{3(1-\nu^2)}} \left(\frac{t}{r}\right) \rightarrow P_{cl} = \frac{2\pi Et}{\sqrt{3(1-\nu^2)}} \quad (26)$$

where: σ_{cl} = classical critical load t = shell thickness
 E = modulus of elasticity r = shell radius
 ν = Poisson ratio

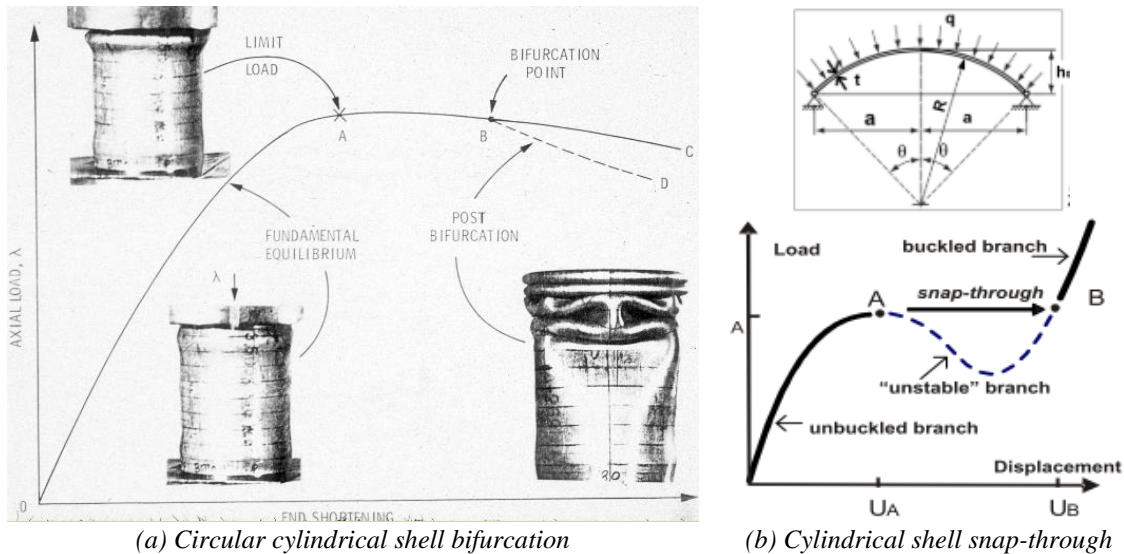


Fig. II-16. Shell bifurcation and snap-through examples [79,80].

The imperfection sensitivity of these cylindrical shells is attributed to interactions among the buckling modes, which tend to be conglomerated around the bifurcation point, leading many potential branches to follow [81]. It is important to state that a structure must buckle at its first bifurcation load [77], which then asks whether a particular theory is suitable to accurately make such prediction.

The amount of published documentation on circular cylindrical shells lends an opportunity to evaluate the application of geometric imperfections and material nonlinearity in Abaqus based on the progress made on these shells. For example, Bažant [77] argues that localized inelastic strain can play an important part in stability through strain softening. He also states that “*the greatest emphasis is currently being placed on the analysis of instabilities and bifurcations caused by propagation of softening damage or fracture in materials*”. Furthermore, Kobayashi and Mihara [82] discussed and validated the use of Abaqus’ implicit and explicit techniques on a circular cylindrical application with linear material. These arguments drove a study on the combined effects that geometric imperfections and material nonlinearity have on this application using the Johnson-Cook (JC)³¹ model in order to understand its application in Abaqus and its relation to shell thickness variations.

The Johnson-Cook model (Eq. 27) is a phenomenological viscoplastic model, developed in 1983 by Johnson and Cook [83]. Initially developed for metals in high strain rate applications, it models stress flow as a function of inelastic strain, strain rate and

³¹ The JC model was used due to the immediate data availability of materials resembling the stiffness properties required by current VLTAS designs, such as the Hexakis; see section 2.1.2.

temperature, to capture isotropic hardening, strain hardening, and thermal softening; left to right with respect to the bracketed expressions in Eq. 27.

$$\sigma_y(\varepsilon_p, \dot{\varepsilon}_p, T) = [A + B\varepsilon_p^n] \left[1 + C \ln\left(\frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_0}\right) \right] \left[1 - \left(\frac{T - T_0}{T_m - T_0}\right)^m \right] \quad (27)$$

where: T_m = melting temperature
 T_0 = reference or room temperature
 ε_p = effective (accumulated) inelastic strain
 $\dot{\varepsilon}_p$ = effective (accumulated) inelastic strain rate
 $\dot{\varepsilon}_0$ = reference strain rate (i.e., effective inelastic strain rate of the quasi-static test used to determine the yield and hardening parameters A, B and n)
A = initial yield stress
B = strain hardening constant
C = strain rate dependent coefficient
n = work-hardening exponent
m = thermal softening coefficient

2.4. Surface Stress

Classical continuum theory dismisses what is commonly referred to as ‘size-dependent effects’, such as stress moments, inertial spin and other second-order effects inherent to the micro-structure of materials [84,85]. Dismissing such effects allows continuum theory to become size-independent [86]. Nonetheless, size-dependent effects can become significant when microstructural material dimensions are near the order of magnitude of structural components, such as in the case of microscale and nanoscale applications. Several generalized theories have been developed in an effort to include these effects, categorized by Raghu et al. [85] as micro continuum, gradient continuum, and nonlocal continuum theories. Earlier works include Eringen and Suhubi’s [84,87] nonlinear theory of micro-elastic solids, where they define a micro-elastic material as a “*solid whose properties and behaviour are affected by the local deformations of the material points in any of its volume element*”. Such theory is developed to account for local effects without

using a molecular point of view, and assuming that the mass of a macroscopic element contains continuous mass distributions formed by microscopic elements, such that the macroscopic element's mass is the average of all microscopic masses. The “stress moments” effect, for example, arise from the sum of the moments resulting from the tractions acting on the surfaces of the deformed microscopic element.

Surface stress can be defined as the change in surface energy per unit of strain [88]. In other words, it can be seen as the work required to stretch a surface [89], which reflects itself as a change in surface energy density. This effect is observed in many applications, such as stress evolution during film and coating synthesis processes [90], third-order shear effects on composites [85], and wrinkling of thin films [91]. Huang et al. [91] discusses the effect that surface stress has on the critical strain needed to wrinkle a thin film bonded to a thick elastic substrate. They treat it macroscopically by assuming that the film surface is a layer with its own modulus and thickness. The surface layer thickness is said to be material-dependent, varying from one atomic spacing for crystalline materials to a few nanometers for polymers. They suggest that, in their application, surface stress effects become important for film thicknesses below 100 nm.

This topic is briefly discussed here as previous VLTAS design dimensionality suggests that surface stress effects may need to be considered. These are evidently size- and material-dependent, and are therefore not considered. Nonetheless, the potential design variability that surface stress may result is kept in mind.

2.5. Summary

Firstly, the types, current use, and development of LTAS were discussed to understand the basic principles behind LTA flight, as well as the motivation behind the regained interest. It was found that LTAS development targeted unmanned long-endurance missions, up to 100 days, for sensorial applications, such as ISR and technology testing. It was also found that operational altitudes were prescribed outside 6-18 km (20,000-60,000 ft) altitude. This last finding precipitated a review identifying the altitude-dependent design factors. From an aircraft dynamics perspective, factors include wind speed and the resulting effects on the equations of motion, to include added mass. From a quasi-static design perspective, factors include pressure, density and temperature changes. Table III-1 lists these and their ranges.

The development of NASA's ULDB followed. The discussion contrasted the benefits and challenges of ZPB and SPB, where the long duration and daily altitude stability of SPB, combined with NASA's altitude goal, drove them to the pumpkin-shaped SPB design. Said design was then described, both from stress reduction and material perspective. NASA's selection of LLDPE as the material for its ULDB program drove the need for orthotropic nonlinear thermoviscoelastic models that could represent the temperature and pressure changes experienced by the balloon as it floats to and maintains operational altitude. It also drove the number of tendons (refer to Fig. II-8) in the design, as to not allow yielding of the SF family of LLDPE films produced for it. The ULDB discussion closed with a summary of the film testing conducted by Sterling and Fairbrother [48], identifying PET, PEN, and LLDPE as potential films for LTAS design.

Recent VLTAS findings were summarized next. In recent research efforts, geodesic configurations and their extensions showed theoretical feasibility with linear elastic approximations of materials, such as CNT and graphene. Their frame/membrane configuration allowed researchers to determine that the frame stiffness and membrane's material properties are key contributors to the stability of the structure. Their efforts left two potential paths: {1} develop the manufacturing processes that would enable their designs; or {2} explore alternate designs that more heavily compensate the lack of material stiffness thru other means. The research presented in this dissertation follows the latter path via air-stiffened designs.

Structural M&S techniques and procedures were subsequently described based on the Abaqus FE package, with key processes presented in Fig. II-11. There, model definitions, such as material, mesh composition, and analysis procedures were discussed. Material definitions include the Hookean representation and creep power law, where the UMAT subroutine can be used to defined material behaviors not available in Abaqus when required. The mesh composition discussion led to the description of membrane elements, as the elements primarily used to describe the envelope films from which the air-stiffened designs are expected to be made from. The discussion on procedures covered both implicit and explicit solutions along with their methods of providing solution stability and convergence, including the inclusion of geometric nonlinearity and strain measures; the relevancy of the implicit and explicit solvers is clear in their use for Chapters V and VI, respectively. Lastly on this section, a review of the processes, algorithms, benefits, and limitations of topology optimization was presented; used to explore alternative designs

paths, as presented in Chapter IV. The chapter closed with { 1 } a discussion of several types of structural instabilities (e.g., bifurcation and snap-through) with findings on the circular cylindrical shell, and { 2 } surface stress effects. The former was brought as background information to support the study on instability and material nonlinearity presented in Chapter III. The latter was brought to raise the concern that resulting VLTAS dimensionality can make size-dependent effects significant. The notable modeling definitions used for the solutions presented in the subsequent chapters are listed in Table II-4 for reference.

Table II-4. Finite element definitions used for solutions detailed in subsequent chapters.

Category	Modeling Component or Definition	Chapter			
		III	IV	V	IV
Solver	Linear implicit static		x		
	Nonlinear implicit dynamic with Euler integration framework	x	x	x	
	Nonlinear explicit dynamic				x
Material	Linear elastic (i.e., Hookean) constitutive representation	x	x	x	x
	Linear elastic with Johnson-Cook plasticity	x			
Mesh	2D axisymmetric shell elements			x	
	2D axisymmetric membrane elements		x	x	
	2D axisymmetric solid elements		x		
	3D shell elements	x			
	3D membrane elements		x		x
	3D solid elements		x		
Interaction	Hydrostatic representation of fluid-structure interactions			x	x
	Normal and tangential contact			x	x
Optimization	Topology optimization		x		

III. STABILITY AND MATERIAL NONLINEARITY

The behavior of circular cylindrical shells subjected to static and dynamic loading has been studied, modeled and documented extensively. Their instability under axial loading is commonly associated with imperfections, boundary conditions and constant elastic moduli. This chapter rather focuses on the effects that increasingly smaller thicknesses and changes in the modulus after yield onset can have on the equilibrium path of circular cylindrical shells, as estimated by FE methods. Considering shell thicknesses from 1E-3 to 1E-6 m and using the Johnson-Cook constitutive relationship, results show that {1} the equilibrium path, as represented by the load-displacement curve, is highly dependent on shell thickness; and {2} inelastic strains drive path deviations and distinct post-collapse shapes. Furthermore, a reduction in the yield onset can result in yielding before collapse, changing the equilibrium path and post-collapse shape.

The study of the circular cylindrical shell as part of the research presented in this dissertation is primarily motivated by the following question: is the selection of the theoretical approach dependent on the thinness of the film intended to shape a VLTAS? Understanding the role that increasingly small thicknesses have on a component's ability to remain stable under compressive stresses was central in the decision-making process that led to air-stiffened designs. From a structural perspective, the idea of using air to shape and tense components with ineffectual bending stiffness for VLTAS design was driven by the limitations presented in this chapter.

3.1. Problem Description

FE solutions of circular cylindrical shell models defined with both linear elastic and JC isotropic hardening were compared for shell thicknesses 1E-3, 1E-4, 1E-5 and 1E-6 m. The intent was twofold: {1} obtain quasi-static FE estimates of the equilibrium paths past the point of collapse for models defined with and without the inclusion of hardening; and {2} contrast the solutions across the thicknesses considered to evaluate the association of increasingly small thicknesses with the elemental formulation.

The response of circular cylindrical shells to various loading configurations and time dependency, as well as boundary conditions, has been studied extensively. These shells are particularly susceptible to instability under axial loading, and their disposition to buckling before reaching their failure stress has led most investigations to focus on their stability under linear elastic material definitions. Patil et al. summarizes some of the published work regarding the instability of plates and shells [92]. Amabili dedicates a chapter on the stability of circular cylindrical shells under static and dynamic axial loading, where he mentions, among many recent developments, the classical prediction of the buckling load, P_{cr} [93]. Shown in Eq. 28, the classical P_{cr} is based on axisymmetric deformation and has been known to overestimate the critical point observed in experiments associated to effects not accounted by the underlying theory, specifically: pre-buckling deformations, boundary conditions, and geometric imperfections [81].

$$P_{cr} = \frac{2\pi E h^2}{\sqrt{3(1-\nu^2)}} \quad (28)$$

where: E = Young's modulus
 h = thickness

ν = Poisson's ratio

The finite element model (FEM) description is introduced next, to include the mesh convergence study and the development of the applied imperfection. The equilibrium solutions for models of various thicknesses under both material definitions abovementioned are presented and discussed thereafter. The effects of a reduction in the onset of yield on the equilibrium path are discussed as well.

3.2. Methodology

3.2.1. Geometry Description

The FEM is based on the geometry shown in Fig. III-1, resulting in a shell of length-to-diameter ratio of one, as to mirror the spherical volumes targeted in prior VLTAS research, with a thickness range representative of the film thicknesses expected for buoyancy potential. The rotational and displacement degrees of freedom (DOF) of the top and bottom edges were constrained, except for the vertical displacement DOF at the top edge, which allows free loading along the z axis.

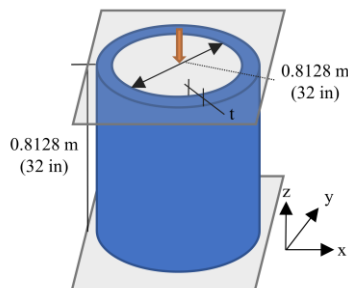


Fig. III-1. Circular cylindrical shell dimensions diagram.

In an FEM environment, a perfect circular cylindrical shell will not collapse when perfectly vertical loading is applied unless an imperfection is applied. Rather than focusing on the imperfection sensitivity of the circular cylindrical shell considered, the use of an imperfection is solely used to trigger deformation patterns that enable bifurcation.

Therefore, a geometric imperfection is selected such that its impact on the shell's collapse response is minimized, therefore intending to estimate the collapse responses of near-perfect shell. Specifically, eigenmode imperfections are chosen since it is a numerical approach easily applied in general FEM codes and useful to produce arbitrary imperfections.

The effect of the eigenmode imperfection on the collapse response is an important consideration since imperfection sensitivity is highly dependent on its shape and amplitude [94]. Teng and Rotter showed that the knock-down factor (i.e., ratio of observed to classical bifurcation or critical loads) approximately ranges from 0.2 - 0.75 for axisymmetric imperfections of amplitudes ranging from $0.1h$ to h , respectively, where h is the thickness of the shell. Furthermore, asymmetric imperfections showed a slightly higher knock-down factor, ranging from 0.25 - 0.9 within the same amplitude range. These exemplify thickness-dependent effects found on curved thin structures under compression.

Considering the former discussion and in the interest of representing an arbitrary imperfection with minimal impact on the bifurcation, the imperfection used to produce the initial state of the shell in the collapse analyses presented was based on the combination of the first 17 eigenmodes obtained from a linear buckling analysis, and an amplitude of $0.1h$. The initial state of the resulting cylinder is shown in Fig. III-2, magnified 1,000 times (top) and 10,000 times (bottom); the magnification was needed to visually discern shell imperfections. Combining eigenmodes provided an imperfection that superimposed axisymmetric and asymmetric modes, where the authors considered 17 eigenmodes enough modes to capture the axial and circumferential waves observed in the modes extracted.

Kobayashi and Mira utilized this approach with 18 eigenmodes and an amplitude of $0.01h$, obtaining a knock-down factor of approximately 0.9 in their FEM solutions and matching the equilibrium path to experimental results within most regions of pre- and post-collapse [82]. Even so, Kobayashi and Mira stated that no significant differences were found in the collapse response when changing eigenmodes assigned to the imperfection.

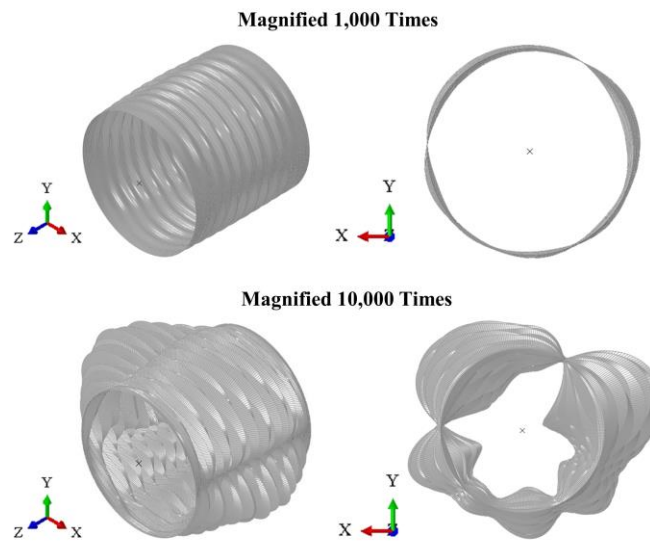


Fig. III-2. Model with imperfection from 17 buckling vectors and $0.1h$ amplitude.

3.2.2. Material Description

The shell material was defined as either linear elastic with $E = 110$ GPa and $\nu = 0.36$ only, or including the JC model with isotropic hardening only as shown in Eq. 29, where σ_f is the flow stress, A is the yield stress, B is the strain hardening constant, ϵ_p is the inelastic strain, and n is the strain hardening coefficient. Fig. III-3 exemplifies the stress and strain curves for the material models selected, and the specific values selected for A , B and n of Eq. 29, here referred to as JC plasticity. It is relevant to mention that, while such high modulus is unavailable in polymer films, such films have shown nonlinear material

behavior (refer to section 2.1.2). This material representation (Fig. III-3) is therefore intended to exemplify effects associated with changes in moduli triggered by accumulated strains.

$$\sigma_f(\epsilon_p) = A + B\epsilon_p^n \quad (29)$$

where: ϵ_p = effective (accumulated) inelastic strain
 $A = 8.62\text{E}8 \text{ Pa}$ = initial yield stress
 $B = 3.309\text{E}8 \text{ Pa}$ = strain hardening constant
 $n = 0.34$ = work-hardening exponent

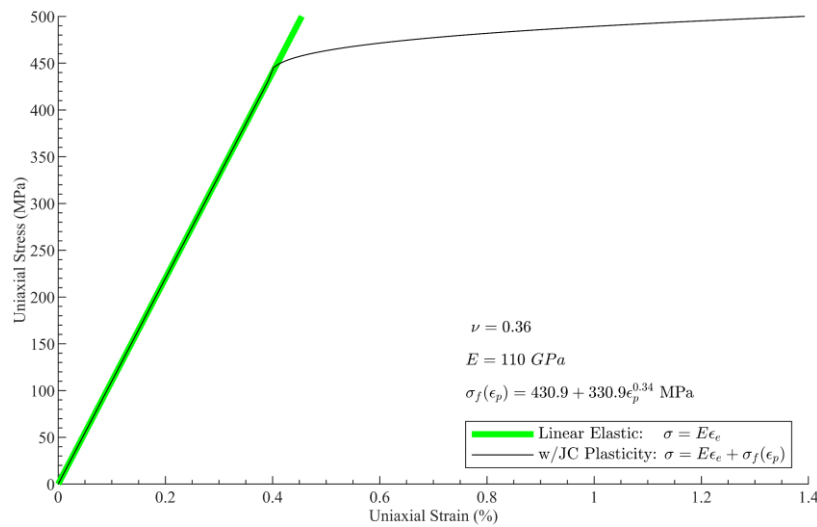


Fig. III-3. Uniaxial stress vs strain representation of material models.

The complete JC relationship is a semiempirical model initially developed to describe the inelastic behavior of metals at high strain, strain rates and temperatures [95–98]. The complete relationship includes strain hardening and thermal softening terms, shown in Eq. 30³². Nonetheless, for the purpose of static and quasi-static solutions at typical room temperatures, just as the solutions presented, the strain hardening and thermal softening terms become negligible and were therefore excluded from these solutions. In

³² Eq. 27 is repeated as Eq. 30 to include specific values used for the model constants in the variable description.

fact, quasi-static solutions that included Eq. 30 under an adiabatic presumption yielded negligible differences compared to those presented³³.

$$\sigma_f(\varepsilon_p, \dot{\varepsilon}_p, T) = [A + B\varepsilon_p^n] \left[1 + C \ln\left(\frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_0}\right) \right] \left[1 - \left(\frac{T - T_0}{T_m - T_0}\right)^m \right] \quad (30)$$

where: T_m = melting temperature

T_0 = 273.15 K = reference or room temperature

ε_p = effective (accumulated) inelastic strain

$\dot{\varepsilon}_p$ = effective (accumulated) inelastic strain rate

$\dot{\varepsilon}_0$ = 1 = reference strain rate (i.e., effective inelastic strain rate of the quasi-static test used to determine the yield and hardening parameters A, B and n)

A = 8.62E8 Pa = initial yield stress

B = 3.309E8 Pa = strain hardening constant

C = 0.012 = strain rate dependent coefficient

n = 0.34 = work-hardening exponent

m = 0.8 = thermal softening coefficient

3.2.3. Applied Loading

A displacement load was located at the top center (refer to Fig. III-1), equally distributed along the vertical displacement DOF of all the nodes located along the top edge of the shell. Said loading was applied linearly in increments over a timescale of 1 s for a maximum target displacement of 0.01 m or 1.2% of the cylinder's initial length. Displaced-controlled analyses were used to look at changes in load resultant at the reference node for an increasing displacement condition. In other words, for an ever-increasing displacement condition, there are equilibrium states that produce varying load resultants.

3.2.4. Solver Selection

The solutions presented used an implicit dynamics solver with the backward Euler operator framework to produce quasi-static solutions, as solution convergence after collapse limited the solutions produced with Abaqus static solvers. Specifically, the Newton-Raphson-based static solver with and without artificial damping for stabilization,

³³ The solutions presented in Appendix A demonstrate the similarity among solutions of models defined with various JC material components.

and the modified Riks static solver [59,61,99]. Additionally, geometric nonlinearity is enabled to allow collapse responses to develop. These solver selections enabled the evaluation of the Helical Sphere air-stiffened design presented in Chapter V, which encounters the fundamental nonlinearities of the collapse solution.

The Abaqus implicit dynamic solver uses either HHT or backward Euler operators for the numerical integration of the equilibrium equations; both provide various degree of algorithmic damping [70,100]. The backward Euler operator was chosen by the authors for the development of the solution presented as it is known to damp the solution significantly over time, providing larger algorithmic damping than the HHT [101]. Its dissipative characteristics are exemplified by the rapid loss of angular momentum of a spinning cube [102] or the damped response of an undamped mass-spring system [103]. Such significant energy dissipation (i.e., energy underprediction of a dynamic system) promotes the desired quasi-static solution, where any remaining inertial effects regularize unstable static behavior [61]. Those inertial effects are reflected as KE in the solution (refer to the energy discussion below). Contrary to the HHT operator, the amount of dissipation is not explicitly controlled, but rather becomes a function of discretization, time step, and stiffness [102]. Static solvers remove damping or use alternate sources of damping to promote convergence, but use attempts of the Abaqus static solver negated convergence early in the solution [61]. It is important to recognize, nonetheless, that the solutions presented here were similar to those produced with implicit static and Riks solvers up to the points where the static solvers stopped converging. Furthermore, Kobayashi and Mihara utilized implicit

static and explicit dynamic solvers to find the equilibrium path of their linear elastic cylinder, showing similar results [82].

In terms of accuracy, the backward Euler operator is first order, that is, the local truncation error (i.e., error associated with the truncation of the Taylor series at each time increment) is $O(\Delta t^2)$ and the global error (i.e., the total error or deviation from the exact solution) is $O(\Delta t)$, where the global error is a function of the truncation error and the time stepping scheme [100,104]. Abaqus controls the accuracy of implicit dynamic solutions with backward Euler using the same convergence criteria as (implicit) static solutions. Among other criteria, Abaqus ensures that errors in the residuals are less than 0.005, or 0.02 if convergence is not achieved within nine iterations [61]. The solution presented below used these default residual values and time steps that ranged from 2E-6 s to 2.6E-2 s to achieve convergence at each increment.

3.2.5. Mesh Selection

A mesh convergence study was conducted before the nonlinear analysis, comparing the first buckling eigenvalue between models discretized with first-order, S4R, or second-order, S8R, reduced integration shell elements. The cylinder was discretized by seeding uniformly along the top and bottom edges, and allowing Abaqus to create a structured mesh. The first eigenvalue, λ_1 , was then extracted and compared with the critical load, P_{cr} (refer to Eq. 28). The percent variation, $\epsilon = |\lambda_1 - P_{cr}|/P_{cr}$, was then calculated to produce Fig. III-4. One percent variation was reached with 8,580 first-order elements (9,000 nodes), while it took 1,540 second-order elements (5,000 nodes) to reach the same variation. This

difference in mesh size is attributed to the improved curvature approximation of the second-order element over that of the first-order element.

Nonetheless, the nonlinear solutions shown were produced with 27,200 first-order shell elements: 340 along the circumference and 80 along the length; such discretization intends to ensure that changes in shape along the circumference are captured accurately while taking the computational efficiency of linear elements. Another effect associated with the discretization, although not shown here, was that the eigenvectors associated with magnitude-ordered eigenvalues were qualitatively different, while the eigenvalues were closely spaced, a characteristic of imperfection-sensitive structures [105,106].

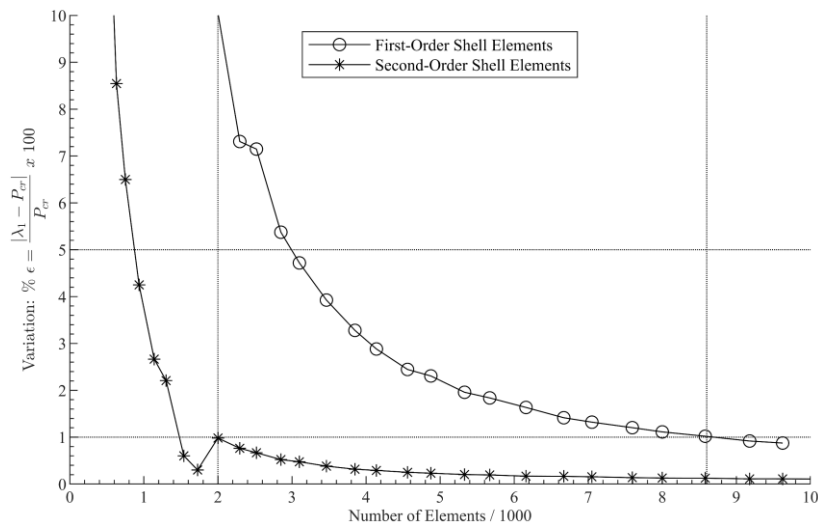


Fig. III-4. Mesh convergence study results.

The mesh convergence process is summarized in Fig. III-5. Each model iteration, defined in Python script, is edited and sent from MATLAB to Abaqus for processing and analysis. The outputs are then read and imported to MATLAB. This mesh convergence process is useful when mesh generation is based on parameter(s) variations that can be automated. In this case, the seed number that controls the number of elements along the

top and bottom edges was changed through a loop. Furthermore, similar convergence processes were followed to evaluate mesh convergence of the air-stiffened designs presented in Chapter V and VI.

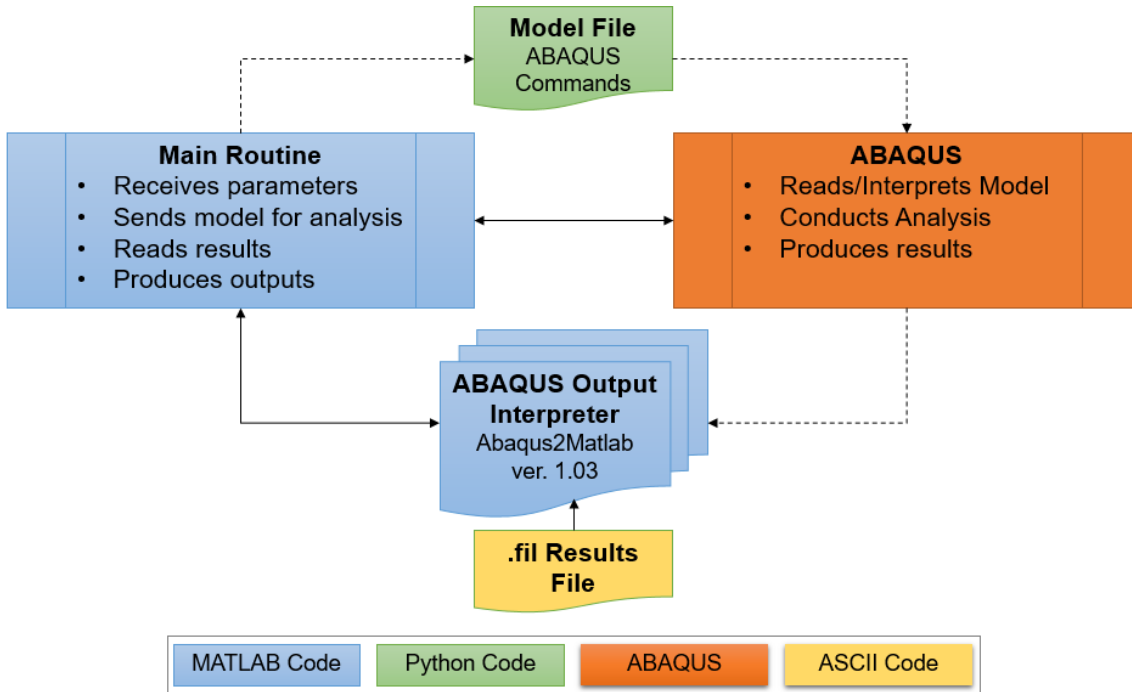


Fig. III-5. Mesh convergence study process.

3.3. Equilibrium Paths

3.3.1. Thickness Effects

Given the FEM description above, solutions were obtained for thickness from 1E-3 to 1E-6 meters, up to and past the collapse points³⁴. Table III-1 shows the four thicknesses considered in this study, along with radius-to-thickness ratios and critical loads as estimated by Eq. 28; note that the estimated critical load decreases with decreasing thickness or increasing radius-to-thickness ratio.

³⁴ The Python codes used to create the Abaqus model and export solution data are presented in sections A.2 and A.3 of Appendix A, respectively.

Table III-1. Shell thicknesses considered.

Thickness (m)	Radius/Thickness	Critical Load (N)¹
1E-3	4.064E+2	4.279E+5
1E-4	4.064E+3	4.279E+3
1E-5	4.064E+4	4.279E+1
1E-6	4.064E+5	4.279E-1
¹ Refer to Eq. 28.		

The resulting equilibrium paths for the four thicknesses in Table III-1 are included in Figs. 6 - 9, respectively, in the form of axial (load) resultant normalized by their respective P_{cr} vs. $\delta L/L$ (i.e., length reduction during loading normalized by the initial length of the cylinder). Each figure plots the equilibrium paths resulting from models defined with linear elastic material in green or JC material in black.

In the case of $h = 1E-3$ m, Fig. III-6, the path of the linear elastic material model remains linear up to collapse (i.e., axial resultant is approximately 0.82 that of the estimated P_{cr}), at which point its slope becomes highly negative until a new equilibrium state that allows incremental loading is found. Each slope change represents collapse of the current state (i.e., positive to negative slope) followed by a new state that allows incremental loading (i.e., negative to positive slope). During this process, equilibrium states often form shapes, or contours, distinct from the previous, rather than the progression of the previous. On the contrary, the shapes of the JC material model show a visual relationship even though its path does not deviate from the linear elastic model until (0.15, 0.22); this is attributed to the location where plasticity starts within the path.

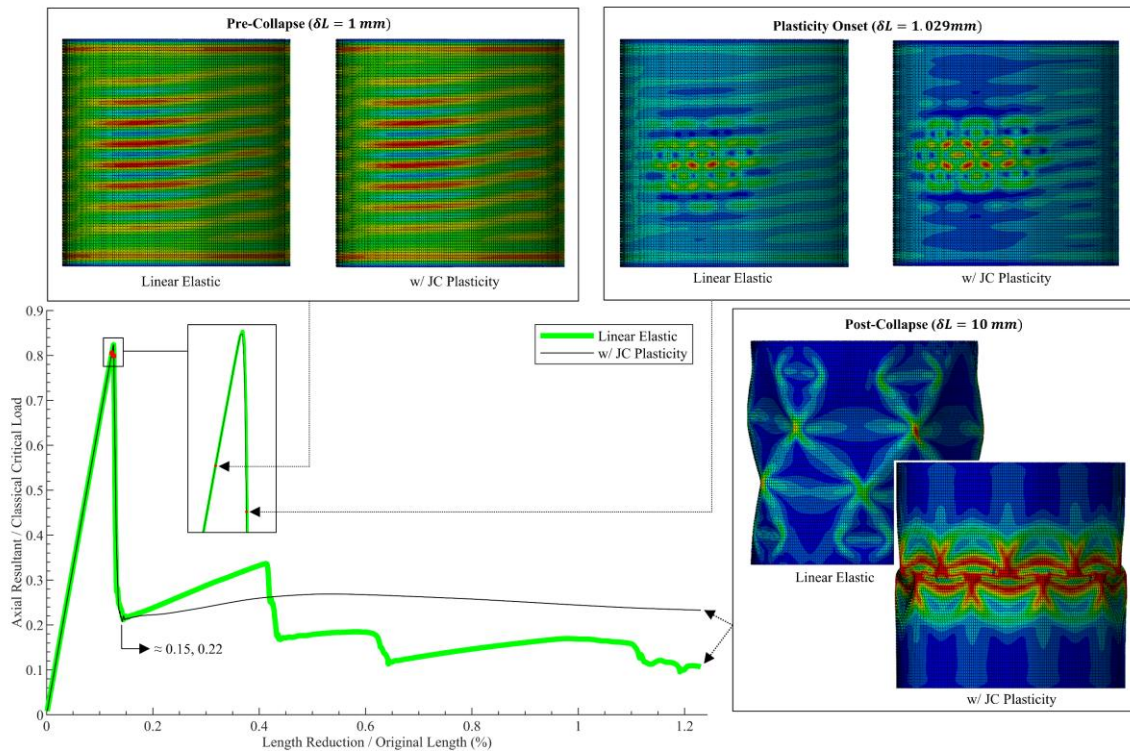


Fig. III-6. Equilibrium paths of elastic and elasto-plastic 1E-3 m thin shells.

The von Mises stress contours in Fig. III-6 show the qualitative shape pre-collapse, at the onset of plasticity, and post-collapse. Note that while both models still follow each other's path until (0.15, 0.22), and the onset of plasticity develops shape deviations that categorically change the post-collapse response. This is a direct result of high stress regions developed during the solution, which enable the JC criteria (i.e., onset of plasticity), effectively changing the stiffness and developing inelastic strains in these regions. This encourages the new shapes to be progressions of the former, and results in the crushing observed in the post-collapse shape for the model with JC plasticity.

Effects of materials nonlinearity and inelastic strains are clearly significant for the model with $h = 1E-3$ m. Nonetheless, those effects become insignificant as $h \rightarrow 1E-6$ m (refer to Figs. 7 - 9). Furthermore, inelastic strain was not present in models with

$h = 1\text{E-}5$ m and $h = 1\text{E-}6$ m, which indicates that material stiffness changes did not occur within the converged solution. These last two models stopped at 3,000 increments³⁵, before reaching $\delta L/L$ similar to those observed in Figs. 6 and 7. Slow convergence is attributed to the lack of bending stiffness, which in turn required significantly smaller time steps.

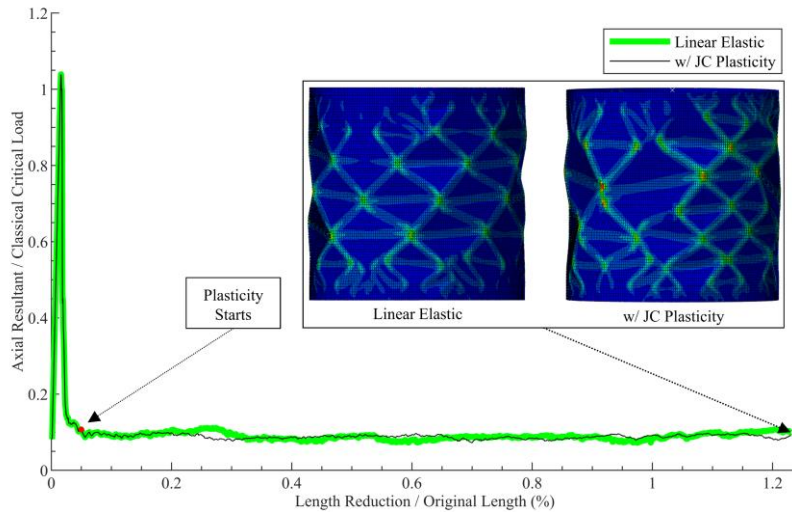


Fig. III-7. Equilibrium paths of elastic and elasto-plastic 1E-4 m thin shells.

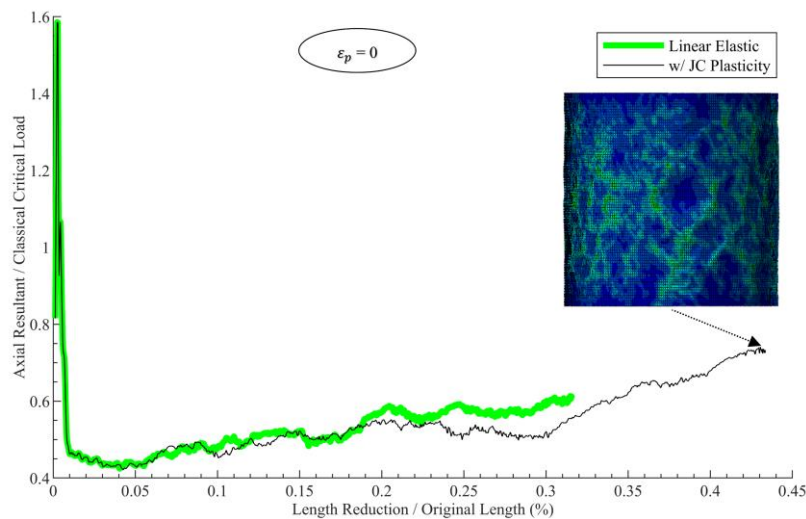


Fig. III-8. Equilibrium paths of elastic and elasto-plastic 1E-5 m thin shells.

³⁵ The maximum of increments was set to 3,000 to constrain computing time. This number of increments allowed for post-collapse behavior, while limiting output storage requirements and post-processing time. This was avoided in the Chapters V and IV solutions by limiting the number of points at which outputs are saved (e.g., solutions discussed in Chapter VI are based on a 1000 equally-distributed time points).

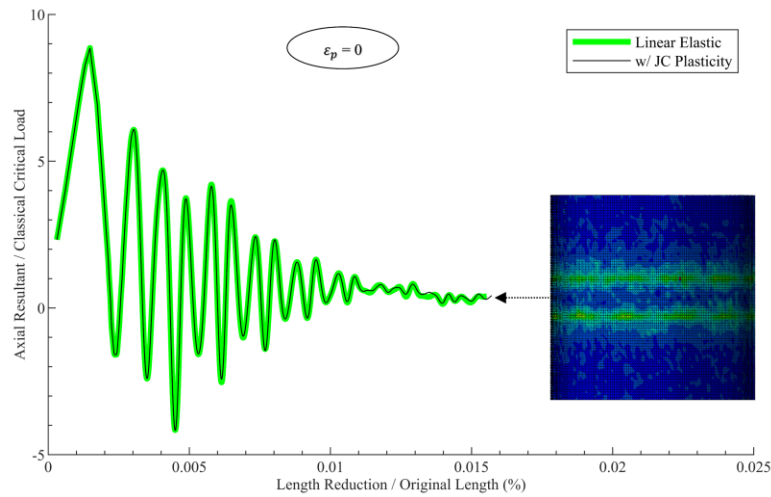


Fig. III-9. Equilibrium paths of elastic and elasto-plastic 1E-6 m thin shells.

Overall, the collapse behavior is shown to be highly dependent on shell thickness. As thickness tends to zero, the collapse load is increasingly greater than P_{cr} (refer to Figs. 7 - 9). It is known that bending generally drives collapse, as with the case of $h = 1E-3$ m, but bending effects reduce as thicknesses tend to zero, which makes the stiffness among elements and membrane stresses drive the collapse behavior. In other words, the shell acts like a membrane while still satisfying continuity along the boundary, which creates folds. These folds are local instabilities characterized as material crushing or wrinkling, generating the global (unstable) response. The theory does not consider these effects, which can justify the collapse load being greater than P_{cr} for thicknesses below 1E-3 m. One course of further study is to ask whether a portion or all of the stiffness as a result of the folds is numerical in nature and associated with the discretization. Another course is to determine whether an alternate collapse path can be found when the imperfection follows a diamond-like *Yoshimura pattern*; Yoshimura proved that the minimum energy solution of a buckled circular cylindrical shell due to a centered compressive load produced a polyhedral surface pattern consisting of identical plane triangles [78,107].

3.3.2. Solution Mesh Dependence

To the first course, the mesh seeding was increased from 340 to 400 along the circumference for the stable case of $h = 1\text{E-}3$ m to evaluate if the mesh size has any effect on collapse; Fig. III-10 compares the resulting paths. The onset of collapse increased from $0.82P_{cr}$ to $0.85P_{cr}$; the difference perhaps attributed to the states of equilibrium now available because of the added DOF. In either case, inelasticity started after the onset of collapse for both cases, and paths merged after $\delta L/L = 0.65\%$ up to the end of the analysis. This result is not sufficient to dismiss the mesh dependence entirely, but rather provides a sample of its level of impact on the solution.

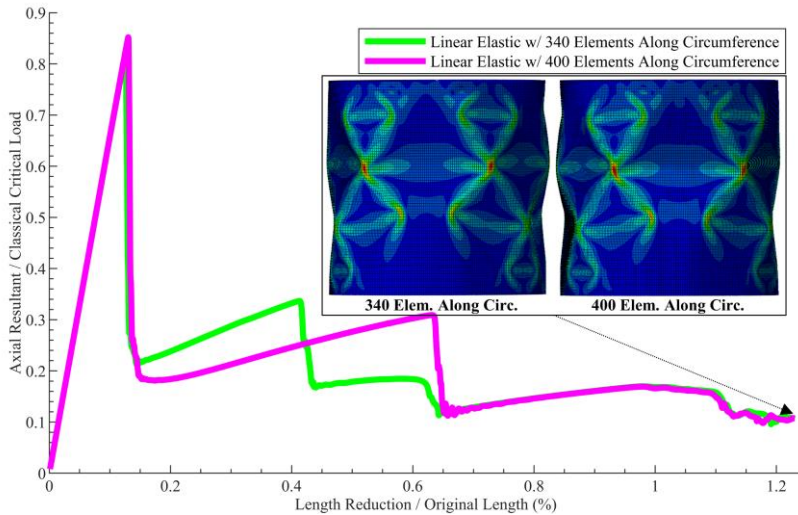


Fig. III-10. Equilibrium paths of an elastic $1\text{E-}3$ m thin shell with two mesh sizes.

3.3.3. Yield Limit Reduction Effects

Another concern was whether an earlier change in material stiffness could have an effect on the collapse behavior. This was evaluated by reducing the yield, A in Eq. 29, by an order of magnitude, and running the collapse analysis for $h = 1\text{E-}3$ m. Fig. III-11 compares the uniaxial stress vs strain curves for the JC model with isotropic hardening at

the original and lowered yield limits. The paths for both cases are plotted in Fig. III-12. The black and blue curves in Fig. III-12 represent the models containing the original and reduced yield limits, respectively. Note that, contrary to the original yield limit, the reduced yield limit results in plasticity before collapse. Such change drives collapse at $0.37P_{cr}$, compared to $0.82P_{cr}$ for the original. Fig. III-12 also shows its effects on the von Mises contour, a collapsed shape significantly different than the original (refer to Fig. III-6).

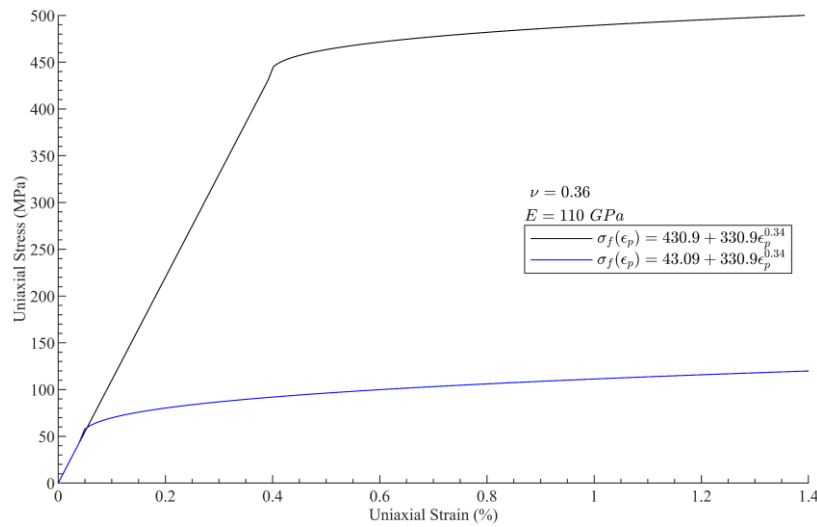


Fig. III-11. Uniaxial stress vs strain representation for two yield limits.

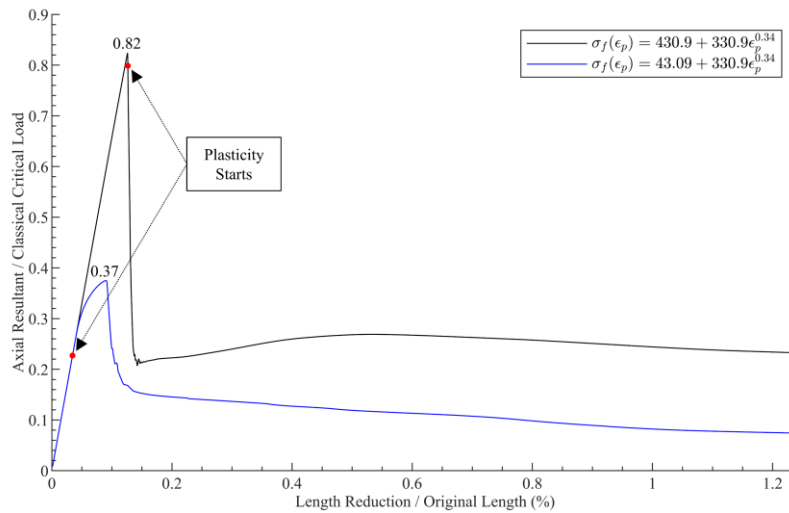


Fig. III-12. Equilibrium paths of an elastic 1E-3 m thin shell with two yield limits.

3.3.1. Effects Associated with Potential Manufacturing Faults

Manufacturing faults is another topic that would need to be considered once an air-stiffened design with buoyancy potential has been identified. As such, it is considered here to illustrate effects

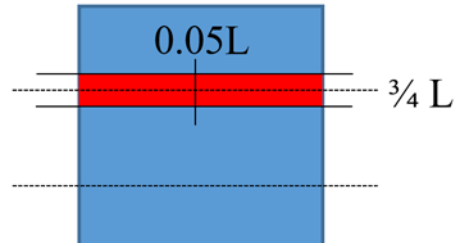


Fig. III-13. Fault location.

associated with hypothetical modulus and thickness changes that manufacturing processes may yield. These were hypothesized by reducing the thickness or modulus to half of the original value around the circumference at three quarters the length ($L \pm 0.025L$), as shown in Fig. III-13. Collapse analyses were then conducted using JC with isotropic hardening and thermal softening material definition. Reducing the thickness within this region resulted in a collapse load that was about 1% lower than that of the model without faults. Interestingly, decreasing the stiffness within the same region resulted in a 10% increase in collapse load. Both shapes are compared in Fig. III-14. The collapse concentrated around the defect region for both cases. Nonetheless, the lower stiffness at the defect in the $1/2 E$ case resulted in further strain energy absorption around the region, delaying the onset of global collapse.

3.3.1. Surface Effects

Lastly, surface effects are a relevant consideration for thicknesses smaller than those considered here. The Gurtin–Murdoch model is one of the first to present a framework for surface stresses [108,109], showing significant surface stress effects at nanoscale thicknesses [110,111]. Surface energy approaches have similarly predicted significant surface effects at the nanoscale [112]. And while there is a fundamental

difference between surface stress and surface energy [113], these theories validate surface effects as additions to the ‘bulk’ response. The thicknesses considered herein, however, are large enough to preclude their significance since they are important at size levels in the angstrom dimension.

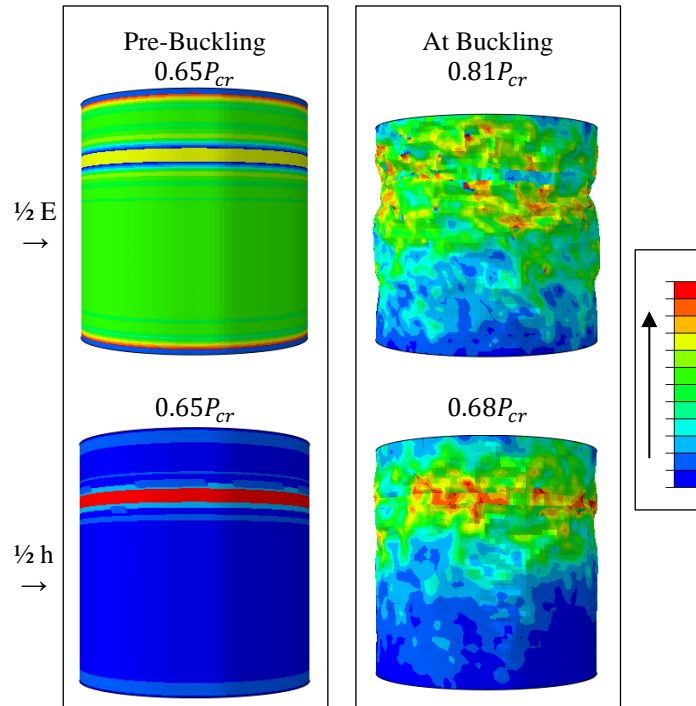


Fig. III-14. Collapsed shapes comparison for faulted models.

3.4. Summary and Conclusions

This chapter highlights the transition of shell to membrane, or loss of significant bending stiffness, through the representation of the equilibrium path of a circular cylindrical shell. The collapse solution of the linear elastic circular cylindrical shell with $h = 1E-3$ m served as a comparative base for its nonlinear material counterpart modeled with a JC material. For such case, the yield onset after collapse drove the further development of the collapsed shape, a result attributed to the change of stiffness and

inelastic strain accumulation. Furthermore, the yield limit proved to be an integral part in the collapse solution, where a yield limit 1/10 of the original changed the point of collapse and the equilibrium path post-collapse. Additionally, finer mesh discretization showed insignificant effects on the collapse point prediction, with notable differences in portions of the equilibrium path post-collapse.

In terms of effects associated with thickness, the material nonlinearity effects observed in the circular cylindrical shell with $h = 1\text{E-}3$ m dissipated as $h \rightarrow 1\text{E-}6$ m. The shell collapse solution showed a strong dependence on thickness, attributed to the loss of bending stiffness as thickness becomes very small, leaving membrane forces as the solution drivers. For example, the classical critical load underpredicted the collapse load observed in the solutions of shells with thicknesses $1\text{E-}4$, $1\text{E-}5$ and $1\text{E-}6$. This clearly opposes what it is typically observed in experiments, where the critical load overpredicts the collapse load and is attributed to effects not accounted in the theory, such as the geometric imperfection applied [81]. Furthermore, the implicit dynamics solver regularized solution nonlinearities (e.g., geometric, contact and material) via the algorithmic damping inherent to the backward Euler operator, allowing convergence well into the post-collapse region, a region not accessible with static solvers.

The loss of bending stiffness discussed here prompted the use of membrane elements to characterize the Helical Sphere and Icoron air-stiffened designs documented in Chapters V and VI, respectively, where thicknesses in the $1\text{E-}6$ order are used. The validity of membrane formulation (as opposed to shell) was confirmed during the mesh convergence study of the Helical Sphere air-stiffened design characterization by noting

negligible differences compared to shell element representations. Also, the implementation of the implicit dynamic solver was carried over to the solutions discussed in Chapter V, where quasi-static behavior was validated.

IV. DESIGN PATHS INVESTIGATION

Prior research of LTA structures withstanding, the study of air-stiffen designs as potential VLTAS is preceded by the consideration of several design paths. The first path was the consideration of a spherical shell. Specifically, the use of a low-density core material exhibiting high specific stiffness, such as Balsa wood. Such path was deemed unfeasible because the thicknesses required to prevent buckling did not allow for positive buoyancy. The second path was the use of topology optimization to challenge prior design assumptions by letting strain-based optimization objectives drive the internal composition of a spherical structure given weight constraints. Such path led to arguably novel compositions that rely on components whose constrain-driven thicknesses have shown insufficient bending stiffness to sustain the resulting loading. The issues of minimum manufacturable dimensions and the loss of bending stiffness are, once again, shown to be central limitations. Lastly, the third path validates the use of the classical hoop stress to estimate membrane radii required to meet stress targets by evaluating a circular geometry resembling a long tube. Such path poises the classical hoop stress as a conservative approximation and exposes load rate effects on the dynamic solution. These three paths shaped the idea of air-stiffened designs and supported the development and characterization of the Helical Sphere and Icoron presented in Chapters V and VI, respectively.

4.1. Coated Spherical Shell

The coated spherical shell is an appealing design due to its natural symmetry and simplicity. Nonetheless, prior VLTAS efforts deviated from such simplicity due to its

apparent infeasibility at the time at which they were considered [52]. Nonetheless, the review of the spherical shell provides background as to why such design is unfeasible, and supports an answer to the following question: can a low-density core, such as foam or light wood, make the spherical shell feasible? The short answer is that feasibility could not be proven given the low-density, commercially available, materials considered.

The maximum material density of which a rigid spherical shell can be manufactured to allow for buoyancy given an internal vacuum is estimated in Fig. IV-1 for a range of radius over thickness ratios. The need for high radius-to-thickness ratios in order to meet W/B ratio requirements is then exemplified by comparing Fig. IV-1 with the densities of the materials in Table IV-1 (e.g., many polymers and most metals have densities over 1000 kg/m³, requiring radius-to-thickness ratios over 2400). The loss of bending stiffness becomes an important issue as radius-to-thickness ratios increase, which in turn challenges the validity of the shell. Therefore, low-density materials were targeted to preserve the bending capabilities characteristic of the shell.

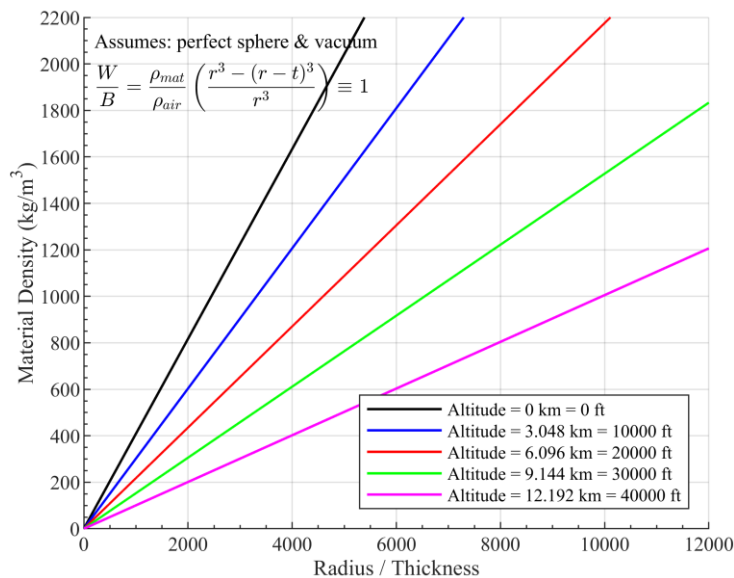


Fig. IV-1. Maximum material density required for buoyancy of spherical shell.

Table IV-1. Specific stiffness and strength of materials considered [50,57,114].

Material	Density (kg/m³)	Specific Stiffness [GPa/(kg/m³)]	Specific Strength [MPa/(kg/m³)]
DIAB Divinycell H45 Foam	48	0.001	0.013
3A BALTEK® SB.50 Structural End-Grain Balsa	94	0.021	0.067
3A BALTEK® SB.100 Structural End-Grain Balsa	153	0.026	0.084
Carbon-Core Carbon Balsa Wood	160	0.025	0.082
ERG Aerospace Duocel® Silicon Carbide Foam	161	0.017	0.009
3A BALTEK® SB.150 Structural End-Grain Balsa	247	0.032	0.106
ERG Aerospace Duocel® Silicon Carbide Foam	385	0.018	0.014
3D Systems DuraForm® AF Nylon	690	0.005	0.064
Carbon Nanotube/Bismaleimide Composite	1250	0.234	3.040
DuPont Teijin Mylar® HP PET Film, 12.2 μm	1390	0.003	0.144
DuPont Kapton 50KN Polyimide Film, 13 μm	1450	0.003	0.143
Beryllium	1844	0.164	0.201
Graphene	2000	0.250	25.000
Titanium Ti-6Al-4V (Grade 5), Annealed	4430	0.026	0.214

Low density materials then led to reduced radius-to-thickness ratios, which reduces the required material stiffness. Given a $W/B = 1$ and a buckling pressure [115] equal to the atmospheric air pressure, Fig. IV-2 estimates the minimum required specific stiffness for buoyancy at several atmospheric altitudes given any radius-to-thickness ratio. Tracing the minimum radius-to-thickness required for buoyancy in Fig. IV-1 given any material with density below 1000 kg/m^3 in Table IV-1, is notable that none meet the minimum specific stiffness required to prevent buckling. For example, Fig. IV-3 shows the W/B ratio across thickness of 0.6096 m in diameter shells made with 3A BALTEK® SB.50 Structural End-Grain Balsa with a 1 μm coating (for permeability-reduction purposes). Note that the minimum thickness required to prevent buckling is above the maximum required for a $W/B < 1$. Moreover, manufacturability and the design changes required to overcome the assumptions made (regarding vacuum levels and material uniformity) further discourages the feasibility of the shell as a VLTAS.

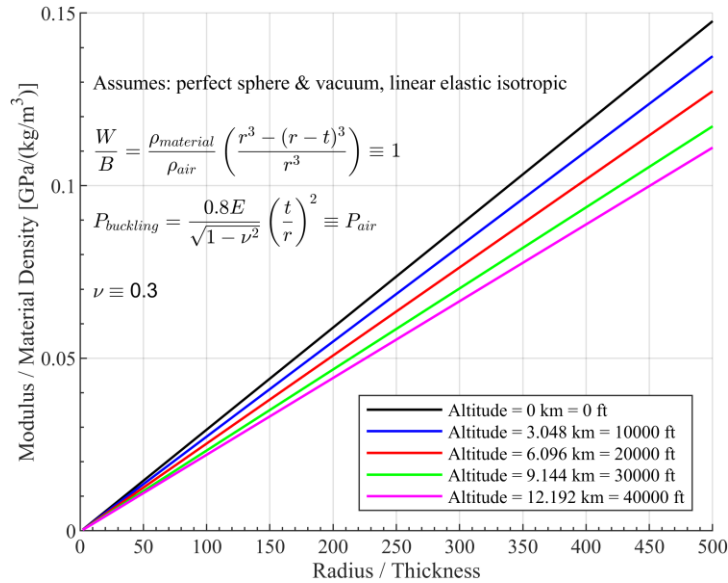


Fig. IV-2. Minimum material specific stiffness required for buoyancy of spherical shell.

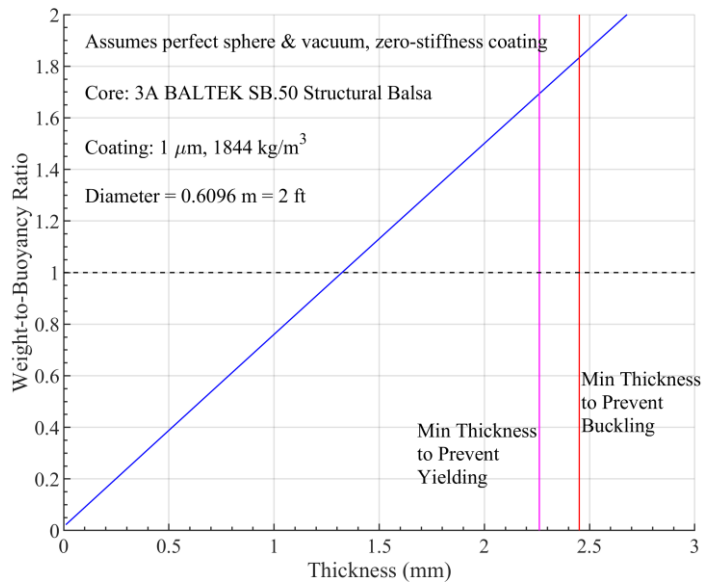


Fig. IV-3. Example of minimum thickness required for buoyancy of coated spherical shell.

Nonetheless, the study of the spherical shell was key to the transition to air-stiffened designs, as it forced the following question: given that a core material stiff enough to enable buoyancy was not found, and that the low-density cores considered need an external barrier that limits air permeability, can such thin barrier(s) drive the geometry while the very lightweight material being removed from its internal, air, shapes or stiffens said barrier(s)?

Such question, informed by the literature on NASA’s SPB research and development, drove the development of the Helical Sphere as an air-stiffened design.

4.2. Topology Optimization

The second path investigated was the use of strain-driven topology optimization based on the following three initial representations (i.e., initial geometric state) of a spherical volume: sandwich shell, solid, and as semicircular sections, as shown in Fig. IV-4. The intent was to find other design paths not envisioned that could either produce a design alternative to the Helical Sphere, which reduced material stiffness and strength requirements, or inform the Helical Sphere’s development.

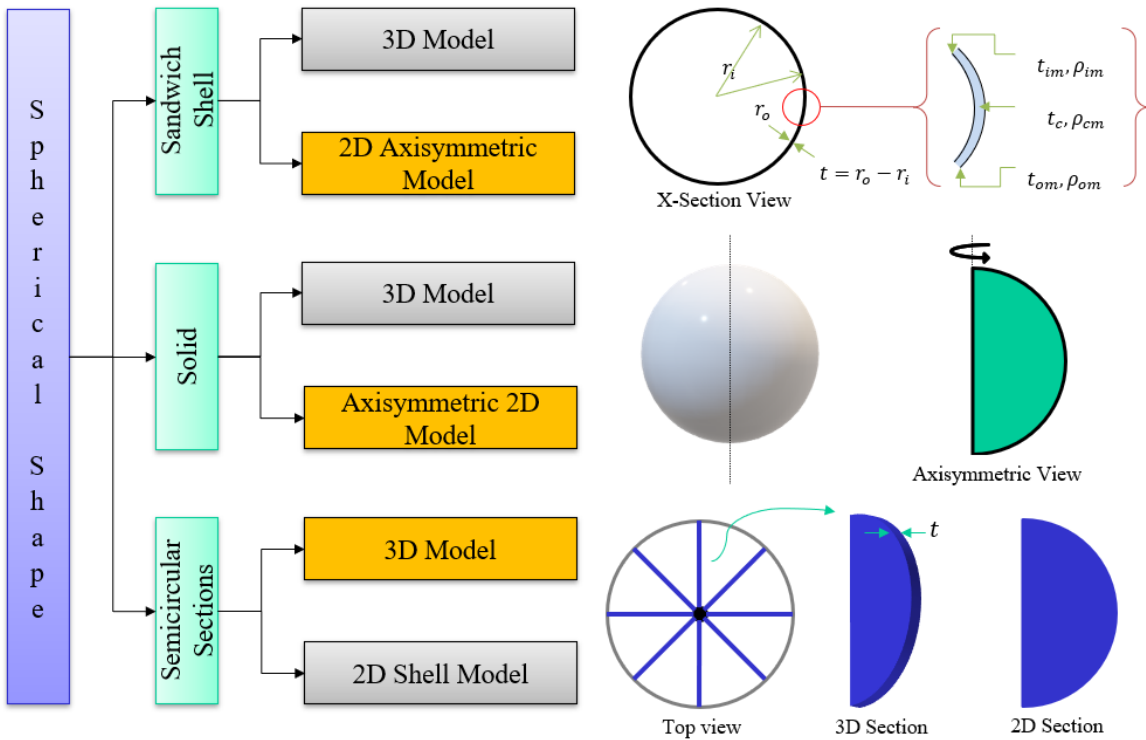


Fig. IV-4. Modeling representations for topology optimization.

The topology optimization of the three representations was conducted using Abaqus/Tosca’s sensitivity-based (i.e., general) algorithm combined with SIMP (i.e., for

each element, its modulus equals the relative density cubed times the initial material density defined), in order to observe continuous relative density changes throughout the design spaces as opposed to discrete changes characteristic of the condition-based algorithm. The objective was to minimize the strain energy, or maximize the stiffness, of the core material given a volume fraction constraint. The membranes covering the cores were not included in the design space as to allow continuity of load transfer, representing the low-permeability barrier separating the core and the atmosphere. Furthermore, all of the model parameters discussed below (and presented in Table IV-2, Table IV-3 and Table IV-4) were selected with manufacturability potential in mind as membrane properties resemble commercially-available PEN film at room temperature, the relatively-small diameter facilitates prototyping, and the volume fraction constraints are estimated to drive their weight below that of the air they displace (when small displacement is assumed).

4.2.1. Two-Dimensional Axisymmetric Representation of the Spherical Sandwich Shell

The first representation considered was a 2D axisymmetric representation of the spherical sandwich shell. Albeit the sandwich shell was deemed unfeasible as presented in section 4.1, its inclusion for topology optimization was intended to explore whether the topology of the core, or connectivity between internal and external membrane, could be optimized. In other words, topology optimization attempted to find topologies that could provide this connectivity by, effectively, removing material from an initially solid core. Starting with the design shown at the top of Fig. IV-4, the membranes and core were made into one homogeneous solid (i.e., a spherical shell) to produce the W/B ratio shown in Eq.

31. Knowing that the core thickness $h = r_o - r_i$, making the $W/B = 1$ and solving for h results in Eq. 32 (i.e., an initial shell thickness that is neutrally buoyant at a chosen altitude and vacuum pressure).

$$\frac{W}{B} = \frac{\frac{4}{3}\pi r_i^3 \rho_{ia} + \frac{4}{3}\pi(r_o^3 - r_i^3)\rho_m}{\frac{4}{3}\pi r_o^3 \rho_{oa}} \equiv 1 \quad (31)$$

$$h = r_o - r_o \left[\frac{\rho_{oa}\{h\} - \rho_m}{\rho_{ia}\{h\} - \rho_m} \right]^{\frac{1}{3}} \quad (32)$$

where: h = thickness	() _{i/o} = internal/external quantity
ρ = density	() _{()a} = air quantity
r = radius	() _{()m} = membrane/material quantity
	{ } = function of

Table IV-2 summarizes the model definitions representing the sandwich shell, including geometry, material, loading and constraint. The thickness was calculated based on 0 km sea level altitude to simulate as the structure was built at SL, PEN material density, 0.6096 m diameter, and a vacuum pressure of 3. kPa³⁶. Lastly, the constraint was set to a volume fraction of 0.52, based on a W/B ratio equals to one at 6 km altitude. Starting from Eq. 31, the volume fraction in Eq. 33 was developed by considering the fraction of original core volume needed for the structure to weight the same or less than the air it displaces at 6 km.

$$v_f \leq \frac{r_o^3 [\rho_{oa}]_{20 \text{ kft}} - r_i^3 [\rho_{ia}]_{0 \text{ kft}}}{(r_o^3 - r_i^3)\rho_m} \quad (33)$$

where: v_f = thickness	() _{i/o} = internal/external quantity
ρ = density	() _{()a} = air quantity
r = radius	() _{()m} = membrane/material quantity

³⁶ A vacuum pressure of 25 torr was chosen since a perfect vacuum is unattainable. The American Vacuum Society considers 25 torr the point between low and medium vacuum [155], which is achievable by some single pumps currently on the market [136,156].

Table IV-2. Axisymmetric model parameters for sandwich shell as initial representation.

Definition	Value(s)	Comments
<i>Initial Geometry</i>	<ul style="list-style-type: none"> - Spherical sandwich shell - External diameter = 0.6096 m - Int./Ext. membranes thickness = 12.60 μm - Core thickness = 61.42 μm 	Total thickness from Eq. 32, material properties, vacuum pressure, and 0 km altitude.
<i>Material</i>	<ul style="list-style-type: none"> - Linear isotropic - Density = 1390 kg/m^3 - Modulus = 4.36E9 Pa - Poisson's ratio = 0.38 	Both the core and the membrane are based on DuPont Mylar® HP 12.6 μm PET film [116] properties.
<i>Analysis</i>	<ul style="list-style-type: none"> - Abaqus static general - Linear geometric 	Abaqus topology optimization disallows dynamic analyses.
<i>Mesh</i>	<ul style="list-style-type: none"> - Core: axisymmetric second-order elements, five across the thickness - Membrane: axisymmetric first-order elements, one across the thickness - Average aspect ratio of one 	Core element code is CAX4R, and membrane element code is MAX1.
<i>Loading</i>	<ul style="list-style-type: none"> - $\Delta P = 101.325(0.967) = 97.981 \text{ kPa}$ 	
<i>Boundary Conditions</i>	<ul style="list-style-type: none"> - Symmetry conditions along horizontal axis 	
<i>Objective</i>	<ul style="list-style-type: none"> - Minimize strain energy 	(i.e., maximize stiffness)
<i>Constraint</i>	<ul style="list-style-type: none"> - Volume fraction ≤ 0.52 	Based on Eq. 33 at 6 km.

Nevertheless, the sandwich shell and the solid core covered by a membrane resulted in inadequate solutions associated with the initial material stiffness and the stringent volume fraction constraints. As Fig. IV-5 shows, the sandwich shell linear static analysis resulted in overly large displacements. Furthermore, nonlinear geometric analysis experienced convergence limitations that prevented optimization iterations to occur. Such issues suggest that the defined material stiffness is insufficient for stable designs to develop, given the initial geometry. The solid core was then pursued after given findings.

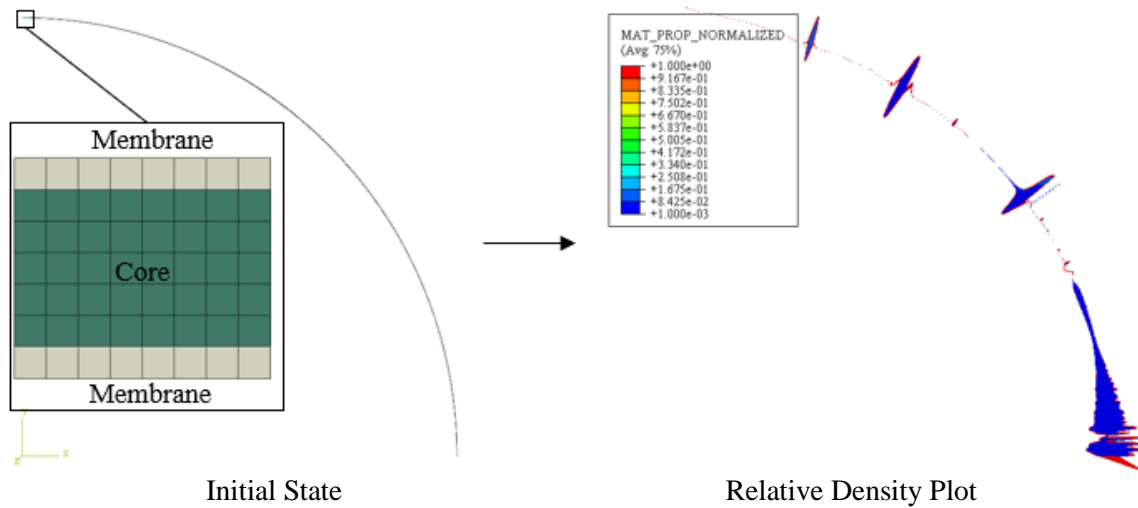


Fig. IV-5. Sandwich shell topology optimization initial state and relative density plot.

4.2.2. Two-Dimensional Representation of a Solid Spherical Core Covered by an External Membrane

The second representation considered was a 2D representation of a solid spherical core covered by an external membrane, as shown at the middle of Fig. IV-4. The core was made of isotropic properties resembling Balsa wood, while the membrane resembles Mylar film. Table IV-3 summarizes the model parameters. However, the use of a solid core as a starting point constrained the optimization to a stringent relative volume of 0.0008 to allow for buoyancy. Such stringent relative volume requirement resulted in a relative density plot with most of the core volume essentially removed. This is viewed in Fig. IV-6, where the relative density plot shows high relative densities concentrated along the axis of symmetry, and relative densities below 0.4 across the majority of the volume. Such design is expected to have the same instability limitations of the spherical shell discussed in section 4.1, and therefore this modeling representation was not studied further. Instead, the semicircular core sections were envisioned as a third representation to be optimized in order to ease the stringent relative volume constraint.

Table IV-3. Axisymmetric model parameters for solid sphere as initial representation.

Definition	Value(s)	Comments
<i>Initial Geometry</i>	- Axisymmetric representation of solid sphere - External diameter = 0.6096 m - Ext. membrane thickness = 12.60 μm	
<i>Membrane Material</i>	- Linear isotropic - Density = 1390 kg/m^3 - Modulus = 4.36E9 Pa - Poisson's ratio = 0.38	Based on DuPont Mylar® HP 12.6 μm PET film [116] properties.
<i>Core Material</i>	- Linear isotropic - Density = 155 kg/m^3 - Modulus = 4.1E9 Pa - Poisson's ratio = 0.48	Based on DIAB ProBalsa PB [117] properties.
<i>Analysis</i>	- Abaqus static general - Linear geometric	Abaqus topology optimization disallows dynamic analyses.
<i>Mesh</i>	- Core: axisymmetric second-order elements - Membrane: axisymmetric first-order elements, one across the thickness - Average aspect ratio of one	Core element code is CAX4R, and membrane element code is MAX1.
<i>Loading</i>	- $\Delta P = 101.325(0.967) = 97.981 \text{ kPa}$	
<i>Boundary Conditions</i>	- Symmetry conditions along horizontal axis	
<i>Objective</i>	- Minimize strain energy	(i.e., maximize stiffness)
<i>Constraint</i>	- Volume fraction ≤ 0.0008	

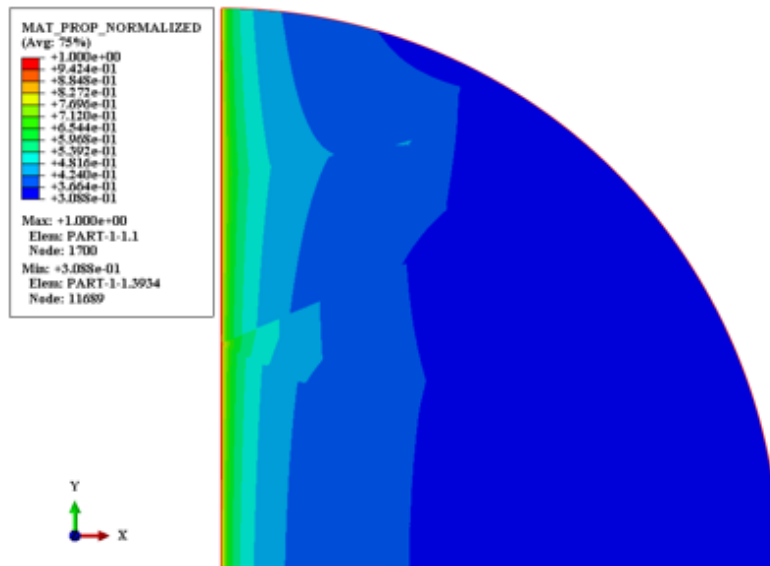


Fig. IV-6. Solid sphere topology optimization relative density plot.

4.2.3. Three-Dimensional Representation of a Semicircular Section

The third and last representation considered was a 3D representation of one semicircular section, of which several form the internal core of the structure. The geometrical development is diagramed in Fig. IV-7. Starting with a spherical membrane, its internal structure is composed of eight semicircular sections joined along an axis. Each section is then initially composed of a solid core covered by membrane material. The intent was to optimize the topology of the core to maximize stiffness given a volume constraint 0.5, considerably relaxed compare to the second representation discussed above. The core was made of isotropic properties resembling Balsa wood, while the membrane resembled Mylar film properties. Table IV-4 summarizes the model parameters.

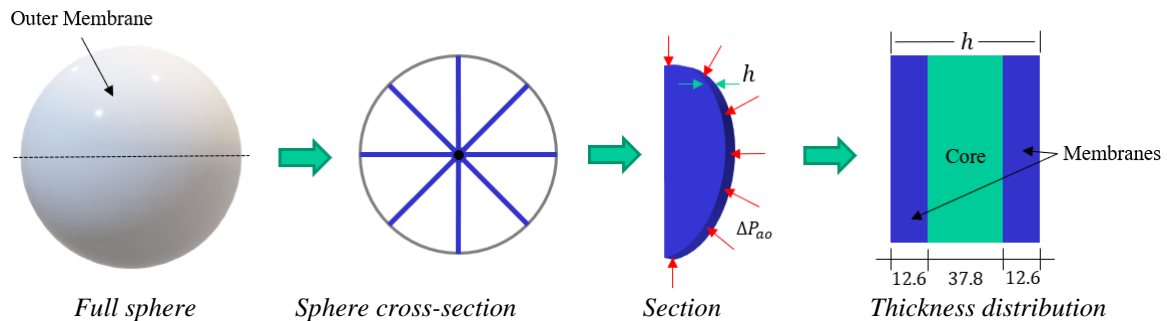


Fig. IV-7. Geometrical development of semicircular section for topology optimization.

Several topology optimizations of the semicircular core section were run to evaluate how the results change as design restrictions were added, specifically a member size greater than 10 mm (i.e., regions size with equal relative densities need to be ≥ 10 mm in length), a 30° rotational symmetry, and both combined. Fig. IV-8 shows the resulting relative density plots when no restrictions are applied and with each of the restrictions; only showing those regions of the core with resulting relative densities ≥ 0.6 . All four optimization results showed high relative densities along the border, with sparse material

spread out across and along the axes of symmetry for those with rotations restrictions, as if borders and restrictions drive high relative densities while the volume constraint forces largely empty regions in between.

Table IV-4. Three-dimensional model parameters for semicircular section as initial representation.

Definition	Value(s)	Comments
<i>Initial Geometry</i>	- Semicircular section - External diameter = 0.6096 m - Ext. membranes thicknesses = 12.60 μm	
<i>Membrane Material</i>	- Linear isotropic - Density = 1390 kg/m^3 - Modulus = 4.36E9 Pa - Poisson's ratio = 0.38	Based on DuPont Mylar® HP 12.6 μm PET film [116] properties.
<i>Core Material</i>	- Linear isotropic - Density = 155 kg/m^3 - Modulus = 4.1E9 Pa - Poisson's ratio = 0.48	Based on DIAB ProBalsa PB [117] properties.
<i>Analysis</i>	- Abaqus static general - Linear geometric	Abaqus topology optimization disallows dynamic analyses.
<i>Mesh</i>	- Core: 141,492 quadratic wedge elements, three across the thickness - Membrane: linear membrane elements, one across the thickness - Average aspect ratio of one	Core element code is C3D15, and membrane element code is M3D4.
<i>Loading</i>	- $\Delta P = 101.325(0.967) = 97.981 \text{ kPa}$	
<i>Boundary Conditions</i>	- Symmetry conditions along horizontal axis	
<i>Objective</i>	- Minimize strain energy	(i.e., maximize stiffness)
<i>Constraint</i>	- Volume fraction ≤ 0.5	

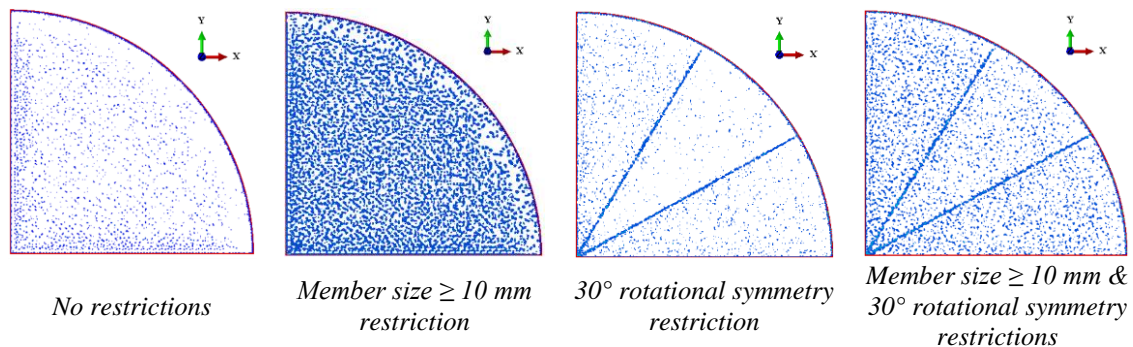


Fig. IV-8. Semicircular core section topology optimization relative density plots (showing elements with resulting relative densities at or above 0.6).

Common factors across all the ‘optimized’ solutions found, considering all three representations in Fig. IV-4, is the concentration of material along borders, and more importantly, the lack of characterization for the instability that has been previously observed when the complete strain tensor (i.e., geometrically nonlinear) is used to evaluate the response of such thin sections subjected to external pressure. While designs, such as the one presented in Fig. IV-9, can be inspired from the solutions found with the geometrically linear static analysis used here with the optimizer, the potential of these solutions should be evaluated when nonlinearity and loss of bending stiffness considerations are included. Nonetheless, these solutions demonstrated the challenges associated with searching for stable representations that enable VLTAS and cemented the idea of air-stiffened designs as the path showing the most potential.

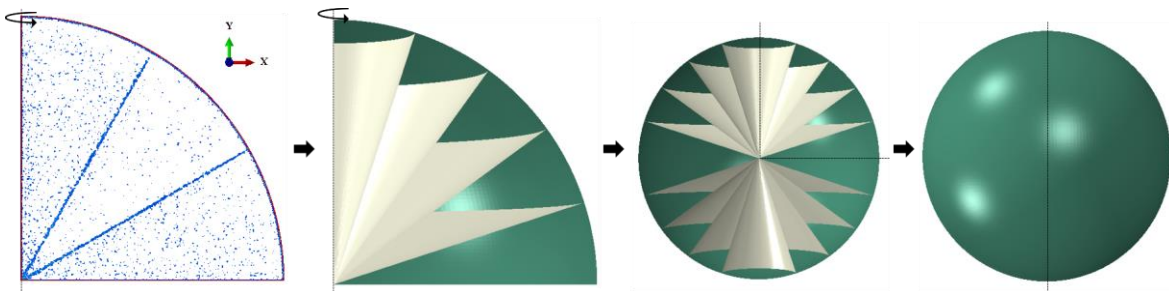


Fig. IV-9. Topology optimization findings driven design.

4.3. Stress Control Initial Validation

Stress control is central in enabling the immediate potential of air-stiffened designs. Specifically, the stress control validation presented here answers whether the hoop stress (Eq. 2 on page 6) is a valid estimate for the membrane stresses that the Helical Sphere will undergo. Since the Helical Sphere starts from a tube of constant internal radius and thickness, a plane strain analysis with a circular 2D cross-section was used to verify the

hoop stress estimates, and to find stresses at different loading conditions. Although the plane strain analysis assumes zero strain along the tube, which is strictly true for infinitely-long tubes, it provided a computationally-efficient, time-dependent solution³⁷.

4.3.1. Methodology

Table IV-5 summarizes the model parameters used in this analysis, where the dimensionality was based on the Helical Sphere with 40 revolutions. This number of revolutions given the diameter chosen was estimated to result in a $0.56 < W/B < 1$ if small displacements occur; details on W/B ratio estimates are discussed in section 5.3.

Table IV-5. Plane Strain Model Parameters

Definition	Value(s)	Comments
<i>Initial Geometry</i>	<ul style="list-style-type: none"> - Circle with 2D thickness - Inner diameter = 23.9 mm - Membrane thickness = 12.6 μm 	Based on Helical Sphere: <ul style="list-style-type: none"> - $d_i = 0.6096 \text{ m}$ - $n = 40 \text{ revs}$ - h_m of Mylar® HP film [116]
<i>Material</i>	<ul style="list-style-type: none"> - Linear isotropic - Density = 1390 kg/m³ - Modulus = 4.36E9 Pa - Yield = 110E6 Pa - Poisson's ratio = 0.38 	<ul style="list-style-type: none"> - Based on DuPont Mylar® HP 12.6 μm PET film [116] - $W/B = 0.56$ at SL
<i>Analysis</i>	<ul style="list-style-type: none"> - Abaqus implicit dynamic - Transient fidelity - Nonlinear geometric 	Time period: 0 to 10 seconds.
<i>Mesh</i>	<ul style="list-style-type: none"> - CPE8 elements - Aspect ratio = 1 - 2 through $t_m \rightarrow 23,820$ elements 	CPE8 are second-order, fully-integrated, plane strain quadrilateral elements.
<i>Loading</i>	<ul style="list-style-type: none"> - Internal tube pressure: P_{ia} - Tube portion exposed to vacuum: P_{ca} - Tube portion exposed to atmosphere: P_{oa} 	Amplitude vs time profiles on Fig. IV-11.
<i>Boundary Conditions</i>	<ul style="list-style-type: none"> - Displacement degrees of freedom at north and south vertical edges of circle 	

Fig. IV-10 shows the model used with the BC and pressure loading. The horizontal DOF of the north and south edges along the thickness were constrained, as opposed to outer

³⁷ Time-dependent solutions of 3D models typically take significant computational time (hours to days).

north and south points, to avoid stress concentration while allowing quasi-static behavior. Time-dependent pressure loading was applied at the outer-left, internal and outer-right surfaces, following the amplitude vs time curves in Fig. IV-11.

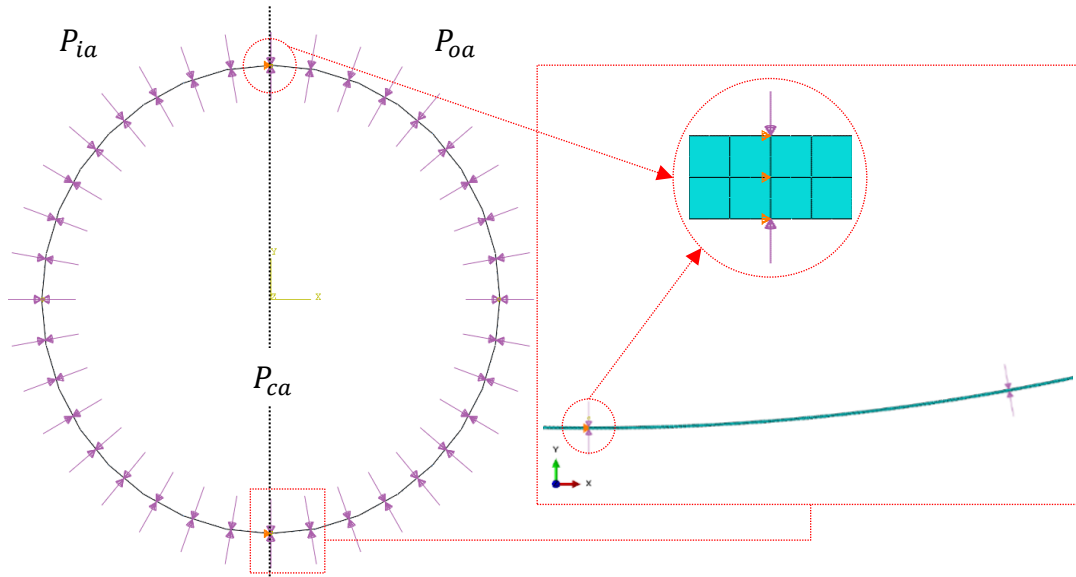


Fig. IV-10. Plane strain model diagram.

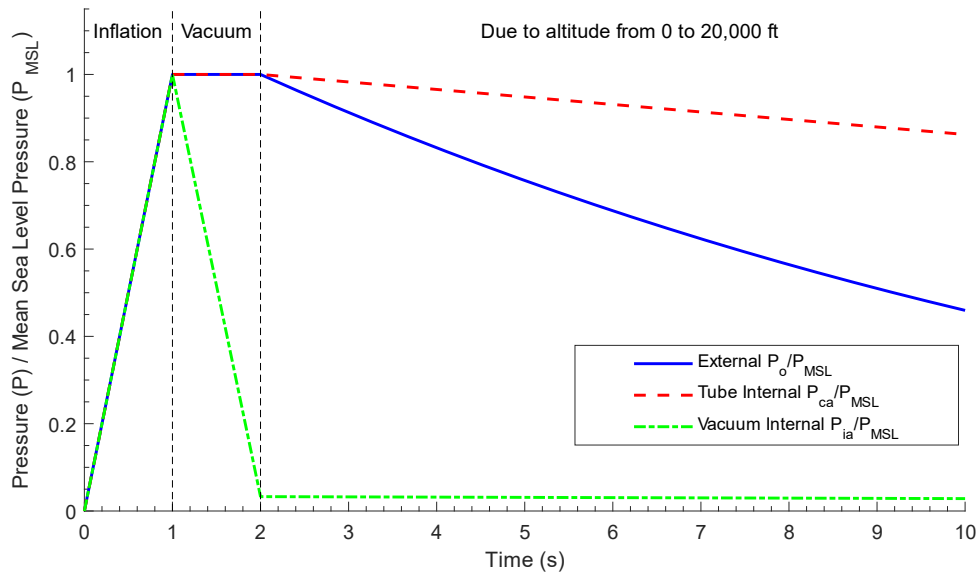


Fig. IV-11. Pressure amplitude vs time applied on plane strain model.

From time 0-1 s in Fig. IV-11, the three pressures increase linearly from zero, where the external (atmospheric) pressure was started at zero for numerical stability. From 1-2 s, the vacuum pressure decreases linearly down to the target pressure of 3.3 KPa, previously claimed as an achievable vacuum level. From 2-10 s, the internal vacuum and internal tube pressures decrease linearly (following an isovolumetric process) down to 2.9 KPa and 87.4 KPa, respectively, and the external pressure decreases exponentially following the standard atmosphere [15,39] down to 46.6 KPa. This loading history intends to mimic pressure changes during inflation, vacuum, and altitude changes, in order to capture any stress states across these, and evaluate their effects on the potential dynamics of the tube. Such loading history would be valuable for early whole flight simulations once a potential design and its parameters are identified.

Two meshes consisting of one and two elements through-the-thickness were used based on second-order, fully-integrated, plane strain quadrilateral (CPE8) elements. Both produced similar stress results, but the finer mesh allowed for improved through-the-thickness stress distribution. The structure was analyzed using the implicit dynamic technique with transient fidelity to minimize numerical dissipation effects on the solution; the types of fidelity were previously discussed.

4.3.2. Results and Discussion

The stress at the elements located next to the outer surface at the north/south and east/west locations are of interest as they reflect the constrained and free locations, respectively. The constrained locations will, in reality, be associated with adjacent revolutions joining (i.e., north and south). The free locations will be exposed to air pressure

differentials, between the vacuum and tube internal air (i.e., $\Delta P_{ci} = P_{ca} - P_{ia}$ on the west side), and between tube internal air and the atmosphere (i.e., $\Delta P_{co} = P_{ca} - P_{oa}$ on the east side). Fig. IV-12 shows the von Mises stress for the four elements located at these four locations vs time, based on the loading shown in Fig. IV-11. North and south elements were selected on the vacuum side as they will be subjected to higher loading.

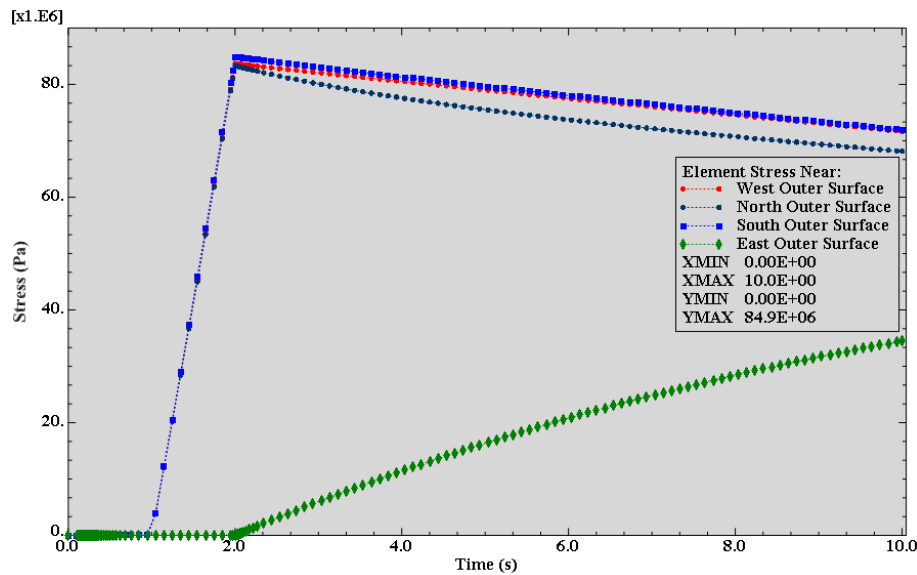


Fig. IV-12. Plane strain model von Mises stress at four locations.

From zero to one second, the stresses are nearly zero given that all pressure differentials are zero. From one to two seconds, the loading is simulated as the internal is vacuumed at sea level altitude down to the desired level, therefore all stresses, but at the east, experience maximum rate and values since the internal is being ‘vacuumed’ up to desired level. The stresses do not exceed 84.9 MPa, which is 8 MPa below the classical hoop stress prediction. The lower stresses are attributed to the geometric nonlinearity included in the analysis, rather than to the differences between the hoop and von Mises stresses, or a stiff mesh. Above two seconds, the loading is simulated as it the LTAS is rising, therefore stresses at the east (i.e., membrane exposed to the atmosphere) increase,

while decrease elsewhere. These results suggest that the classical hoop stress can and does provide a conservative and computationally simple stress estimate, which can be valuable in first approximations.

Fig. IV-12 also shows a dynamic effect that is important to keep in mind for future studies. As previously mentioned, the elements selected to represent the south, west, and north locations lie in the vacuum side (i.e., semicircle to the west). Given that stress concentration effects due to BC were essentially removed by constraining the horizontal DOF, one would expect the three elements to have similar stress results. Nonetheless, the north element's stress deviates from other two starting at two seconds, and remains lower for the rest of the time history. This deviation is attributed to a loss of geometric symmetry at two seconds. This asymmetry is observed in the contour plot shown in Fig. IV-13, with the unloaded cross-section, in light gray, superimposed for reference. The displacement magnitude scale is in meters, with a maximum of 0.23 mm at the west. The asymmetry resulted from dissimilar pressure differentials, ΔP_{ci} and ΔP_{co} , and the model being unconstrained vertically. After, ΔP_{co} (on the east side) becomes nonzero but lower than ΔP_{ci} at two seconds, which triggers a vertical acceleration, resulting in the asymmetry observed. The spike in KE observed in Fig. IV-14, although small in magnitude, resulted from the acceleration aforementioned.

Overall, this effort validated the use of the hoop stress as an initial estimate for the stress accumulated on the membrane of the Helical Sphere, that is, only if the radius of curvature of the membrane did not change considerably during loading. Nonetheless is important to note that, contrary to this 2D model, {1} any portion of the tube in the Helical

Sphere configuration is constrained north and south of its cross-section by adjacent tube revolutions, and {2} replacing the equivalent air pressure loading used here with a fluid-structure interaction drives the solution.

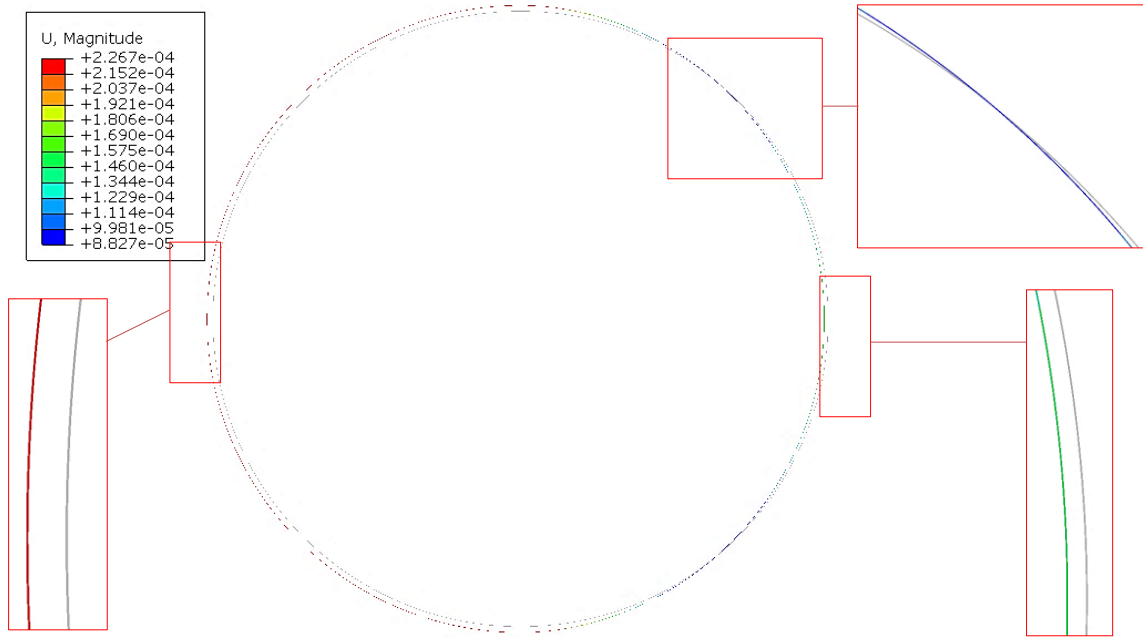


Fig. IV-13. Plane strain model displacement contour (in meters) at 2.004 seconds.

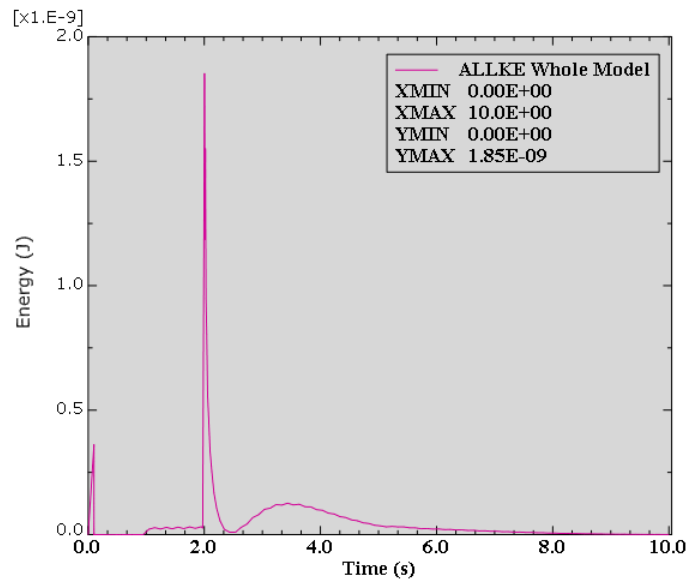


Fig. IV-14. Plane strain model kinetic energy versus time.

4.4. Summary and Conclusions

Three design paths build upon prior literature and the instability and material nonlinearity findings presented in Chapter II and III, respectively. The first path was the reconsideration of the spherical sandwich shell with the use of low-density core. It was found unfeasible due to the design not meeting stability requirements given the materials considered. More importantly, the path led to the idea of air-stiffened designs by forcing the question of whether thin barriers shaped by air can be potential VLTAS. The second path was a set of design space explorations intended to exhaust potential design avenues through topology optimization. While the author believes that the exercise of topology optimization, as presented, did not lead to designs with significant potential, it confirmed, nonetheless, the need for a low-density material that supports or shapes the membranes displacing the atmosphere which enables buoyancy, ergo supporting the use of air to stiffen the membranes. And the third path validated the use of the classical hoop stress for initial design estimates that enable membrane-to-membrane connectivity and local membrane radii determination; essential for the stress control technique of air-stiffened designs.

V. HELICAL SPHERE AIR-STIFFENED DESIGN

The trajectory summarized in Fig. I-4 and detailed in Chapters II-IV served as the foundation for the development and characterization of the Helical Sphere air-stiffened design presented in this chapter. With that in mind, this chapter starts with the Helical Sphere design description, followed by a preliminary analysis that highlights the weight and membrane stress reduction benefits motivating air-stiffened designs. The simulation framework used to develop structural solutions is then presented, emphasizing the fluid-structure interaction used to represent the air-membrane behavior. The chapter closes with an extensive discussion of a structural solution based on an axisymmetric representation of the Helical Sphere.

Regarding the use of fluid-structure interaction, the characterization of membrane structures under static pressure is often based on an equivalent loading representation that captures the pressure difference between the fluid within and that of the atmosphere, applied on a membrane representation of the film barrier [118]; that is in addition to the material, boundary conditions, etc. Such loading representation presumes fluids states are known a priori for the entire solution space of interest and therefore decoupled from the solution. This is not the case here since the state of the air within enclosing membranes shaping an air-stiffened design is neither constant during loading nor known a priori throughout the solution. Instead, the air state is coupled with the enclosing membranes, making the solution strictly dependent on the fluid-structure interaction. Therefore, the simulation framework presented in this chapter lays a path for characterizing air-stiffened designs, where air-membrane coupling is a requirement for structural characterization.

5.1. Design Description

The Helical Sphere is a particular air-stiffened design and part of the generalized representation introduced in Chapter I and Fig. I-2. The design is described as follows: consider an Archimedean spiral (Fig. V-1a), which has constant separation between revolutions. The 3D projection onto a sphere becomes a spherical helix (Fig. V-1b). A circular cross-section, normal and centered on the spherical helix, then forms the Helical Sphere (Fig. V-1c).

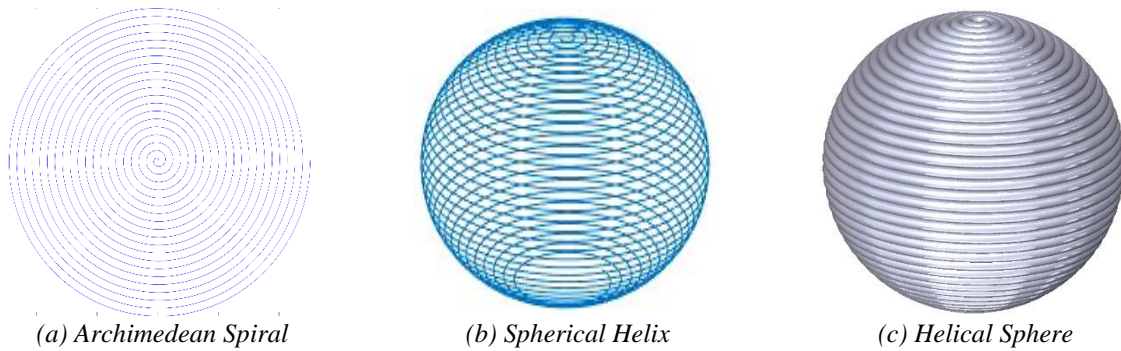


Fig. V-1. Geometric development of the Helical Sphere.

The geometry is characterized by envisioning the Helical Sphere to be formed from a single tube made of a thin membrane and filled with air. Fig. V-2 shows the tube's pressure loading and dimensional parameters. Its weight is driven by the internal spherical radius, r_i , and the tube's radius, r_{it} , thickness, h_m , and density. The tube is subjected to the vacuum pressure, P_{ia} , external air pressure, P_{oa} , and tube internal pressure, P_{ca} , where $P_{ia} \ll P_{oa}$ such that enough air weight is removed to permit buoyancy, and $P_{ca} \geq P_{oa}$ in order for the tube's membrane to remain in tension. Membrane stresses are associated with the pressure differentials, ΔP , formed by the difference between the tube internal and vacuum pressures, $\Delta P_{ci} = P_{ca} - P_{ia}$, and the difference between the tube internal and external air pressures, $\Delta P_{co} = P_{ca} - P_{oa}$. Premised on the membrane stresses following the

hoop stress (Eq. 2 on page 6) levels, the membrane stress requirement is reduced significantly compared to a perfect sphere design, where the latter stress is proportional to its overall radius. Furthermore, increasing the number of tube revolutions reduces the local stress further by consequentially reducing r_{it} , that is given any target structural radius.

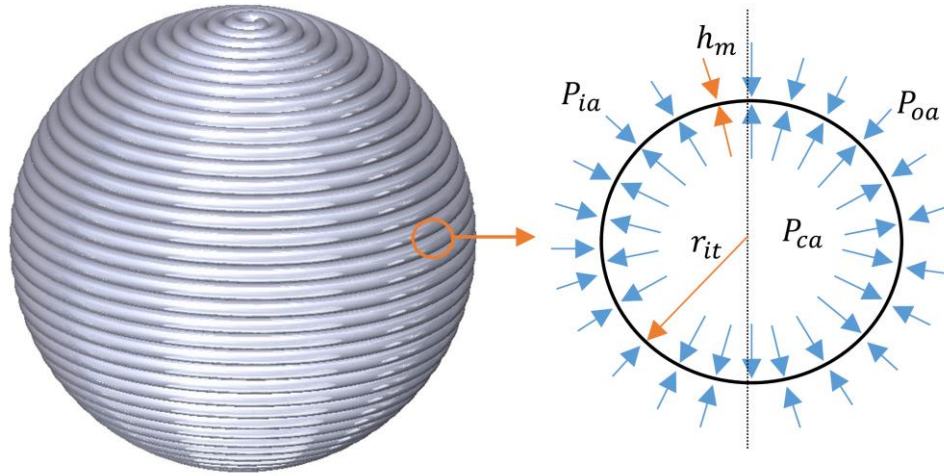


Fig. V-2. Tube loading and dimensionality.

5.2. Weight-to-Buoyancy Ratio Estimation

As described above, the Helical Sphere was developed as a 3D projection of the Archimedean Spiral (Fig. V-1a). The basic equation of the latter in polar coordinates is given by,

$$r(\phi) = a\phi \quad (34)$$

where a establishes the distance between revolutions or successive turns (i.e., rotational multiplier), angle ϕ controls the number of revolutions (n) such that $\phi = 2\pi n$, and r is the resulting radius at any given ϕ . Mapping Eq. 34 into a 2D Cartesian representation results in the following parametric equations as a function of ϕ .

$$x(\phi) = a \cos(\phi) \quad (35)$$

$$y(\phi) = a \sin(\phi) \quad (36)$$

Then, mapping the spiral onto a spherical volume produces the helix (Fig. V-1b) represented by Eqs. 37-39 [119], where a in Eqs. 35 and 36 is replaced by the summation of the internal radius r_i , membrane thickness, h_m , and internal tube radius, r_{it} . The helix evolves from $\phi = 0$ at the north pole to $\phi = \phi_{max} = 2\pi n$ at the south pole. It reaches the south pole since the rightmost cosine and sine terms guarantee that the azimuth increases from zero at the north to π at the south through $\frac{\phi}{\phi_{max}}\pi$. Such helix represents the center line of the tube envisioned to form the Helical Sphere air-stiffened design (Fig. V-1c).

$$x = (r_i + h_m + r_{it}) \cos(\phi) \sin\left(\frac{\phi}{\phi_{max}}\pi\right) \quad (37)$$

$$y = (r_i + h_m + r_{it}) \sin(\phi) \sin\left(\frac{\phi}{\phi_{max}}\pi\right) \quad (38)$$

$$z = (r_i + h_m + r_{it}) \cos\left(\frac{\phi}{\phi_{max}}\pi\right) \quad (39)$$

where: r_i = spherical radius to nearest tube surface $0 \leq \phi \leq \phi_{max} = 2\pi n$
 h_m = tube membrane thickness n = number of revolutions
 r_{it} = tube internal radius

The tube's length, L_t , is obtained with the 3D representation of the arc length, as shown in Eq. 40, which takes the derivative of the parametric Eqs. 37-39 with respect to ϕ .

$$L_t = \int_0^{\phi_{max}} \sqrt{\left(\frac{dx}{d\phi}\right)^2 + \left(\frac{dy}{d\phi}\right)^2 + \left(\frac{dz}{d\phi}\right)^2} d\phi \quad (40)$$

where: $\frac{dx}{d\phi} = (r_i + h_m + r_{it}) \left[\frac{\pi}{\phi_{max}} \cos(\phi) \cos\left(\frac{\phi}{\phi_{max}}\pi\right) - \sin(\phi) \sin\left(\frac{\phi}{\phi_{max}}\pi\right) \right]$
 $\frac{dy}{d\phi} = (r_i + h_m + r_{it}) \left[\frac{\pi}{\phi_{max}} \sin(\phi) \cos\left(\frac{\phi}{\phi_{max}}\pi\right) + \cos(\phi) \sin\left(\frac{\phi}{\phi_{max}}\pi\right) \right]$
 $\frac{dz}{d\phi} = (r_i + h_m + r_{it}) \left[-\frac{\pi}{\phi_{max}} \sin\left(\frac{\phi}{\phi_{max}}\pi\right) \right]$

The total weight is given by the summation of the tube (first term in the numerator of Eq. 41), air inside the tube (second term in the numerator of Eq. 41) and the partially-vacuumed air inside (third term in the numerator of Eq. 41) weights. The buoyancy (denominator of Eq. 41), or weight of the displaced atmospheric air, is given by the spherical volume of radius $r_i + h_m + r_{it}$ plus half of the tube's volume, all multiplied by the atmospheric air density.

$$\frac{W}{B} = \frac{2\pi r_{it} h_m L_t \rho_m + \pi r_{it}^2 L_t \rho_{ca} + \frac{4}{3} \pi r_i^3 \rho_{ia}}{\left[\frac{4}{3} \pi (r_i + h_m + r_{it})^3 + \frac{1}{2} \pi r_{it}^2 L_t \right] \rho_{oa}} \quad (41)$$

where: r_i = spherical radius to nearest tube surface $0 \leq \phi \leq \phi_{max} = 2\pi n$
 r_{it} = tube internal radius n = number of revolutions
 h_m = tube membrane thickness L_t = tube length
 ρ_m = membrane density
 $\rho_{ca}, \rho_{ia}, \rho_{oa}$ = air densities in the tube, in the innermost cavity (partially-vacuum), and of the atmosphere, respectively.

Eqs. 40 and 41 produce an estimate of the Helical Sphere's W/B ratio, when assumed that the structure will maintain its initial geometry through the loading history. While the latter assumption makes the W/B ratio estimate optimistic by dismissing the effects associated with stress accumulation along joints of adjacent tubes and overall structural stability, it allows the membrane properties and geometry to be inputted into the FEM to be based upon a $W/B < 1$, as estimated by these equations.

Furthermore, the stress control benefits upon which the Helical Sphere was built is evident in the relationship between n and r_{it} . The hemispherical cut shown in Fig. V-3 served to develop this relationship, given by Eq. 42, and based on γ and the sides of the triangle presented in Fig. V-3, such that $r_{it} = r_{it}(n, r_i, h_m)$ and $\gamma = \pi/(2n)$. For any given

r_i , there is a tradeoff among h_m , r_{it} and n that needs to be considered. This tradeoff is the topic of the preliminary analysis discussed next.

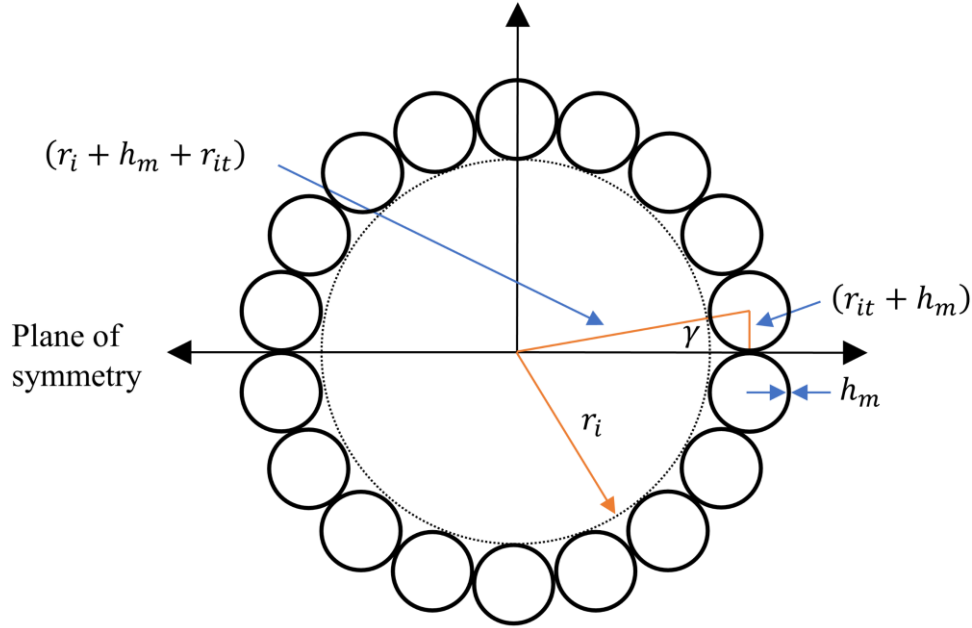


Fig. V-3. Hemispherical cut diagram.

$$\sin(\gamma) = \frac{r_{it} + h_m}{r_i + h_m + r_{it}} = \sin\left(\frac{\pi}{2n}\right) \Rightarrow r_{it} = \left[\frac{\sin\left(\frac{\pi}{2n}\right)}{1 - \sin\left(\frac{\pi}{2n}\right)} \right] r_i - h_m \quad (42)$$

5.3. Preliminary Analysis

The preliminary analysis explores the effect that varying the number of revolutions has on the tube diameter (Eq. 42), tube length (Eq. 40), W/B ratio (Eq. 41) and membrane hoop stress (Eq. 2 on page 6), given a Helical Sphere of internal radius equal to 0.3048 m and made from a 12.6 μm -thick PET film. This membrane thickness is commonly available in commercial PET and PEN films (e.g., Mylar HP) [50,116].

Top to bottom, Fig. V-4 shows the tube diameter, and the resulting tube length, W/B ratio and tube local hoop stress vs n . The W/B ratio and stress are calculated for sea level altitude and vacuum pressure of 25 torr (3333 Pa). It is clearly beneficial to have less

revolutions from a manufacturing perspective. Nonetheless, material yielding will limit the maximum tube diameter. Furthermore, W/B ratio is reduced with increased n since the tube's weight grows linearly with length, but quadratically with diameter. The W/B ratio, in this case, tends to 0.5 for large n .

It is important to realize that the membrane thickness and yield stress (both part of the selected film product), as well as the vacuum pressure, dictate the tube diameter required to meet desired hoop stress levels independent of the Helical Sphere internal diameter. A $n = 40$, for example, provides a safety factor³⁸ equal to 1.18, with tube diameter of 23.9 mm and length of 50.7 m, resulting in a $W/B = 0.56$ at SL, when the internal radius is 0.3048 m. These results show the potential alluded to the Helical Sphere design when it was introduced, but fail to account for any displacement that would reduce the buoyancy and deviate membrane stresses from hoop stress estimates.

³⁸ Based on a yield stress for Mylar® C of 110 MPa as shown in Table II-2 [48].

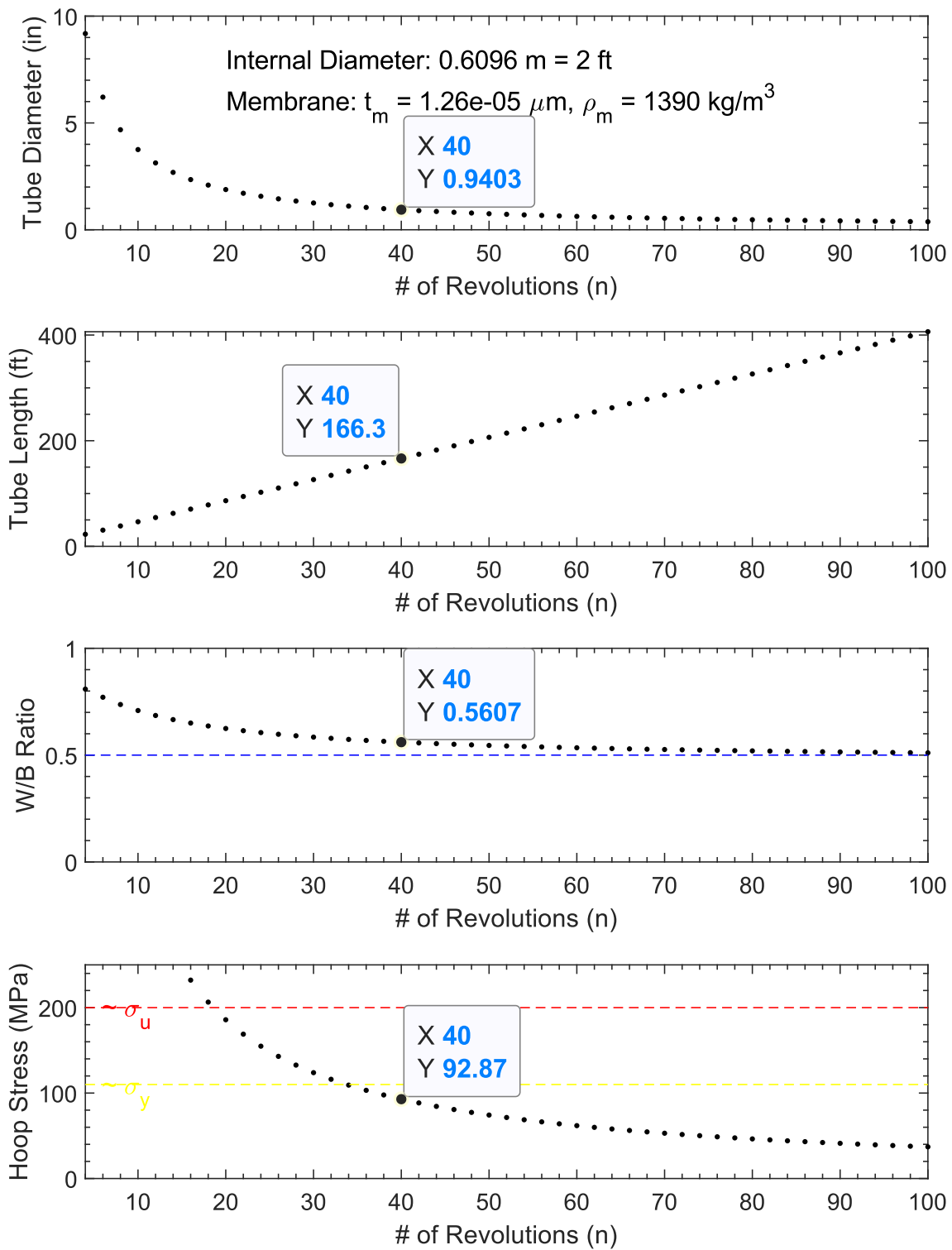


Fig. V-4. Tube diameter, tube length, weight-to-buoyancy ratio, and hoop stress vs number of revolutions.

5.4. Structural Simulation Framework

The structural simulation of air-stiffened designs, as fluid-filled membrane structures undergoing large displacements and rotations, necessitated the inclusion of geometrically nonlinear effects, fluid-structure interactions, and surface-to-surface contact, among others. Before presenting specific structural solutions associated with the Helical Sphere, the main modeling components used to generate such solutions are described below in the context of FEM with SIMULIA™ Abaqus 2016³⁹ [120].

5.4.1. Geometric Nonlinearity

Membranes undergoing large deformations, and/or subjected to contact constraints and pressure loads, inherently produce geometrically nonlinear responses as equilibrium states are a function of the deformed [121]. Geometrically nonlinear analyses {1} use the complete strain tensor and material local directions to ensure that the element strain components are appropriately decoupled from rigid body motion [62], {2} introduce “pressure load stiffness” components that couple nodal displacements to nodal forces associated with pressure loads to account for the load directional dependence on the deformed geometry, and {3} include thickness variation of shell and membrane elements by assuming plane stress conditions [58,122].

5.4.2. Hydrostatic Representation of Fluid-Structure Interactions

The presence of air enclosed by adjacent membranes (i.e., air-stiffening) couples the membrane deformation to the pressure exerted by the air contained within it, making

³⁹ Finite element solutions of an inflated circular membrane using these components were compared to Bouzidi et al. [126] analytical solution and Coelho et al. [127] numerical solutions with displacements within 6-8% of the referenced solutions. The solutions found are discussed in section B.1 (Appendix B).

the structural response a function of this coupled interaction along with any external loading. In the interest of quasi-static solutions (i.e., a timescale where the membrane and the fluid inertial effects become negligible), the common coupled Eulerian-Lagrangian fluid-structure interaction is replaced by a simpler hydrostatic fluid formulation that couples the deformation of the air-filled membranes to the pressure of the air contained within. Abaqus refers to this interaction as *surface-based fluid cavity*. The description and theoretical development of this interaction are found in Ref. [59] and Ref. [58], respectively, and are summarized here in reference to its capabilities to represent the air-membrane interactions driving the structural response of air-stiffened designs.

The surface-based fluid cavity capability couples the state of a fluid to the structure by constraining the volume of the cavity to be approximately equal to the volume of the fluid enclosed (i.e., $V - \bar{V}$). A cavity is defined by a set of surfaces that enclose a fluid, where such set is discretized by the user with structural FE (e.g., structural membrane elements). Abaqus, in turn, discretizes the volume occupied by the fluid with *hydrostatic fluid elements*, following the nodal locations of the structural elements. The fluid behavior is then represented by a single point, referred to as the *cavity reference node*, which contains the fluid pressure inside the cavity as its single degree of freedom. The latter statement indicates hydrostatic behavior by implying that the fluid state is, by definition, uniform across the cavity and therefore negates any spatial variation associated with fluid flow. Nonetheless, it does not constrain the fluid state to be constant during the load (or time) history. The development of the volume calculations driving the constraint and its implementation follow.

Cavity Volume Calculations

The cavity volume, V , is a geometrical representation of the cavity calculated via the volume elements representing the cavity. Specifically, Abaqus covers the cavity surface with facets that are tied to the structural nodes given by a user-specified structural mesh.

Those surface facets are referred to as hydrostatic volume elements, and require the coordinates of the reference node to form the volume element (e.g., a surface facet becomes a pyramid by adding the reference node as shown in Fig. V-5). Each element volume is obtained using shape or

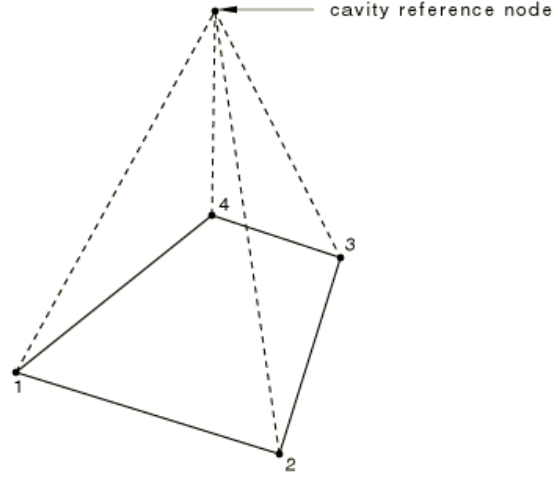


Fig. V-5. Hydrostatic volume element [58].

interpolation functions of the surface facets and the position of the cavity reference node, with their summed contribution resulting in V .

For example, the coordinate vector, \mathbf{x} , of any point along the base of the hydrostatic volume element (Fig. V-5) is described by,

$$\mathbf{x} = \sum_N N^N(g, h) \mathbf{x}^N \quad (43)$$

where N^N are the isoparametric shape functions for the base as a function of parametric coordinates g and h , and \mathbf{x}^N are the vectors of nodal coordinates. The element normal, \mathbf{n} , times an infinitesimal area, dA , is then given by,

$$\mathbf{n}dA = \left(\frac{\partial \mathbf{x}}{\partial g} \times \frac{\partial \mathbf{x}}{\partial h} \right) dg dh = \left(\sum_N \frac{\partial N^N}{\partial g} \mathbf{x}^N \times \sum_N \frac{\partial N^N}{\partial h} \mathbf{x}^N \right) dg dh \quad (44)$$

where the term inside the parenthesis represents the Jacobian (i.e., determinant of the Jacobian matrix). The element infinitesimal volume, dV^e , associated with the infinitesimal area (Eq. 44) is then given by,

$$dV^e = \frac{1}{3}(\mathbf{x}_R - \mathbf{x}) \cdot \mathbf{n}dA = \frac{1}{3}(\mathbf{x}_R - \mathbf{x}) \cdot \left(\sum_N \frac{\partial N^N}{\partial g} \mathbf{x}^N \times \sum_N \frac{\partial N^N}{\partial h} \mathbf{x}^N \right) dg dh \quad (45)$$

where \mathbf{x}_R is the coordinate vector of the cavity reference node forming the hydrostatic volume element. Subsequently, the element volume, V^e , and the cavity volume, V , are obtained by Eq. 46 and Eq. 47, respectively.

$$V^e = \frac{1}{3} \int_{-1}^1 \int_{-1}^1 (\mathbf{x}_R - \mathbf{x}) \cdot \left(\sum_N \frac{\partial N^N}{\partial g} \mathbf{x}^N \times \sum_N \frac{\partial N^N}{\partial h} \mathbf{x}^N \right) dg dh \quad (46)$$

$$V = \frac{1}{3} \sum_e \left[\int_{-1}^1 \int_{-1}^1 (\mathbf{x}_R - \mathbf{x}) \cdot \left(\sum_N \frac{\partial N^N}{\partial g} \mathbf{x}^N \times \sum_N \frac{\partial N^N}{\partial h} \mathbf{x}^N \right) dg dh \right] \quad (47)$$

This volume is strictly a geometrical representation of the cavity, effectively tying the behavior of the membrane, or generally that of the structural surfaces, to the fluid elements' geometry. The fluid volume is the other variable driving the volume constraint.

Fluid Volume Calculations

Abaqus has both incompressible and compressible representations of the fluid. Given the use of air in this application, the incompressible representation is discussed. Therefore, compressibility is represented via the ideal gas law, such that the fluid volume, $\bar{V} = \bar{V}(\theta, m, P)$, is given by,

$$\bar{V}(\theta, m, P) = mR \frac{(\theta - \theta_A)}{(P + P_{oa})} \quad (48)$$

where m is the total fluid mass, R is the specific gas constant, θ is the fluid temperature, θ_A is the temperature at absolute zero, P is the fluid gauge pressure, and P_{oa} is the reference pressure (e.g., external air pressure). The density of the fluid, ρ , is then given by,

$$\rho(\theta, P) = \frac{(P + P_{oa})}{R(\theta - \theta_A)} \quad (49)$$

Since the cavity is discretized with hydrostatic volume elements, each of these contributes to the fluid volume, $\bar{V}^e = \bar{V}^e(P, \theta, m^e)$, where m^e is their contribution to the fluid mass. Given that Eq. 49 is based on global cavity properties (i.e., the density is uniform across the fluid cavity and therefore across the volume elements), $\bar{V}(\theta, m, P)$ is expressed as a summation of the elemental mass, such that:

$$\bar{V}(\theta, m, P) = \sum_e \bar{V}^e(P, \theta, m^e) = \frac{\sum_e m^e}{\rho(P, \theta)} = \frac{m}{\rho(P, \theta)} = mR \frac{(\theta - \theta_A)}{(P + P_{oa})} \quad (50)$$

Is important to realize that the hydrostatic assumption enables the decoupling of the fluid pressure and density from elemental properties by imposing uniformity. An underlying result of the fluid state equation is its resulting volume-pressure compliance, $d\bar{V}/dP$. The latter is presented in Eq. 51, and its negative reciprocal times the volume results in the bulk modulus, a measure of a fluid resistance to compression [123]. Such relationship dictates volume-pressure changes of the air-filled membrane.

$$\frac{d\bar{V}}{dP} = mR \frac{(\theta - \theta_A)}{(P + P_{oa})^2} \quad (51)$$

Fluid-Structure Coupling

The fluid-structure coupling occurs by augmenting the potential energy, or virtual work expression, of the structure without the fluid, $\delta\Pi$, with the cavity pressure contribution and the volume constraint: $V - \bar{V} = 0$, where V and \bar{V} are based on Eqs. 47

and 50, respectively. The augmented virtual work, $\delta\Pi^*$, is shown in Eq. 52, where the negative signs imply that a decrease in cavity volume increases the energy of the fluid, and vice versa. $\delta\Pi^*$ represents a mixed formulation, where displacements associated with the structural mesh and the fluid pressure in the cavity are primary variables.

$$\delta\Pi^* = \delta\Pi - P\delta V - \delta P(V - \bar{V}) \quad (52)$$

Plainly, the virtual work of the structure is constrained by $V - \bar{V} = 0$ along with any other constraints (e.g., contact) and solved as a minimization problem. The minimum energy solution with the volume constraint results in a cavity pressure that ensures the fluid volume is approximately equal to the cavity volume. Any change in P from its previous state is reflected as a distributed pressure load on the structural mesh. Volumes within $\delta\Pi^*$ (Eq. 52) can be represented in discretized form as the sum of each of the hydrostatic volume element contributions, such that:

$$\delta\Pi^* = \delta\Pi - P \sum_e \delta V^e - \delta P \left[\sum_e V^e - \sum_e \bar{V}^e \right] = \sum_e [\delta\Pi^e - P\delta V^e - \delta P(V^e - \bar{V}^e)] \quad (53)$$

Comments of Approach Validity

The validity of the ideal gas law as an equation of state dictating the fluid volume is of practical concern. The ideal gas model yields a simple P , \bar{V} and θ relation for a fluid behavior by assuming fluid molecules do not interact. Nonetheless, said interactions do exist and influence the behavior of real gases. The virial equation shown in Eq. 54 is one of many models that intend to represent the influence of such interactions, where the compressibility factor, Z , is a non-dimensional parameter, and R is the gas constant [124].

$$Z = \frac{PV}{R\theta} \quad (54)$$

A compressibility factor $Z \neq 1$ indicates a deviation from ideality, where molecular interactions play a significant role in the gas behavior; notably at low temperatures or high pressures. In the case of air, Z fluctuates within $0.0052 \leq Z \leq 2.11$ for the following temperatures and pressure ranges: $75 \leq T \leq 1000$ K and $100 \leq P \leq 50,000$ kPa [125]. In the context of the results shown throughout this dissertation, $0.9987 \leq Z \leq 0.9999$, indicating that the ideal gas law was an accurate state equation for the air inside the cavities during the load histories considered. The former statement does not speak of the accuracy of the hydrostatic representation of the fluid-structure interaction, but rather cements the use of such equation of state in this context, as well as provides some level of confidence on its validity for coupled Eulerian-Lagrangian analysis.

Regarding accuracy, FE solutions of an inflated circular membrane using the simulation components presented throughout this section (5.4) were compared to Bouzidi et al. [126] analytical solution and Coelho et al. [127] numerical solutions with displacements within 6-8% of the referenced solutions. The solutions found are discussed in section B.1 (Appendix B).

Overall, is important to keep in mind that this formulation removes any fluid flow described in the Navier-Stokes and Euler equations, including any spatial variation of the pressure and body forces, and replaces them with a pressure-volume coupling based on a uniform fluid state (represented by the ideal gas law for compressible fluids). Compared to a generalized fluid-structure interaction, this formulation removes any fluid inertial effects, makes temperature a user-defined variable in implicit solutions, and requires the cavity to remain fully-filled at all times. Such limitations make the hydrostatic formulation presented

a simple yet useful coupling for the prediction of quasi-static responses within the Lagrangian description. Other hydrostatic formulations recently developed to represent gas-structure interactions include work on hyperelastic membranes [128] and elastic membranes and shells [129,130]. A notable difference in the latter formulations is the inclusion of the hydrostatic interaction as a rank-one update of the stiffness matrix without the need for volume elements. Subsequently, the inverse of the updated stiffness matrix is obtained efficiently by using the inverse stiffness matrix of the structure without the interaction via Sherman–Morrison formula [131].

5.4.3. Contact

Contact is another simulation component critical to the development of the solution characterizing the structural behavior of air-stiffened designs that are subjected to relatively low-pressures along its innermost air cavity. As an equivalent (negative) pressure load is applied to the innermost cavity to represent a reduction of air pressure (i.e., air removal to create a partial vacuum), the lack of inherent bending stiffness is replaced by the fluid-structure interaction and membrane-to-membrane contact, the latter increasingly relevant as significant displacements occur. The equilibrium state under loading is ergo dependent on the normal and tangential contact interactions.

In the case of normal contact, *hard* contact represents the type of pressure-overclosure relationship in which no penetration between surfaces is desired [59]. The equilibrium state, or more precisely, the potential energy (or virtual work expression), $\delta\Pi^*$, is modified to account for this interaction, such its addition to Eq. 52 results in:

$$\delta\Pi^* = \delta\Pi + [-P\delta V - \delta P(V - \bar{V})] + [h\delta P_c + P_c\delta h] \quad (55)$$

where: P_c = contact pressure
 h = overclosure or distance between adjacent surfaces

such that: $P_c = 0$ for $h < 0$ (open)
 $h = 0$ for $P_c > 0$ (contact)

Eq. 55 represents the first variation of the potential energy of a structure subjected to the hydrostatic representation of fluid-structure interaction (detailed section 5.4.2) and normal *hard* contact [58]. This equation is a fundamental representation of the state of equilibrium driving the structural characterization of air-stiffened designs, as defined in this document. The enforcement of the hard contact condition depends on the enforcement method used. Regarding the general contact algorithms used in the implicit and explicit solutions presented in this dissertation, hard contact was strictly enforced in implicit solutions (Chapter V) through Lagrange multipliers, and approximated in explicit solutions (Chapter VI) through the penalty method⁴⁰.

Fig. V-6 demonstrates the importance of contact in the Helical Sphere solution. Fig. V-6a shows the initial contact conditions established among adjacent tube revolutions (or air-stiffened toroidal membranes in an axisymmetric representation), given by a point tie at the neighbor points coincident with the sphere that passes through all the circle midpoints. Fig. V-6b then shows the development of contact as each portion of the tube (or each air-stiffened toroidal membrane in an axisymmetric representation) progressively deforms and subsequently “pinned”. While this diagram is a general representation of the

⁴⁰ Section 13.2 and 13.3 of Ref [72] provides a generic description of both enforcement methods. Its applications and effects on the solution are detailed in Refs. [58,59]. The Lagrange multipliers method solves the contact constraints exactly, avoiding overclosures that can result when approximate methods are used, such as the penalty method, while typically at a higher computational cost [157].

behavior observed in the solution (presented and discussed in section 5.5.3), it shows the solution dependency on the contact areas developed as deformation occurs; deformation that is a function of air-membrane interactions.

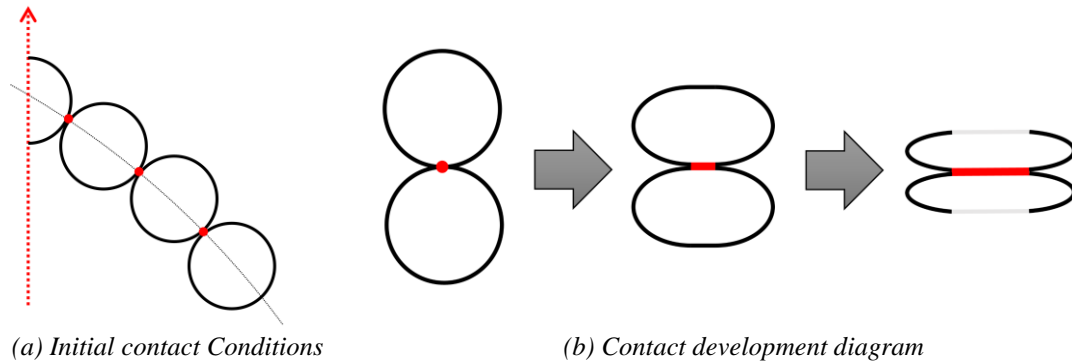


Fig. V-6. Helical sphere contact diagram.

Regarding the solutions presented in this chapter, normal hard contact is strictly enforced, negating any surface-to-surface penetration, and without separation after contact. The latter was selected to understand how nearby surface progressively contacts each other as membranes deform. A “rough” tangential condition was chosen, which represents an infinite friction coefficient, negating surface-to-surface slipping. Numerically, both constraints are added to the potential energy and strictly enforced with Lagrange multipliers. Therefore, contact is defined such that surfaces are tied upon contact considering that such solutions are expected to {1} provide upper stress limits since shear stress accumulation resulting from the tangential contact is not restricted to an upper limit that a friction model would have and {2} maximize any structural stability associated with these. The general contact algorithm in Abaqus was used with these contact definitions and surface-to-surface discretization, as well as accounting for membrane thickness. While these definitions were used in the quasi-static results included below, solutions with a

frictionless tangential condition and separation after contact allowed were also computed. These solutions, albeit not included, provided similar stresses and collapse loads.

5.4.1. Solver Selection

As stated during the discussion of fluid-structure interaction, the goal is to find quasi-static solutions that represent long-time responses. Since convergence limitations negate the use of static solvers when faced with multiple sources of nonlinearities, such as geometric, fluid-structure interaction and contact, the implicit dynamic solver with the backward Euler operator framework within Abaqus/Standard was selected to simulate quasi-static solutions. This is the same solver used to develop the collapse solutions described in section 3.2.4.

Geometric nonlinearity (section 5.4.1), hydrostatic representation of fluid-structure interactions (section 5.4.2), contact interactions (section 5.4.3) and solver selection (section 5.4.1) were presented here as they drive the simulation framework from which quasi-static solutions of air-stiffened designs are produced. The following section introduces an axisymmetric representation of the Helical Sphere, from which structural responses were obtained using this framework.

5.5. Axisymmetric Representation of the Helical Sphere

Removing the slope of the spiral turn, or driving the spiral slope to infinite, replaced the Helical Sphere (Fig. V-7a) with a set of toroidal membranes connected to each other (Fig. V-7b), which allowed for an axisymmetric representation (Fig. V-7c). Such representation, in turn, enabled the use of axisymmetric FE to describe the structure, trading geometric accuracy for increased mesh density for a given model size. This

computationally efficient modeling description drove the use of an axisymmetric representation to produce structural solutions.

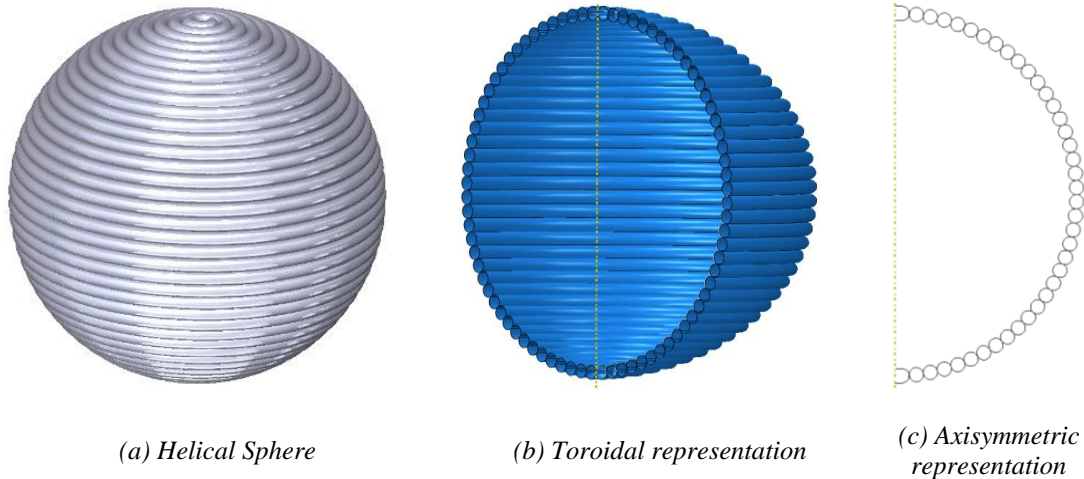


Fig. V-7. Axisymmetric representation of the Helical Sphere.

The toroidal representation (Fig. V-7b) consists of a set of toroidal membranes with constant minor radius (i.e., the radius of the circle forming the torus, r_{it}) and varying major radius (i.e., the distance from a center of the torus to the center of the circle forming the torus, r_i), such that the set can be bonded along a spherical mandrel and completed with two spherical membranes at the poles; the radius of the spherical membranes is assumed to be equal to the minor radius of the tori.

A toroidal representation composed of eight toroidal and two spherical membranes is represented in Fig. V-8 [122]. Each membrane is subjected to internal pressure, P_{ia} , external pressure, P_{oa} , and tori internal pressure, P_{ta} , where $P_{ia} \ll P_{oa}$ and $P_{ta} \geq P_{oa}$ in order for the membranes to remain in tension; note that the cavity pressure P is a gauge pressure such that $P = P_{ta} - P_{oa}$. The membrane mass is driven by r_i , r_{it} , its thickness, h_m , and its density, where air masses associated with P_{ia} and P_{ta} are part of the overall

design mass. It is important to distinguish that changes in P_{ia} account for changes in mass, while changes in P_{ta} resulting from air compressibility and membrane deformation do not.

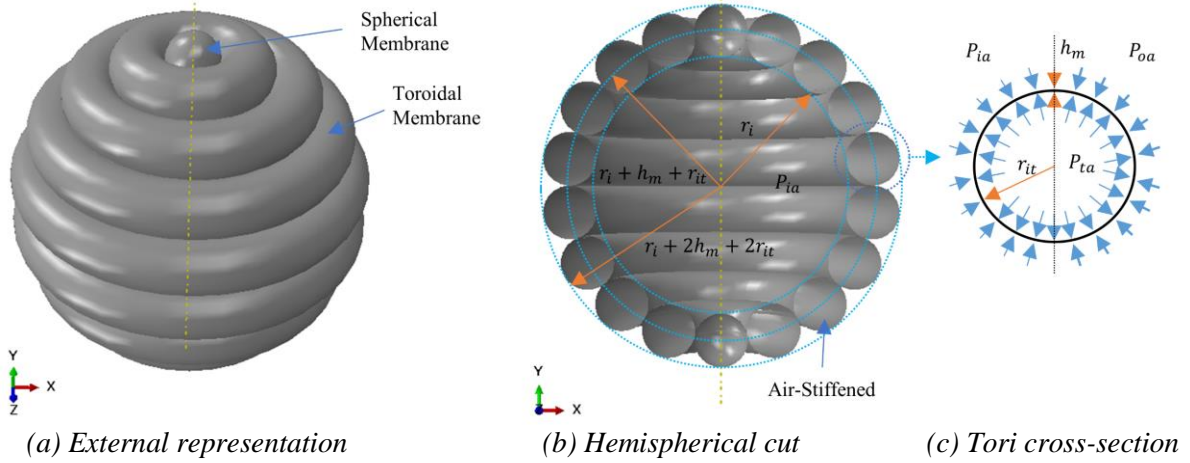


Fig. V-8. Toroidal representation diagram ($n = 9$).

5.5.1. Weight-to-Buoyancy Ratio Estimation

The W/B ratio of the toroidal representation is developed by considering that the volume of a torus, V_T , and surface area, S_T , is given by [132],

$$V_T = 2\pi^2 r_{it}^2 c \quad (56)$$

$$S_T = 4\pi^2 r_{it} c \quad (57)$$

where c is the distance from the axis of revolution to the center of each torus. Following Fig. V-3, $c = (r_i + h_m + r_{it}) \cos(\theta)$, where $\theta = \gamma(2j - 1) = \pi/(2n) (2j - 1)$ ⁴¹. Since V_T and S_T represent the air volume within and the surface area of the membrane,

⁴¹ Based on the relationship between the membrane radius and the number of revolutions in Eq. 42. Note in Fig. V-3 and Eq. 42 that n must be an odd integer in order to represent the spherical membranes at the poles. Odd n values result in the Helical Sphere's 'tube' being collocated along two poles when considered in 2D. Enforcing axisymmetry on the 2D representation, in turn, generates spherical membranes (i.e., air-filled spheres) at such poles. In other words, restricting n to odd values preserves the initial membrane radius (i.e., r_{it}) across the structure.

respectively, the air volume of each torus, $V_{ta}^{(j)}$, and the membrane volume of each torus, $V_m^{(j)}$, are respectively given by,

$$V_{ta}^{(j)} = 2\pi^2 r_{it}^2 (r_i + h_m + r_{it}) \cos\left(\frac{\pi}{2n} (2j - 1)\right) \quad (58)$$

$$V_m^{(j)} = 4\pi^2 r_{it} h_m (r_i + h_m + r_{it}) \cos\left(\frac{\pi}{2n} (2j - 1)\right) \quad (59)$$

Given Eqs. 58 and 59, and considering that the structure is also symmetric about the $x - z$ plane (horizontal plane) and the poles contain air-filled spherical membranes, the total air volume inside tori, V_{ta} , and the total membrane volume, V_m , are respectively given by,

$$V_{ta} = 2 \sum_{j=1}^{\frac{n-1}{2}} V_{ta}^{(j)} + \frac{8}{3} \pi r_{it}^3 \quad (60)$$

$$V_m = 2 \sum_{j=1}^{\frac{n-1}{2}} V_m^{(j)} + 8\pi r_{it}^2 h_m \quad (61)$$

Placing the right sides of Eqs. 58 and 59 into Eqs. 60 and 61 results in Eqs. 62 and 63, respectively, where the thickness is assumed uniform across the toroidal membranes.

$$V_{ta} = 4\pi^2 r_{it}^2 (r_i + h_m + r_{it}) \sum_{j=1}^{\frac{n-1}{2}} \left[\cos\left(\frac{\pi}{2n} (2j - 1)\right) \right] + \frac{8}{3} \pi r_{it}^3 \quad (62)$$

$$V_m = 8\pi^2 r_{it} h_m (r_i + h_m + r_{it}) \sum_{j=1}^{\frac{n-1}{2}} \left[\cos\left(\frac{\pi}{2n} (2j - 1)\right) \right] + 8\pi r_{it}^2 h_m \quad (63)$$

The quantity inside the cosine terms, θ , ranges in odd increments of $\pi/(2n)$ as follows: $\pi/(2n) \leq \theta \leq \pi/2 - \pi/n$, where $j = 1, 2, \dots, n - 1$.

The total weight is approximated by the summation of the weight of the air inside the tori (first term in the numerator of Eq. 64), toroidal membranes (second term in the numerator of Eq. 64), and the partially-vacuumed air inside (bracketed term in the

numerator of Eq. 64). The buoyancy (denominator of Eq. 64), or weight of the displaced atmospheric air, is approximated as the spherical volume of radius $r_i + h_m + r_{it}$ plus half of the tori's volume, all multiplied by the atmospheric air density. The air densities ρ_{ta} , ρ_{ia} , and ρ_{oa} are associated with P_{ta} , P_{ia} , and P_{oa} shown in Fig. V-8.

$$\frac{W}{B} = \frac{V_{ta}\rho_{ta} + V_m\rho_m + \left[\frac{4}{3}\pi(r_i + h_m + r_{it})^3 - \frac{1}{2}V_{ta}\right]\rho_{ia}}{\left[\frac{4}{3}\pi(r_i + h_m + r_{it})^3 + \frac{1}{2}V_{ta}\right]\rho_{oa}} \quad (64)$$

The inversely proportional relationship between r_{it} and n (Eq. 42) once again exemplifies the motivation for characterizing air-stiffened designs as VLTA envelopes. The effect of such relationship in the W/B ratio (Eq. 64) and hoop stress (Eq. 2 on page 6) is presented in Fig. V-9 and Fig. V-10, respectively, as the number of tori increases from nine to 197. The W/B ratio and hoop stress were plotted for $P_{ia} = [0.5, 0.25, 0.03]P_{oa}$, and $P_{ta} = [1, 2]P_{oa}$. Both figures set $r_i = 0.3048$ m, $h_m = 12.6 \mu\text{m}$, and $\rho_m = 1390 \text{ kg/m}^3$, and P_{oa} and ρ_{oa} are in accordance with the U.S. Standard Atmosphere at sea level [15].

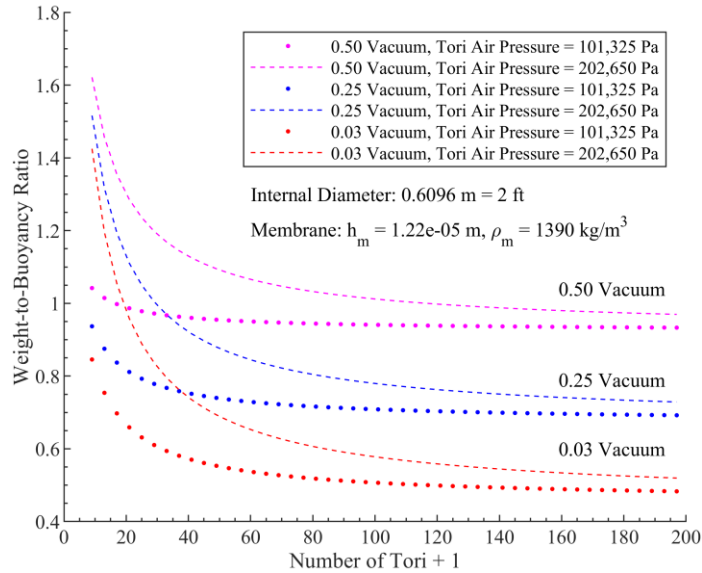


Fig. V-9. Weight-to-buoyancy ratio versus number of tori + 1.

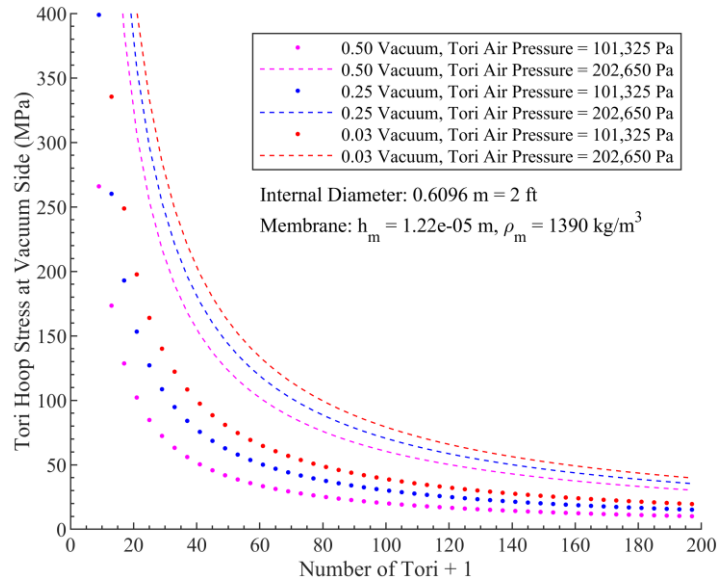


Fig. V-10. Hoop stress versus number of tori + 1.

A $n = 41$ was chosen to simulate the structural response since Fig. V-9 suggests a $W/B \cong 0.98$ at 50% and $W/B \cong 0.6$ at 3% remainder of a (full) vacuum when $P_{ta} = P_{oa}$. Furthermore, the resulting r_{it} produces $\sigma \cong 100$ MPa at 3% remainder of a vacuum, which is below the proportional limit of typical PET films (e.g., 110 MPa for Mylar® C [48]).

Generally, increasing the number of tori (or number of revolutions forming the Helical Sphere) reduces the W/B and stress. These benefits become evident as tori air pressure increases due to the cubic dependence of air volume on tori radius combined with the quadratic dependence of membrane volume on tori radius. Axisymmetric half cuts of the toroidal representation with $n = 9$ (in black), 41 (in red), and 161 (in blue) are superimposed in Fig. V-11 to visually represent the weight reduction as n increases. For example, an $n = 9$ toroidal representation results in a $W/B (P_{ta} = P_{oa}) = 0.97$ and $W/B (P_{ta} = 2P_{oa}) = 1.6$, a 50% difference. But the W/B ratio only varies by 8% between the two P_{ta} variations when $n = 161$. In terms of scalability, membrane thickness, as well

as P_{ta} and P_{oa} , drive the initial tori radius required to meet desired hoop stress levels, independent of the overall size of the structure.

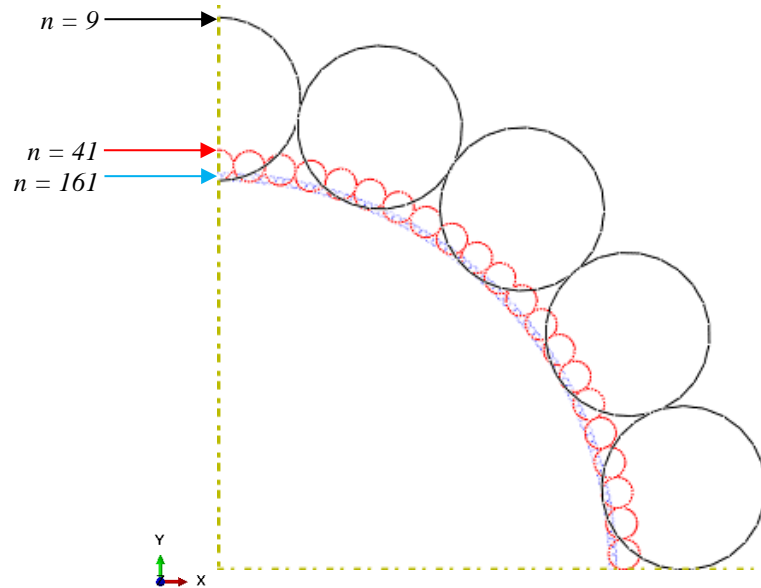


Fig. V-11. Superimposed axisymmetric cuts of toroidal representation for $n = 9$ (black), 41 (red), and 161 (blue).

5.5.2. Model Description

As diagramed in Fig. V-12, the toroidal representation was modeled axisymmetrically about the y axis, with x - z plane symmetry and $n = 41$ (i.e., representing 40 tori and two spheres). The internal radius was set to $r_i = 0.3048$ m, and all tori membrane thicknesses were set to $h_m = 12.6$ μm . The membrane material was assumed to be isotropic and to follow Hooke's law with modulus $E = 4.36$ GPa, Poisson's ratio $\nu = 0.38$, and $\rho_m = 1390$ kg/m^3 ; these material properties were selected to represent typical properties found in PET film at room temperature. A $n = 41$ was selected for r_i and h_m because the resulting r_{it} produces a hoop stress below 100 MPa, which is near the typical PET yield limit.

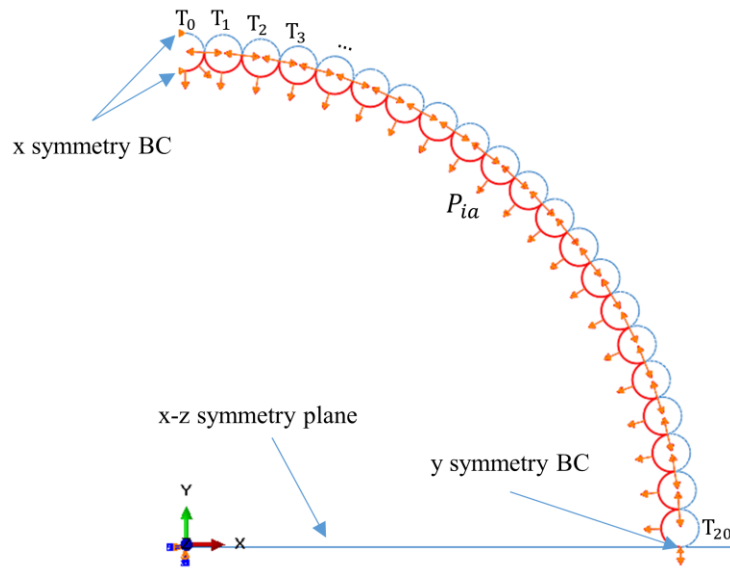


Fig. V-12. Axisymmetric $n = 41$ model diagram.

The x-z plane was represented as a rigid line along the x axis to account for any contact along said plane, including the simulation of symmetric contact conditions of the tori located on opposite sides. Each torus is initially tied to its neighbors at points coinciding with circle of radius $r_i + h_m + r_{it}$. The cavity fluid is defined isothermal through the time history, at a temperature $\theta = 288.15$ K.

The axisymmetric model was meshed with linear axisymmetric membrane elements⁴², and element sizes ranging linearly from $2h_m$ at the location where the tori are initially tied to $5h_m$ at the points coinciding with circle of radius r_i and circle of radius $r_i + 2h_m + 2r_{it}$ (i.e., farthest away from the point ties). This mesh discretization follows the results of a mesh convergence study, while favoring small elements at locations where

⁴² Axisymmetric membrane elements are line elements representing axially symmetrical membrane behavior along an axis of revolution and subjected to axially-symmetrical loading. These are internally described in cylindrical coordinates r, z and θ , with displacements u_r and u_z and stress components $\sigma_{ss}, \sigma_{\theta\theta}$ (i.e., hoop stress) and $\sigma_{s\theta}$; s represents the direction parallel to the element line. *Linear* axisymmetric membrane elements then refer to those of which are characterized with a linear shape, or interpolation, function.

contact is expected to occur. The mesh convergence study was conducted using a reduced-order model based on {1} an axisymmetric representation of the air cavity T_0 only and a contact line (i.e., the contact line ran from the origin to the vertex where air cavities T_0 and T_1 meet in Fig. V-12), {2} the same model definitions discussed in this paper, as appropriate, and {3} the discretization with linear axisymmetric membrane or shell elements; shells were only considered to validate that the bending stiffness had an insignificant effect on the solution. Considering uniform element sizes ranging from $100h_m$ as the courser discretization to $4h_m$ as the finer, the following outputs showed convergence with differences below 1%: strain energy, cavity pressure, vertical displacement of nodes located on the axis of symmetry, and maximum von Mises stress. This agreement was shown between adjacent data points as the model was discretized further with either membrane or shell elements, as well as when membrane and shell models of the same discretization were compared. Furthermore, convergence was particularly observed for discretization below or equal to $20h_m$, with agreement within 0.2% for each of the aforementioned outputs. Such results validated the use of membrane elements and informed the discretization utilized in the solution. Detailed results of the mesh convergence study are presented in section B.2 (Appendix B).

The internal pressure load rate followed a smooth load amplitude over a total time of 1 s. The amplitude, a , follows the function described in Eq. 65, defined by the initial time and amplitude (t_i, Γ_i) and final time and amplitude (t_{i+1}, Γ_{i+1}) , where $da/dt = d^2a/dt^2 = 0$ at t_i and t_{i+1} .

$$a = \Gamma_i + (\Gamma_{i+1} - \Gamma_i)(10 - 15\xi + 6\xi^2)\xi^3, \text{ given } \xi = \frac{t-t_i}{t_{i+1}-t_i} \text{ and for } t_i \leq t \leq t_{i+1} \quad (65)$$

The resulting pressure load vs time is shown in Fig. V-13 with a maximum of 101.325 kPa at 1 s, commensurate with standard sea level pressure [15]. A total time of one second is approximately six times the period of the lowest eigenvalue (i.e., 5.7 Hz). The latter was obtained using the Lanczos eigensolver [100] with the axisymmetric representation shown in Fig. V-12, without interactions and using axisymmetric shell elements. While shell elements were not used in the quasi-static solution presented, they provided an estimate of the first eigenvalue while avoiding the spurious (i.e., non-physical) modes present in the eigenvalue extraction using membrane elements [133]. Said eigenvalue is taken as a lower bound for estimating the timescale that promotes quasi-static behavior as the addition of fluid mass is expected to increase its value, reducing its period⁴³.

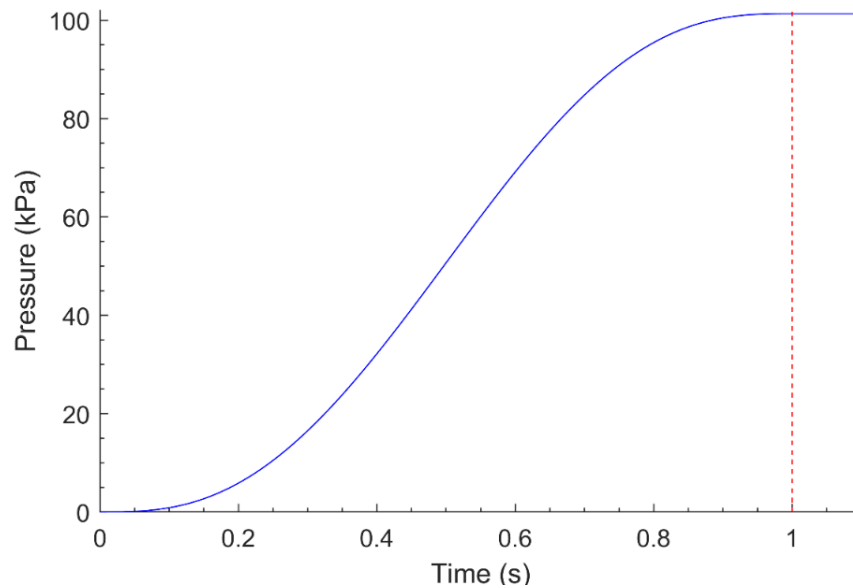


Fig. V-13. Smooth pressure load versus time.

⁴³ Comparisons of the eigen value and mode with analytical solutions are discussed in section B.3 (Appendix B).

5.5.3. Structural Response

Given the modeling components described, including geometric nonlinearity, the hydrostatic representation of the fluid-structure interactions, contact and the solver selection, a quasi-static response of the toroidal representation with $n = 41$ was solved up to and past collapse, based on the model definitions just described⁴⁴. As pressure P_{ia} is applied on the internal membrane surfaces (refer to Fig. V-12) following the smooth load amplitude (refer to Fig. V-13), the axisymmetric model of the toroidal representation displaces along the axisymmetric axis (i.e., y axis). This is represented in Fig. V-14 by the von Mises stress contours⁴⁵, where equilibrium states A-E are associated with the load magnitudes marked in the pressure vs time plot within said figure. From the initial state shown in A, states B and C show progressive displacement along the axisymmetric axis, with increased stresses along membrane regions that display significant arc radii changes. Arc radii changes, in turn, result in the development of contact interactions with contact areas proportional to the change in arc radius. Contact then affects the response in two ways: {1} it delays the instability of the structure by replacing the response of individual tori and sphere (i.e., volume change associated with membrane strain) to that of a coupled response, where increasingly larger contact areas dampers acceleration; and {2} increases the membrane stress along the contact areas (as observed by the development of high stress regions in Fig. V-14). To the latter, increases in membrane stress are primarily associated with radii, rather than with the contact pressure. As such, the stability of the structure is

⁴⁴ The Python code that created the Abaqus CAE file used to produce the results in question is included in section B.4 of Appendix B.

⁴⁵ The von Mises stress is obtained based on the Cauchy stress tensor, as stress states are reported based on this tensor regardless of the stress rates used to represent the constitutive relationship in the solver [58].

associated with a membrane modulus (i.e., elastic modulus times thickness) that prevents catastrophic volume changes as well as with the contact area resulting from the development of said volume changes.

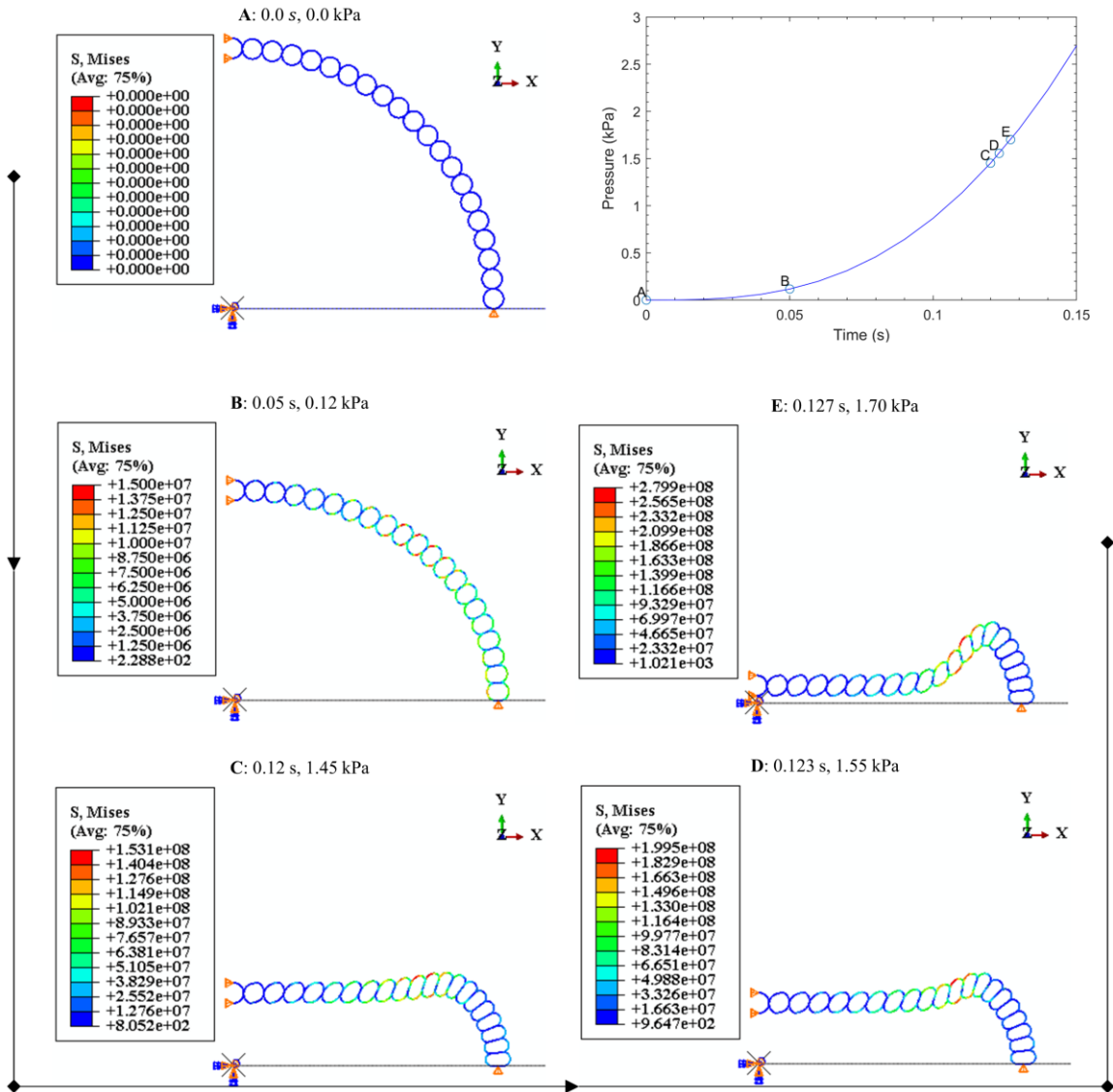


Fig. V-14. Von Mises stress contours at multiple loading magnitudes.

The impending collapse is prompted by a significant change in shape shown in state D, where tori with higher stresses (shown in red) have changed shape enough to allow further displacement along the axis of symmetry with such displacement occurring at a

faster pace after collapse; energy ramifications are discussed below. State E then shows progressive displacement, reaching the x-z plane soon after. Such collapse response is a result of the design axisymmetry. Following the tori $T_0 - T_{20}$ shown in Fig. V-12, the radial components of $P_{i\alpha}$ as applied to each T_i are increasingly taken tangentially by the structure as $T_i \rightarrow T_0$ through the contact interactions, and balanced at the x-z symmetry plane. This produces increasingly larger axisymmetric stresses and tori shape changes as $T_i \rightarrow T_{12}$, reducing their respective cavity volumes. These volume changes result in an accumulation of vertical displacement that leads to collapse, as well as membrane radii changes and proportional stress changes. In other words, the loading on tori nearest to the axis of symmetry push against those below in the direction of symmetry, rather than taking the load as axisymmetric stress.

Each tori and sphere have their own cavity enclosing an air volume with fluid state following Eq. 48, such that a reduction of cavity volume increases the fluid energy and ergo pressure, as guaranteed by the volume constraint in Eq. 52. Therefore, the structural collapse is partly a function of these fluid states, their impact on the fluid-structure interactions, and the resulting contact as displacement develops.

Pressure and Volume Plots as Collapse Indicators

Air pressure in each cavity is plotted as a function of the applied internal pressure in Fig. V-15, following labels $T_0 - T_{20}$ in Fig. V-12. Vertical dashed lines demark resulting pressure at equilibrium states C and D, per Fig. V-14. Cavity pressures shows progressive increase from $T_0 - T_{12}$ up to equilibrium state D, with notable grouping for $T_i > T_0$. This behavior is consistent with the design, where force equilibrium produces less axisymmetric

stresses as $T_i \rightarrow T_0$. Collapse is represented by the negative slopes present for T_{11} and T_{12} . While all slopes change at different load magnitudes, the slope signs change equally across T_i , except after state D, where T_{11} and T_{12} have negative slopes even though the rest turn positive.

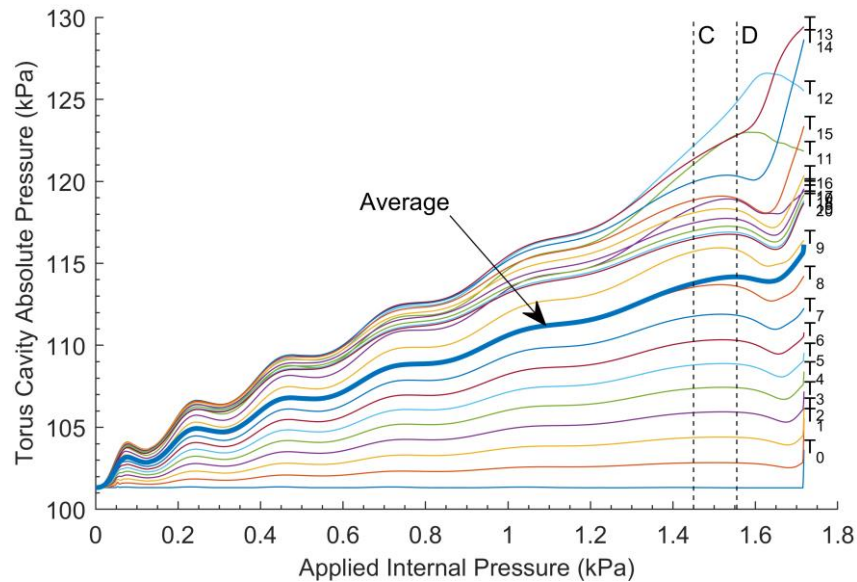


Fig. V-15. Cavities $T_0 - T_{20}$ absolute pressures versus applied pressure.

These pressure changes are a result of the minimum energy solution found when Eq. 52 is solved with displacements and cavity pressures, P_i , as variables at each increment, and applying the constraint $V_i - \bar{V}_i = 0$ for each cavity. Rather than keeping the volume constant, such constraint ensures that at each increment and for each cavity, \bar{V}_i (calculated from Eq. 50) is approximately equal to V_i ; the latter being a geometrical representation calculated from the interpolation functions of the volume elements within each cavity. \bar{V}_i and P_i are related by the equation of state (Eq. 48). In other words, Fig. V-15 in an indirect representation of the internal (or external work, if the KE is negligible) energy fraction

taken by each T_i , where greater cavity pressures correspond to T_i with greater energy fractions. Greater energy fractions are then a reflection of greater volume changes.

These volume changes are plotted in Fig. V-16 for $T_0 - T_{20}$ vs. applied pressure. The cavity pressure increases shown in Fig. V-15 translate to the volume reductions observed in Fig. V-16 as a direct consequence of the inversely proportional relationship between pressure and volume displayed in Eq. 48. Therefore, cavity volumes progressively reduce from $T_0 - T_{12}$ up to equilibrium state D, with T_{13} and T_{14} showing the most volume reduction of 15 - 22% past state D.

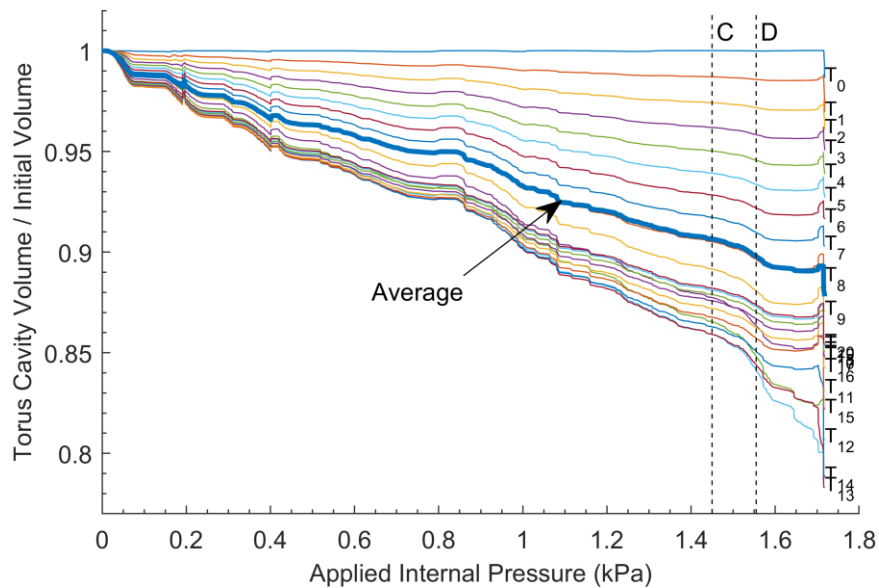


Fig. V-16. Cavities $T_0 - T_{20}$ volumes vs. applied pressure.

Tori membranes with the most volume changes are subjected to higher stresses as a result of the added cavity pressure and membrane arc radii increases. The von Mises stress contours shown in Figs. 17-19 demonstrate the stress effects associated with the pressure-volume relationship for equilibrium states B, C and D, respectively. These contours are a hemispheric visualization of the axisymmetric membrane results.

State B (Fig. V-17) shows minimum stresses (colored blue) along contact areas and on those membranes that did not have significant volume changes (i.e., T_0 and T_1). On the contrary, maximum stresses are observed along tori with the largest radii changes (i.e., T_{11} and T_{12}). The resulting cavity volume reductions drove significant displacement along the axis of symmetry, arriving at state C (Fig. V-18). The overall change of structural shape shown by state C changed the tori interaction and energy distribution, reflected in slope reductions in Fig. V-15 for $T_i > T_{12}$. The collapse shape then developed from state C into the post collapse region, shown in state D (Fig. V-19). Further applied pressure on the internals of the collapsed shape drove further displacement until opposite poles contacted each other along the x-z plane.

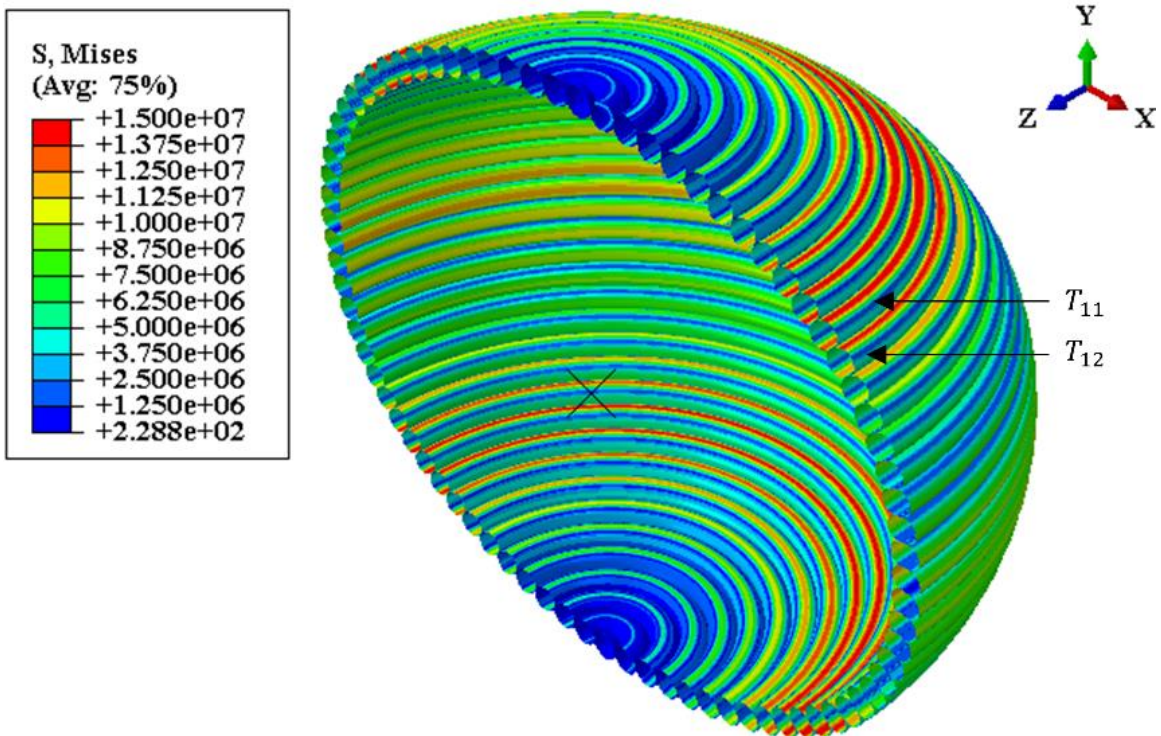


Fig. V-17. Hemisphere visualization of von Mises stress contour at equilibrium state B.

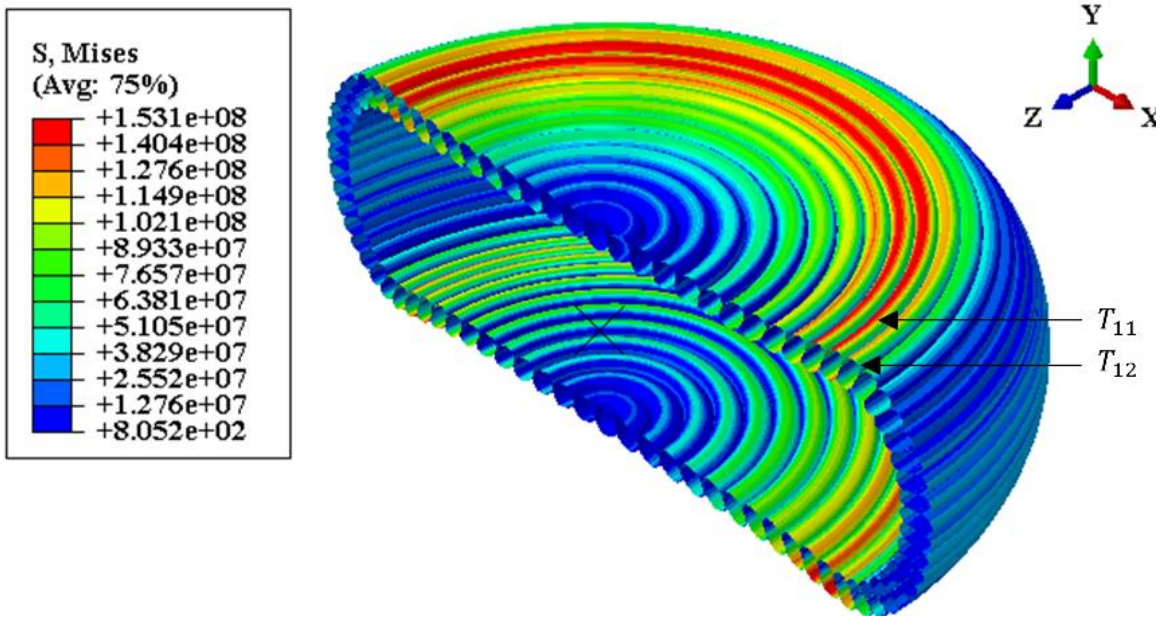


Fig. V-18. Hemisphere visualization of von Mises stress contour at equilibrium state C.

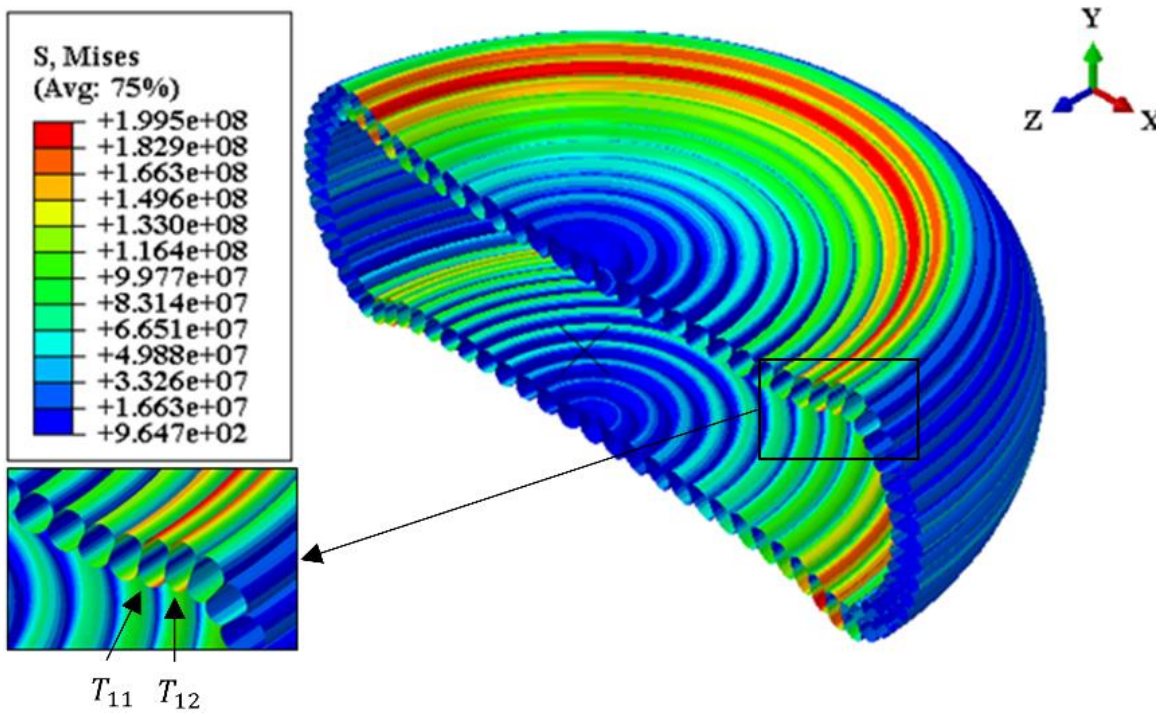


Fig. V-19. Hemisphere visualization of von Mises stress contour at equilibrium state D.

Energy Discussion

The energy balance was considered to evaluate whether the solution tended towards a quasi-static approximation. The energy balance at each equilibrium state is driven by the applied internal pressure work and balanced with the internal energy (IE) and KE; that is, balanced in the absence of other dissipative mechanics such as bulk viscosity, friction or numerical. Energies are part of the time history outputs, and the above mentioned three are plotted in Fig. V-20 through the load history for the results discussed. As desired for quasi-static responses, most of the work done propagated as IE throughout the simulation. The notable slope increases in the three energies from states D - E are another result of the collapse observed, which drove significant increases in IE and KE; the latter, albeit relatively small, is associated with the rapid movement after collapse. The IE increases reflect higher membrane strains (and stresses); a result of {1} the volume changes that increase the membranes radii and ergo their stresses (refer to Fig. V-14) and {2} increases in cavity pressures associated with those volume changes.

The KE to IE ratio plotted in Fig. V-21 shows that the KE did not exceed 5% of the IE during most of the load history, except near state B and after state D. A maximum KE/IE of 22 is shown at the beginning of load history. As stated in the modeling definitions, adjacent tori were tied to one another at their closest points (refer to Fig. V-12). Such points only share displacement DOF as membrane elements do not possess, by definition, rotational degrees of freedom. Consequently, rotation about these points is only restricted by contact initiated after deformation has occurred. Therefore, KE at the beginning of load history reflects the movement of the structure while the surface-to-surface contacts

develop. Nonetheless, this movement occurs at low energy levels, below 1 J, as shown in Fig. V-20. After state B, KE/IE reaches a maximum of 0.18 at 0.16 kPa, with tertiary changes in slope near 0.36 kPa, 0.65 kPa and 0.95 kPa. Each one of these changes follows a change in slope of the cavity pressure curves in Fig. V-15, indicating a notable shape change and therefore volume change (refer to Fig. V-16).

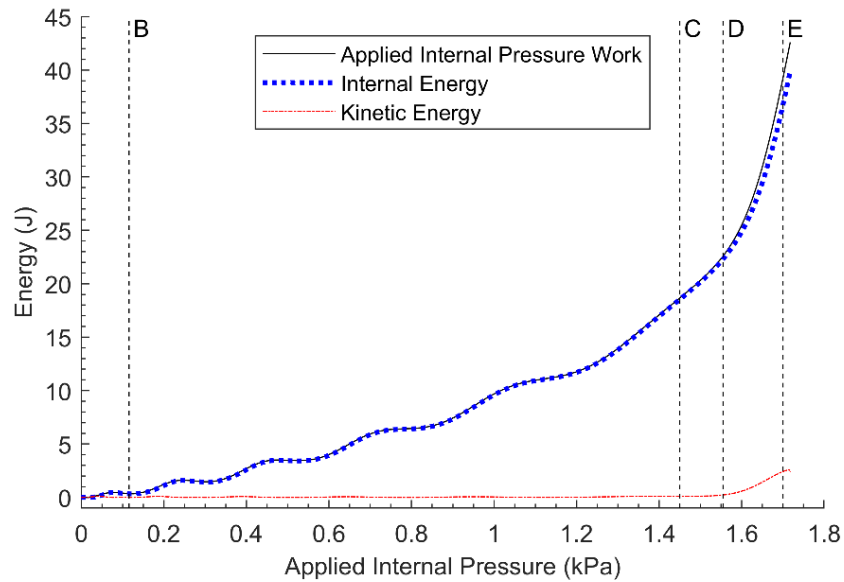


Fig. V-20. Applied work, internal energy and kinetic energy versus applied pressure.

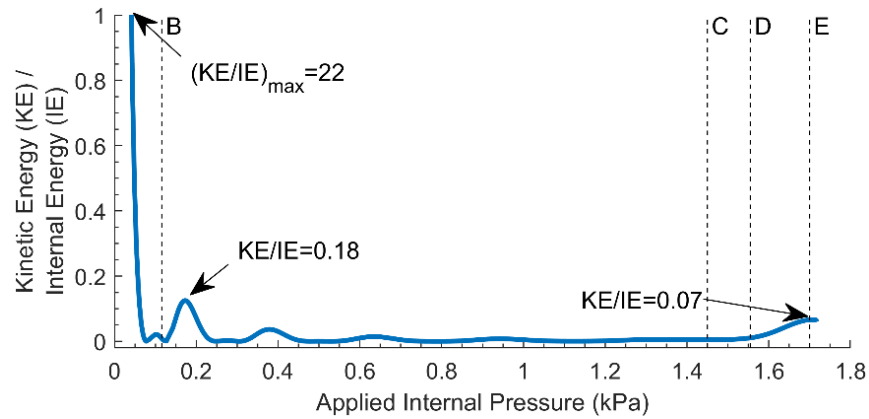


Fig. V-21. Kinetic to internal energies versus applied pressure.

It is relevant to note that, generally, a quasi-static solution disregards the formal dynamic problem for the search of a time-independent approximation that offers an

asymptotical limit to the former. In the work presented, acceleration is used as a dissipative term through the selection of the Euler operator and its inherent algorithmic damping, rather than dismissing the term as “static” solvers do. Consequently, the solution presented is one of dynamic nature, which approaches the time-independent approximation desired, where KE tends to zero. A formal definition of the quasi-static problem is proposed in Ref. [134].

Weight-to-Buoyancy Estimates Revisit

It is worth revisiting the W/B ratio considering the deformed structural shape. Let us estimate the W/B ratio of state C (V-14). The maximum displacement at state C is given by $u_z|_{state\ C} = 0.2291\ m$ (i.e., a 69.6% reduction in the external radius), resulting from $P_{ia}|_{state\ C} = 0.12\ kPa$. The weight, W , of the structure follows the numerator of Eq. 64 (page 64), where the internal density is given by $\rho_{ia}|_{state\ C} = \frac{(P_{ia}|_{state\ C})}{R\theta}$. The buoyancy, B , follows the denominator and is coarsely estimated as the initial buoyancy minus the volume of the spherical caps colored in gray on Fig. V-22, where the deformed shape shown is congruent with state C described in Fig. V-14. Given by Eq. 66, each of the spherical caps volume represents half of the estimated air volume no longer displaced by the structure, where $R \cong (r_i + 2h_m + 2r_{it})$. Given the above, the W/B ratio at state C is estimated as $\frac{W}{B}|_{u_z} \cong 3.7$. From the standpoint of buoyancy potential, this result suggests that the instability associated with its axisymmetry combined with the assigned material stiffness makes the Helical Sphere unfeasible as structurally defined.

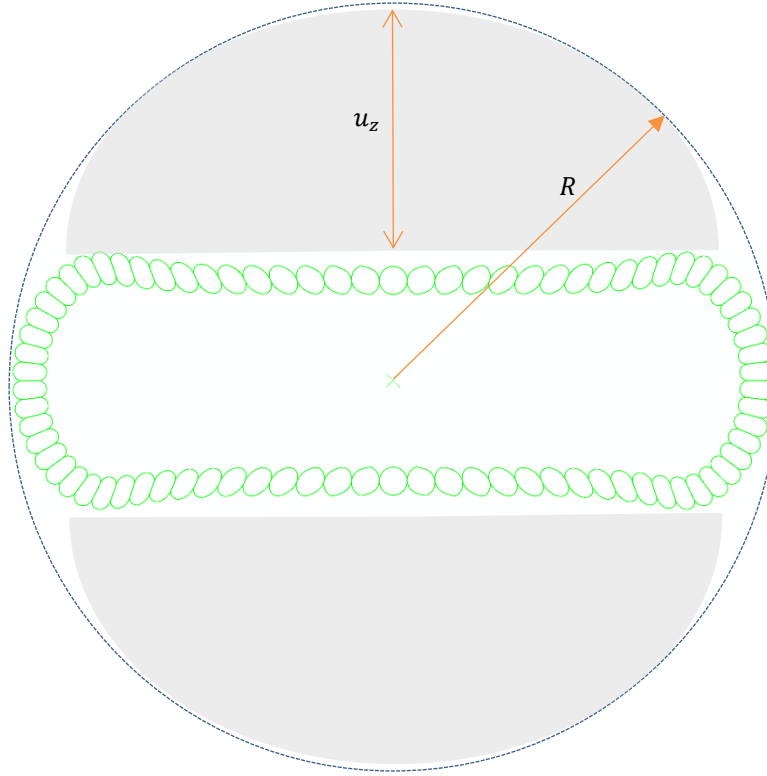


Fig. V-22. Deformed shape at state C for weight-to-buoyancy estimate.

$$V_{cap} = \frac{\pi}{6} u_z (3R^2 + u_z^2) \quad (66)$$

$$\frac{W}{B} \Big|_{state\ c} = \frac{V_{ta} \rho_{ta} + V_m \rho_m + \left[\frac{4}{3} \pi (r_i + h_m + r_{it})^3 - \frac{1}{2} V_{ta} \right] \rho_{ia} |_{state\ c}}{\left\{ \left[\frac{4}{3} \pi (r_i + h_m + r_{it})^3 + \frac{1}{2} V_{ta} \right] - \frac{\pi}{3} u_z (3R^2 + u_z^2) \right\} \rho_{oa}} \quad (67)$$

5.6. Summary and Conclusions

Firstly, the Helical Sphere was described, parametrizing its geometry in terms of the number of revolutions and membrane radius, among others. Secondly, the preliminary analysis demonstrated the weight and stress control potential of the design. Thirdly, the structural simulation framework was detailed, highlighting the hydrostatic representation of the fluid-structure interaction as the method to simulate the air-membrane interaction key to characterizing air-stiffened designs. This framework is then used to characterize the

structural response through the axisymmetric geometrical representation of the Helical Sphere. The structural response showed considerable deformation at a loading below a 99% remainder of a vacuum, decreasing the weight of air displaced and effectively negating a $W/B \leq 1$ (e.g., an actual $W/B = 3.7$ was estimated for state C). Nonetheless, the response exemplified important aspects of air-stiffened designs that warrant their further study as VLTAS:

1. The structural characterization of air-stiffened designs necessitates the inclusion of formulations that capture the fluid-structure and contact interactions developed when inward pressure loading is applied. These interactions are illustrated in Figs. V-14 to V-19, enabled by:
 - a. Large displacements occurring as a result of insignificant bending stiffness typically found in membrane films.
 - b. Fluid states within membrane cavities.
2. The inclusion of geometric nonlinearity is a necessary condition to correctly model membrane behavior and air-membrane interactions.
3. The energy balance provides KE/IE ratios throughout the load history, which support the quasi-static solution validation.
4. The ideal gas law is a representative state equation for the air within cavities shaping the air-stiffened design across the pressures and temperatures that a buoyant VLTA structure is expected to encounter.

While the need for contact interactions and its tangential definitions is design dependent, the general characterization of fluid-stiffened structures that contain multiple

fluid barriers generally requires a fluid-structure interaction formulation that {1} provides accurate fluid states within the pressures and temperatures observed and {2} represents the kinematics associated with the solution timescale. With this in mind, the framework introduced in this chapter provides a long-time structural solution of an air-stiffened membrane envelope subjected to uniform pressure, based on a quasi-static approach and a fluid-structure interaction approximated with a hydrostatic fluid formulation that allows constitutive states to be estimated as a function of the fluid states within a Lagrangian description. The use of such interaction is adequate when the kinematics of fluid flow are negligible.

VI. ICORON AIR-STIFFENED DESIGN

The design trajectory, summarized in Fig. I-4, culminates with the development of the *Icoron* design as another iteration of the generalized representation of stiffened designs (Fig. I-1). Devised around the vertex locations of an icosahedral polyhedron, the *Icoron* aims at overcoming the instability characteristics anticipated in the Helical Sphere (Chapter V) by extending symmetry to an icosahedral nature while maintaining stress control characteristics. Specifically, the *Icoron* is based on membrane-to-membrane connectivity that follows the icosahedron and its subdivisions, which approximate a sphere via a tessellation of triangles, leading to a highly symmetric connectivity arrangement on which to shape the air-stiffened design.

To characterize the *Icoron* as a potential VLTAS, this chapter starts with a geometrical description of the *Icoron*, where unit cell membrane and air volumes are developed in terms of the concept of unit cells from which the *Icoron* is built. The structural description follows, where the finite element assembly, the unit cell loading, and the force balance are detailed. This is followed by the development of the W/B estimates and a discussion on the effects that geometry parameters have on the predicted W/B ratio and stress levels. This discussion lays the foundation for a case study, where the development of the solution throughout the loading history is studied along with the energy distribution associated with the solution. The weight-to-buoyancy of the deformed structure is also evaluated throughout the load history. The chapter closes with discussions on modulus and preload (air) pressures dependencies, and displacement field equivalence associated with modulus-thickness products (i.e., Eh_m). Fig. VI-1 traces the aforementioned development

and provides a guide to follow as the chapter is read in an attempt to ease understanding; chapter section numbers are found in parenthesis.

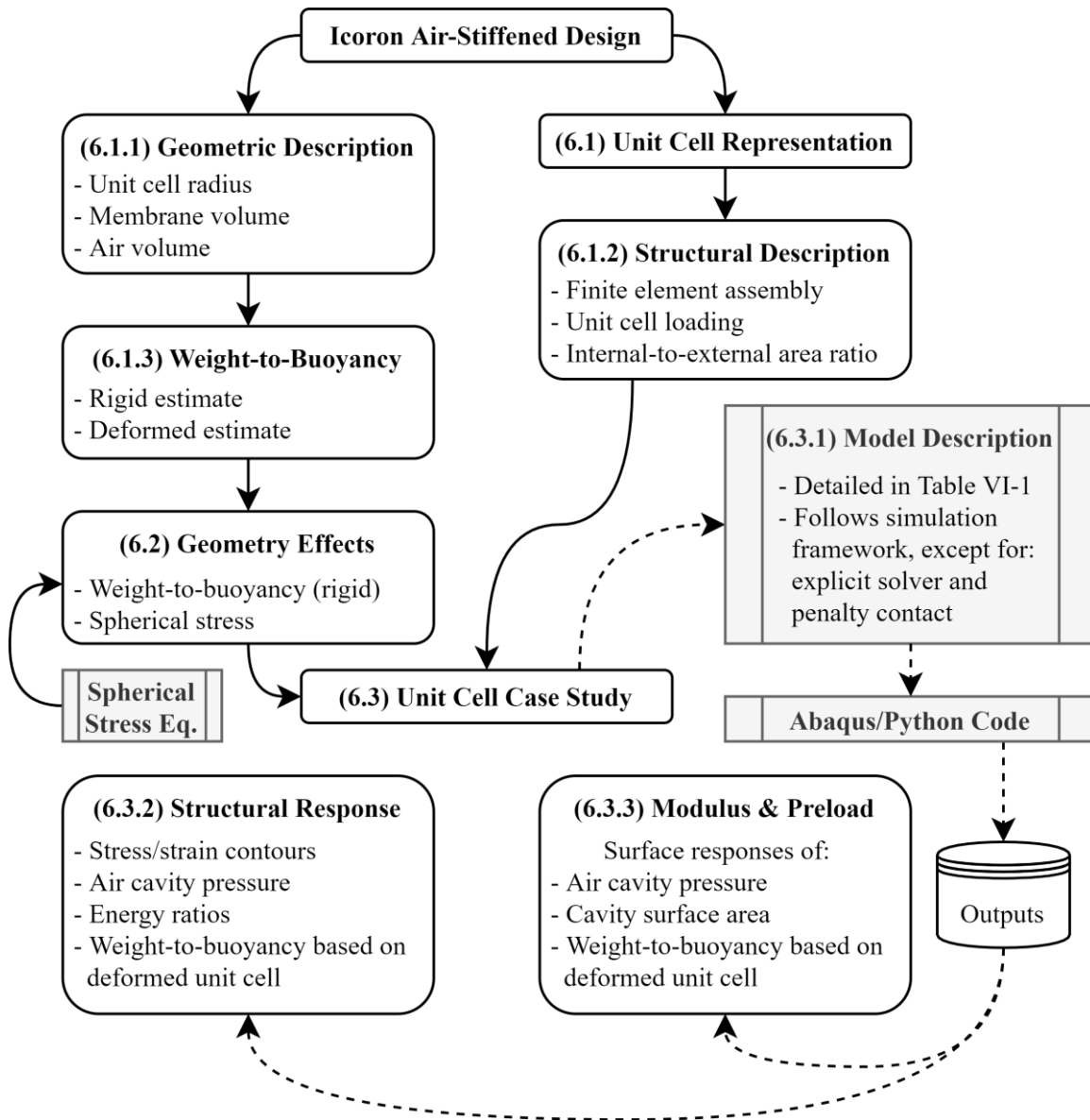


Fig. VI-1. Diagram tracing the development of Icoron and its structural solutions as presented.

6.1. Design Description

The Icoron geometry can be visualized by starting with the icosahedron and its subdivisions, shown in Fig. VI-2. Given any structural radius⁴⁶, the subdivisions allow for smaller distances among vertices forming a triangle, while preserving the rotational symmetry characteristic of the icosahedron [2,5]. The location of vertices, rather than the triangular faces, are of particular interest here as the goal is to extend symmetry beyond the axisymmetry inherent in the Helical Sphere design.

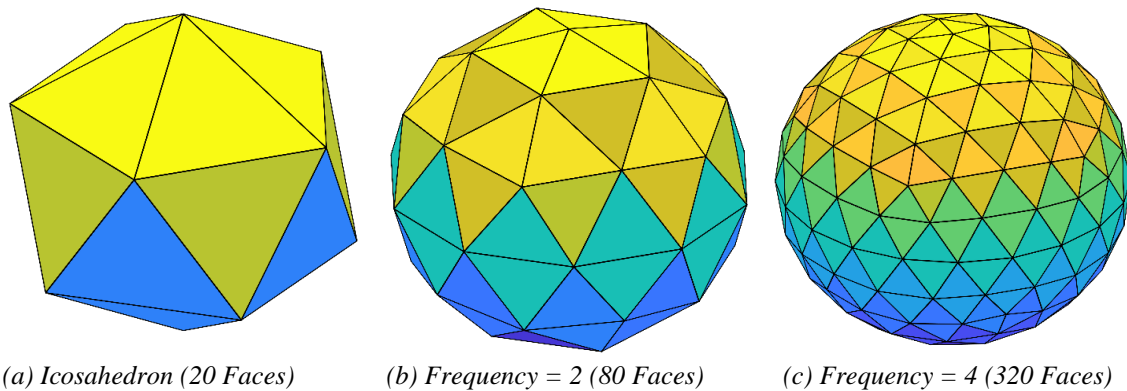


Fig. VI-2. Icosahedron and its subdivisions.

The air-stiffened idea is incorporated by replacing each triangular flat surface with what is noted here as an air-stiffened *unit cell* (UC), where the mid-plane is positioned parallel to the triangular surface removed, and its side surfaces follow the planes formed by adjacent vertices (forming each triangle) and the center of the structure. This development is depicted in Fig. VI-3 for subdivision frequency of four⁴⁷. The geometry of the UCs therefore relies on the positioning of the vertices as a function of subdivision frequency, which includes the UC radius, surface areas and internal volume.

⁴⁶ Radius of the sphere circumscribed by the polyhedron (i.e., distance between the center and any vertex).

⁴⁷ The subdivision frequency, f , is defined as the number of times each triangle of an icosahedron (Fig. V-14) is divided such the number of vertices is given by $10f^2 + 2$ and the number of triangular faces (or UCs) is given by $20f^2$ [158,159].

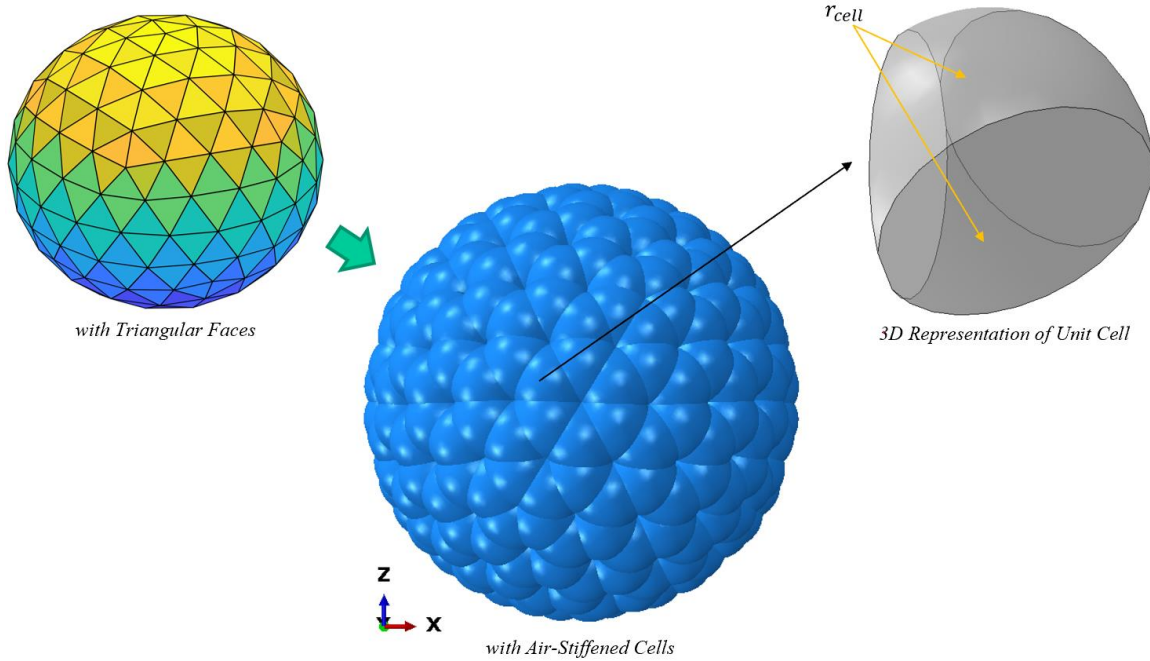


Fig. VI-3. Icoron geometry development for frequency 4 (320 faces or unit cells).

6.1.1. Geometric Description

With the above in mind, the following geometrical relationships are used to develop each UC along with its radius, areas and volumes, indexed $()_k$. Consider the vertices' locations⁴⁸ of any one triangle forming an icosahedron or any of its subdivisions (Fig. VI-2). A triangle centroid, \mathbf{CP} , is the coordinate mean among vertices, V_{ij} , such that:

$$(\mathbf{CP}_j)_k = \frac{1}{3} \sum_{i=1}^3 V_{ij} \quad (68)$$

The UC radius, r_c , is then selected as the maximum Euclidean distance between each vertex and \mathbf{CP} , such that:

$$(r_c)_k = \max[|\mathbf{V}_1 - \mathbf{CP}|, |\mathbf{V}_2 - \mathbf{CP}|, |\mathbf{V}_3 - \mathbf{CP}|] \quad (69)$$

⁴⁸ Vertices' locations were obtained using the Geodesate function within Wolfram Mathematica 11.3, which subdivides the Icosahedron, among other polyhedrons, and projects new vertices onto the circumscribed sphere [160].

Following the 2D representation in Fig. VI-4, a sphere of radius r_c is positioned at such centroid. Three *cutting planes* are then formed, each by any two adjacent vertices and the structure's center, intersecting the sphere and isolating the UC from three spherical caps. The UC is located such that the vertices forming its midplane are coincident with structural radius r_s .

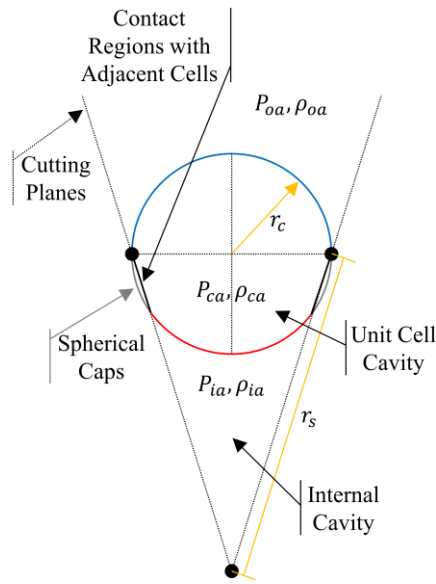


Fig. VI-4. Two-dimensional representation of unit cell boundary and loading conditions

Volumes and areas are formulated by subtracting said spherical caps from the initial sphere of radius r_c . Therefore, the surface area occupied by the three spherical caps, A_{cap} , is obtained with Eq. 70 as a summation of each cap height, l , indexed ()_q [135].

$$(A_{cap})_k = 2\pi(r_c)_k \sum_{q=1}^3 l_q \quad (70)$$

In other words, the three cutting planes effectively remove portions of the sphere (of radius r_c) to form the UC. Each of these portions is a spherical cap, where their areas and volumes are characterized in terms of r_c and l . As such, A_{cap} represents the total surface area removed from the sphere used to form the UC. The volume occupied by these

three spherical caps, V_{cap} , is similarly obtained with Eq. 71 [135] as a summation of the volume occupied by each UC.

$$(V_{cap})_k = \frac{\pi}{3} \sum_{q=1}^3 l_q^2 [3(r_c)_k - l_q] \quad (71)$$

The UC consists of air enclosed by membrane material. The membrane volume, V_m , is approximated by subtracting Eq. 70 from the surface area of the sphere (of radius r_c), adding the cap base areas (i.e., circles coincident with the cutting planes), and multiplying by the membrane thickness, h_m ; shown in Eq. 72. Similarly, the volume of the air enclosed, V_c , is approximated by subtracting Eq. 71 from the volume of the sphere; shown in Eq. 73.

$$(V_m)_k = h_m \left[4\pi(r_c^2)_k - (A_{cap})_k + \sum_{q=1}^3 \pi(r_{cap}^2)_q \right] \quad (72)$$

$$(V_c)_k = \frac{4}{3}\pi(r_c^3)_k - (V_{cap})_k \quad (73)$$

This development generates geometric quantities for one (of k) UC. In fact, the number of UCs is equal to the number of triangular faces they replace (i.e., the number of UCs is equal to $20f^2$). This is observed in Fig. VI-5, where both the number of UCs and their radii (Eq. 69 above), normalized by the structural radius, are plotted as a function of subdivision frequency. The dependence of r_c with frequency drives the consideration of this design as small radii correspond to smaller membrane stresses, allowing for a stress control-driven design.

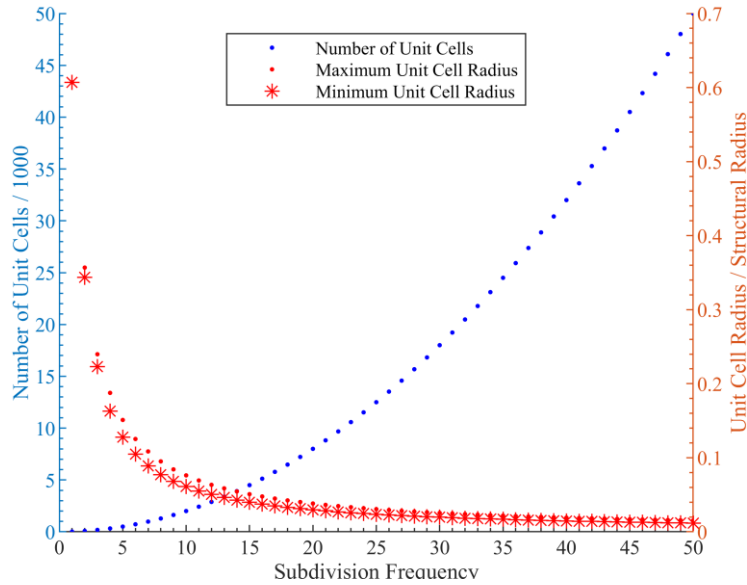


Fig. VI-5. Number of unit cells (blue) and unit cell radii (red) as a function of subdivision frequency.

Fig. VI-5 also exposes two aspects that can challenge the manufacturing feasibility of the structure. Firstly, the relationship between the number of UCs and their radii may limit the subdivision frequency that can be manufactured as UC radii can fall below tolerances. This is compounded by the increasingly high number of UCs required to form the spherical volume (i.e., the left vertical axes of Fig. VI-5 and Fig. VI-6). And secondly, small variations in UC radii exist for any subdivision of the icosahedron (i.e., subdivision frequencies above one). This variance is amplified with frequency, where, for example, a frequency of 15 has 30 unique UC radii and a frequency of 50 has 441; portrayed in Fig. VI-6 (in red)⁴⁹ along with the number of UCs (in blue)⁵⁰. Despite these limitations, the

⁴⁹ This was obtained by calculating the UC radii at each subdivision frequency, where uniqueness was defined as any radius within 1E-6 tolerance of those adjacent in value. The inverted peaks in Fig. VI-6 are associated with frequency multiples of three.

⁵⁰ Section 8.5 of Popko [158] suggests that asymmetry is a result of the subdivision schema, discussing one that allows for rotational symmetry to be maintained. Specifically, by imposing the intersection of three great circles (i.e., circles along circumscribed sphere formed by Icosahedron vertices) based on equal-arc spacing to generate new subdivisions. To the author's knowledge, Wolfram Mathematica does not detail the subdivision schema implemented in Geodesate (used here to produce vertex locations). Preserving rotational

Icoron offers a significant set of parameters, or design variables, from which to study a design space in the search for a set that suggests potential as a VLTAS.

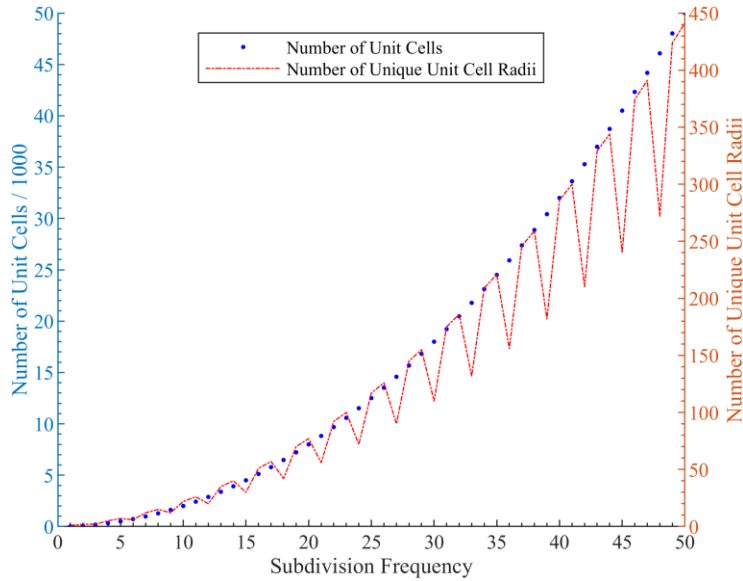


Fig. VI-6. Number of unit cells (blue) and number of unique unit cell radii (red) as a function of subdivision frequency.

6.1.2. Structural Description

Despite the small variations in radii, the similarities among UCs present a computationally efficient method of estimating the structural behavior of the Icoron by considering a single UC. Following the 2D representation in Fig. VI-4 above, the outer membrane is exposed to external air of density ρ_{oa} and pressure P_{oa} . The UC cavity is defined as air-filled with pressure P_{ca} at outer temperature T_{oa} . The internal cavity is defined with density ρ_{ia} and pressure P_{ia} . The external membrane surface (colored blue in Fig. VI-4) and internal membrane surface (colored red in Fig. VI-4) are subjected to $\Delta P_{co} = P_{ca} - P_{oa}$ and $\Delta P_{ci} = P_{ca} - P_{ia}$. These densities and pressures define the initial state and

symmetry becomes an increasingly relevant topic as specific subdivision frequencies are studied, as such symmetry leads to equally-sized UCs, eliminating any instability potentially associated with dissimilar UCs.

the structural response along the loading history, as well as the resulting W/B ratio. Adjacent UCs are represented via rigid contact planes along the cutting planes used to develop the UCs. These contact planes act as symmetry boundary conditions along the radial direction. This description is represented three-dimensionally by the assembly presented in Fig. VI-7, where the assembly (Fig. VI-7a) is formed by collocating the contact cone (Fig. VI-7b) and the UC (Fig. VI-7c) along the three vertices used to define the UC. Further details regarding the FEM definitions are discussed in section 6.3.1.

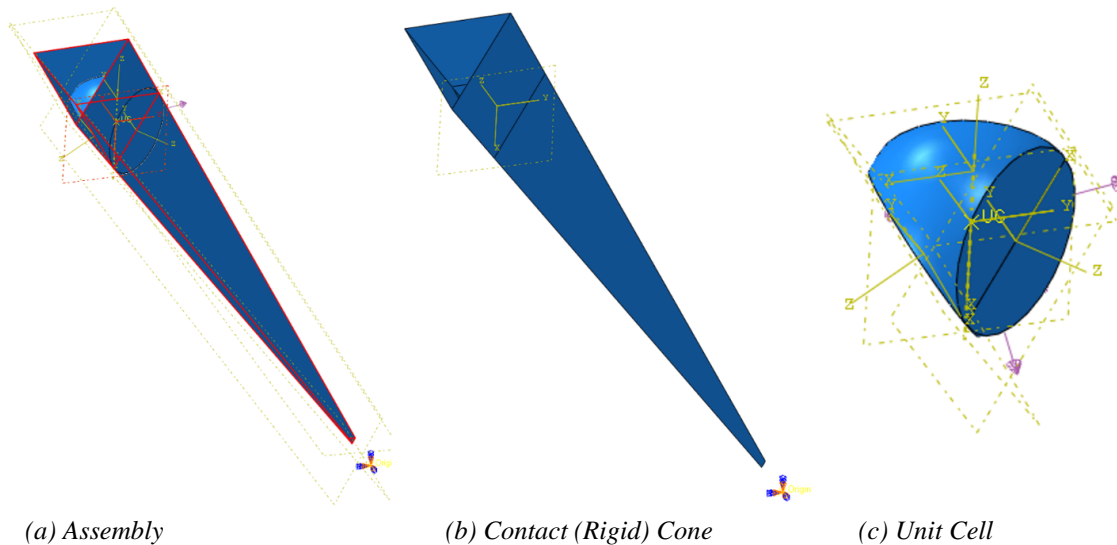


Fig. VI-7. Finite element model unit cell assembly.

Structural responses are driven by the initial state and loading conditions structures are subjected to. In the interest of evaluating the Icoron’s response to a low-pressure⁵¹ air environment within its internal cavity, the loading history is defined as follows: the UC is initially static and stress-free with equal air pressures ($P_{ia} = P_{ca} = P_{oa}$). The loading

⁵¹ “Low-pressure” is used here to differentiate a (perfect) vacuum, which is an unattainable condition, from a condition which approaches it. Nonetheless, structural responses discussed in this chapter used a loading history up to what would be considered a perfect vacuum given that the vacuum level is not strictly defined. The American Vacuum Society considers 25 torr the point between low and medium vacuum [155], which is achievable by some single pumps currently on the market [136,156].

history then follows one or two steps⁵². A one-step loading history applies a pressure load onto its internal membrane from 0 to P_{ia} in opposite direction (i.e., towards the structural center⁵³). This loading represents air removal from the internal cavity of the Icoron. A two-step loading history initially increases P_{ca} from P_{ia} up to a target cavity pressure and subsequently subjects the internal membrane to the applied load, going from 0 to P_{ia} .

The structural response of each UC is associated with the force resultant of these pressures (i.e., the work done by the applied loading is associated with the direction and magnitude of the force resultant). It is therefore beneficial to establish the force resultant in relation to P_{ia} , P_{ca} and P_{oa} , particularly along the radial direction as it is the UC's primary direction of movement. Following Fig. VI-4, Eq. 74 becomes a statement of the radial resultant, $R^{(r)}$, as a function of P_{ia} , P_{ca} and

P_{oa} and the radial component of the external and internal loading areas, $A_{oa}^{(r)}$ and $A_{ia}^{(r)}$, respectively⁵⁴.

These surface areas are diagrammed in Fig. VI-8; also viewed as the cross-sectional areas of the contact cone (Fig. VI-7b), tangent to the membrane surfaces above and below the UC.

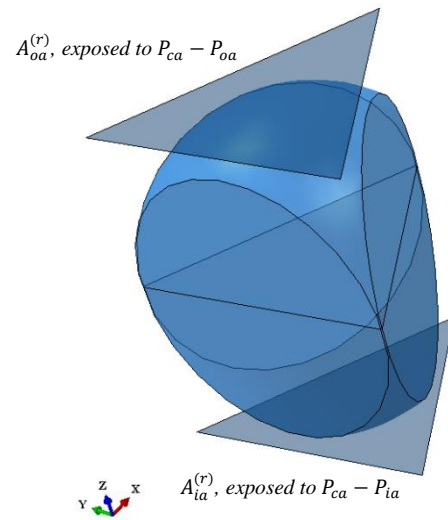


Fig. VI-8. Diagram of surface areas associated to radial component of pressure loading.

⁵² Abaqus separates the problem history based on steps defined by the user, such that loads, BCs and other state variables are defined at each step or propagated from previous steps. For example, a static analysis can be defined as the first step and a dynamic analysis as the second step, where the BCs are propagated (kept in place) throughout the entire problem history, but new loading conditions are defined for the second step.

⁵³ The structural center is the center of the Icoron structure. For the purpose of the UC assembly (Fig. VI-7), this center is [0,0,0].

⁵⁴ $A_{oa}^{(r)}$ and $A_{ia}^{(r)}$ are given by the area of a triangle formed by three vertices, \mathbf{U}_i . Each \mathbf{U}_i is located $r_c + h_m$ away from \mathbf{V}_i (Eq. 69) in either radial direction. In other words, each vertex is located at $r_s \pm (r_c + h_m)$ in spherical coordinates. The area of the triangle is given by $1/2\|(\mathbf{U}_2 - \mathbf{U}_1) \times (\mathbf{U}_3 - \mathbf{U}_1)\|$.

$$R^{(r)} = P_{ca}A_{oa}^{(r)} - P_{oa}A_{oa}^{(r)} - P_{ca}A_{ia}^{(r)} + P_{ia}A_{ia}^{(r)} \quad (74)$$

In other words, Eq. 74 is the summation of each force component (i.e., pressure times area) in the radial direction given P_{ia} , P_{ca} and P_{oa} on area components $A_{oa}^{(r)}$ and $A_{ia}^{(r)}$, where positive forces are defined in the radial direction away from the structural center. An $R^{(r)} = 0$ leads to a force balance⁵⁵ expressed in terms of area ratio (Eq. 75). The latter serves to determine a subdivision frequency, and therefore the number of UCs, required to achieve a force balance at a state with given P_{ia} , P_{ca} and P_{oa} .

$$\frac{A_{ia}^{(r)}}{A_{oa}^{(r)}} = \frac{\left(1 - \frac{P_{oa}}{P_{ca}}\right)}{\left(1 - \frac{P_{ia}}{P_{ca}}\right)} \quad (75)$$

At the stress-free initial state, the $A_{ia}^{(r)}$ to $A_{oa}^{(r)}$ ratio is solely a function of the initial geometry. Following Fig. VI-9, this ratio reflects the exponential relationship between the subdivision frequency and UC radius, where the former asymptotes at $A_{ia}^{(r)}/A_{oa}^{(r)} = 1$. In other words, an increased subdivision frequency drives the vertices (shaping the UC) closer to each other, which in turn produces a smaller UC radius. The cutting planes (formed by the center and these vertices) then produce a UC with increasingly smaller radial distance. Such radial distance drives $A_{ia}^{(r)}/A_{oa}^{(r)}$, which is independent of r_s .

Eq. 75 can also be rearranged to express P_{ca}/P_{oa} as a function of $A_{ia}^{(r)}/A_{oa}^{(r)}$ illustrated in Eq. 76. The P_{ca}/P_{oa} ratio is particularly valuable as it provides the air pressure required inside the cavity to drive towards a static balance. In other words, for any given subdivision frequency, there is set of P_{ia} , P_{ca} and P_{oa} that can allow the UC to end

⁵⁵ The “force balance” statement used here is not to be confused with an equilibrium statement; the latter would imply the inclusion of the internal forces resulting from the applied loading.

approximately at its initial position after loading given that the radial resultant is defined to be zero. This condition has the potential to minimize changes in the estimated buoyancy resulting from any displacement.

$$\frac{P_{ca}}{P_{oa}} = \frac{A_{oa}^{(r)} - \left(\frac{P_{ia}}{P_{oa}}\right) A_{ia}^{(r)}}{A_{oa}^{(r)} - A_{ia}^{(r)}} \Rightarrow \frac{P_{ca}}{P_{oa}} \Big|_{P_{ia} \rightarrow 0} \rightarrow \frac{1}{1 - \left(\frac{A_{ia}^{(r)}}{A_{oa}^{(r)}}\right)} \quad (76)$$

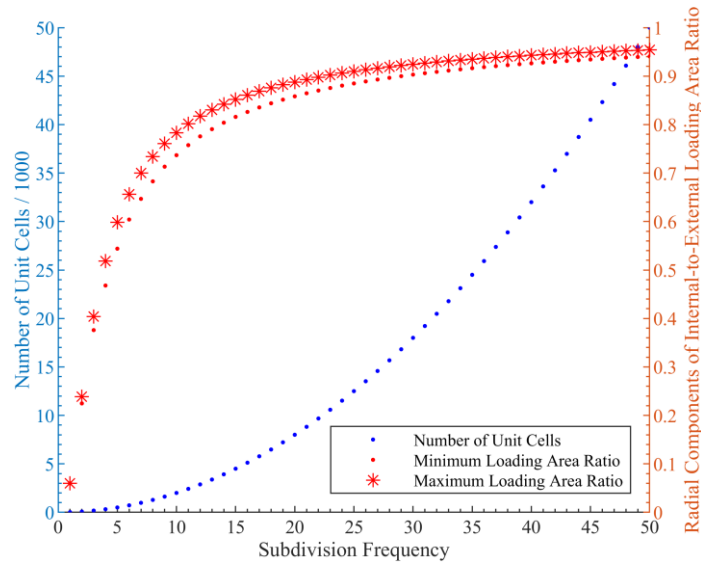


Fig. VI-9. Number of unit cells (blue) and radial components of internal-to-external loading area ratio (red).

This *balanced* condition is challenged by the fact that the loading history (i.e., UC pressurization and/or applied loading akin to air removal from the Icoron internal cavity) results in membrane strains that drive displacement within the confines of the contact cone, or more generally, the cutting planes (Figs. VI-4 and VI-7b). Nonetheless, it sets a ceiling on the maximum UC cavity air pressure required to minimize the radial displacement about the initial state after loading. The pressure ratio is plotted (in red) as a function of frequency in Fig. VI-10. Note that the exponential relationship of the area ratio (Fig. VI-9) with frequency linearizes the resulting pressure ratio (Eq. 76 and Fig. VI-10). Notwithstanding,

the proportionality between the pressure ratio and the subdivision frequency significantly increases the material stress limit required to meet this *balanced* condition. Furthermore, both Figs. 9 and 10 reflect the minimums and maximums over a range of values associated with the small variations in UC radii shown in Fig. VI-6. These structural considerations associated with the Icoron play a key role on the selection of parameters that show potential to produce W/B ratios below one.

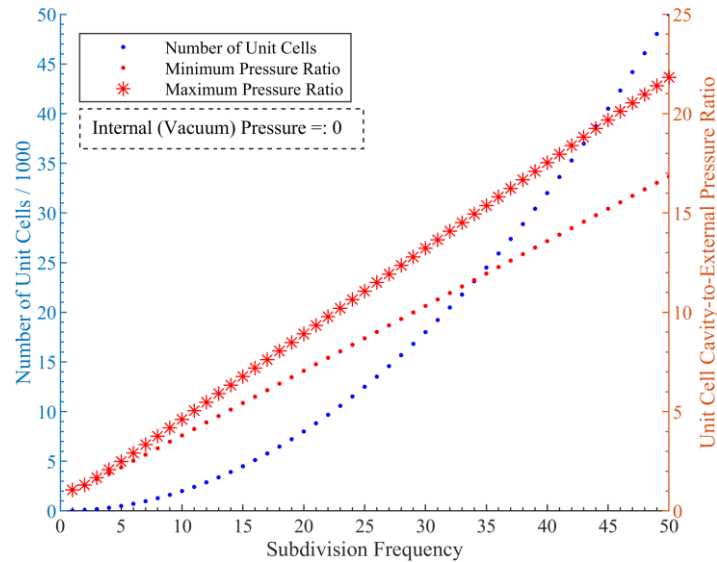


Fig. VI-10. Number of unit cells (blue) and unit cell cavity-to-external pressure ratio (red).

6.1.3. Weight-to-Buoyancy Ratio Development

As shown in previous chapters, estimating the W/B ratio in advance of a structural analysis allows the selection of dimensionality and material properties that can lead to a feasible structure. Albeit unrepresentative of an elastic structure, such a priori estimation starts with the assumption of rigidity (i.e., the atmosphere displaced by the structure prior to loading remains constant during and after loading) given that deformed states are unknown. Following the geometrical description presented in section 6.1.1, the W/B ratio is approximated by Eq. 77, given Eqs. 72 and 73, the membrane density, ρ_m , and the air

densities (contained within the UCs, ρ_{ca} , internal to the structure, ρ_{ia} , and external or of the atmosphere, ρ_{oa}).

$$\frac{W}{B} = \frac{\rho_m \sum_{k=1}^n (V_m)_k + \rho_{ca} \sum_{k=1}^n (V_c)_k + \rho_{ia} \left[\frac{4}{3} \pi r_s^3 - 0.5 \sum_{k=1}^n (V_c)_k \right]}{\rho_{oa} \left[\frac{4}{3} \pi r_s^3 + 0.5 \sum_{k=1}^n (V_c)_k \right]} \quad (77)$$

The first term in the numerator sums the membrane mass for all the UCs. The second term sums the mass of air contained within said UCs. The third term approximates the mass of the air internal to the structure (i.e., air volume to be evacuated leaving a low-pressure region) by subtracting $0.5V_c$ from the spherical volume of radius r_c . Similarly, the buoyancy mass in the denominator is obtained by adding $0.5V_c$ to the spherical volume of radius r_c . Selecting $0.5V_c$ approximates the volume bounded by a sphere of radius r_c and either side of the UC. The validity of this approximation is a function of the structure's radius, r_s , and the number of UCs, or subdivisions, considered. The mass of the air internal to the structure and the buoyancy are underestimated given such ratio, regardless of the number of subdivisions. One can argue that underestimating the buoyancy affects the W/B ratio significantly more than underestimating the mass of air internal to the structure given the relative magnitude of their densities. Such argument is not complete, or as important, until the rigidity assumption is removed.

To address the latter, the W/B after deformation can be estimated by redefining the denominator of Eq. 77 to the volume of the sphere of radius $r_s + r_c + h_m - u_r$, where u_r is the radial displacement incurred by the representative UC at loading $P_{ia}|_{deformed}$. Furthermore, $\rho_{ia}|_{deformed} = \frac{(P_{ia}|_{deformed})}{R\theta}$, where R and θ are the ideal gas constant and the atmospheric temperature.

$$\left. \frac{W}{B} \right|_{deformed} = \frac{\rho_m \sum_{k=1}^n (V_m)_k + \rho_{ca} \sum_{k=1}^n (V_c)_k + \rho_{ia} \left[\frac{4}{3} \pi r_s^3 - 0.5 \sum_{k=1}^n (V_c)_k \right]}{\rho_{oa} \left[\frac{4}{3} \pi (r_s + r_c + h_m - u_r)^3 \right]} \quad (78)$$

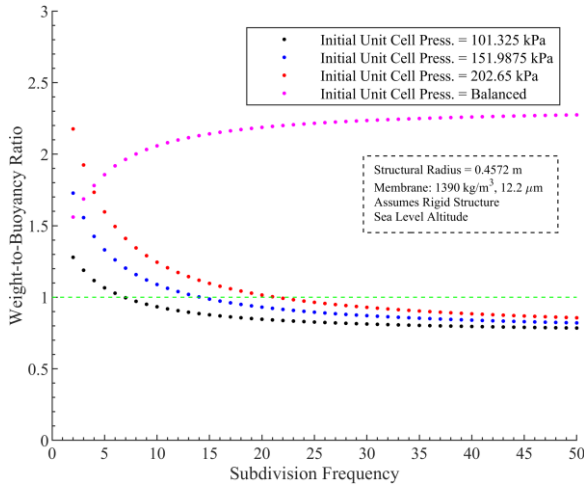
6.2. Geometry Effects on Expected Weight-to-Buoyancy Ratios and Stress

As in previous chapters, the selection of structural radius and target UC pressure (i.e., r_s and P_{ca}) is based on the W/B ratio (Eq. 77), as well as the stress estimate; the latter being the spherical stress, σ_s , as shown in (Eq. 79) rather than the hoop stress⁵⁶.

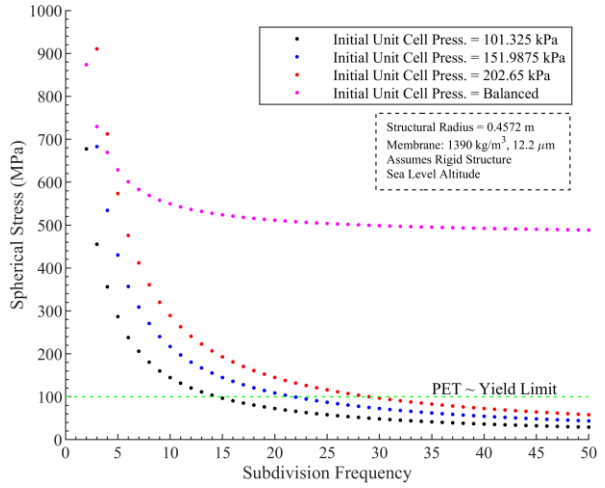
$$\sigma_s = \Delta P \left(\frac{r_c}{2h_m} \right) \quad (79)$$

W/B estimates were obtained across subdivision frequencies $2 \leq f \leq 50$ considering membrane thickness $h_m = 12.2 \mu\text{m}$, membrane density $\rho_m = 1390 \text{ kg/m}^3$, $r_s = 0.4572 \text{ m}, 0.6096 \text{ m}, 1.2192 \text{ m}$ and $P_{ca} = [1, 1.5, 2]101.325 \text{ kPa}$. Plotted in Figs. VI-11a, VI-11c and VI-11e, the nature of the W/B curves is associated with the quadratic relationship between the membrane material volume forming each cell and its radius, and the cubic relationship between the air volume contained within each cell and its radius. Such cubic relationship dictates the weight contribution associated with the UC air pressure, which sets a limit on its maximum air pressure. Not surprisingly, the W/B decreases with P_{ca} and increases with r_s ; that is given fixed h_m and ρ_m . The effect of P_{ca} on W/B lessens with f since the UC volume decreases, reducing the overall mass contribution of the added air. Notably, the curves presented in VI-11b and VI-11c are essentially vertical shifts of VI-11a driven by increased r_s .

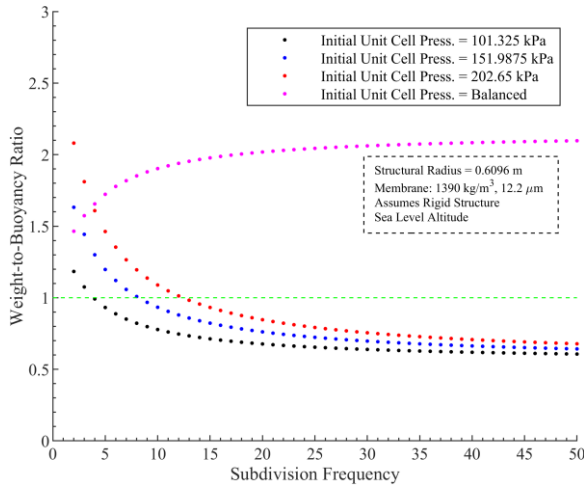
⁵⁶ The spherical stress estimate is a result of thin-walled spherical shell theory, assuming small strains and isotropy. The loading and inherent spherical symmetry lead to equal stress in any tangential direction. In general terms, a force equilibrium results in $\sigma_s [2\pi h(r_i + h/2)] = \Delta P (\pi r_i^2)$, where r_i is the internal radius, h is the thickness, σ_s is the spherical stress and ΔP is the differential pressure. This leads to $\sigma_s = \Delta P r_i^2 / [2h(r_i + h/2)] \cong \Delta P [r_i / (2h)]$ since $r_i + h/2 \cong r_i$ [161].



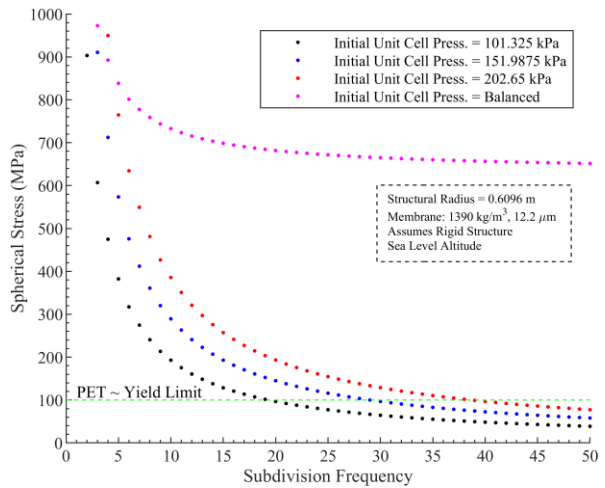
(a) Weight-to-buoyancy for $r_s = 0.4572$ m



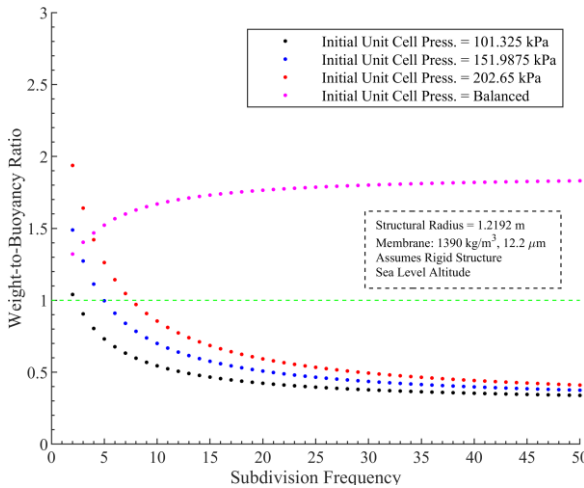
(b) Spherical stress for $r_s = 0.4572$ m



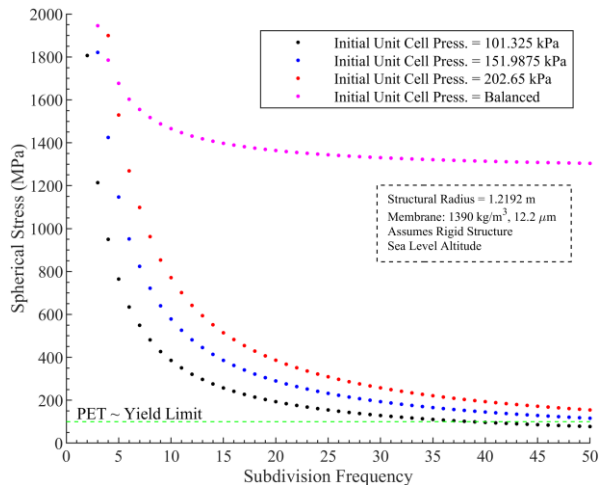
(c) Weight-to-buoyancy for $r_s = 0.6096$ m



(d) Spherical stress for $r_s = 0.6096$ m



(e) Weight-to-buoyancy for $r_s = 1.2192$ m



(f) Spherical stress for $r_s = 1.2192$ m

Fig. VI-11. Weight-to-buoyancy (rigid) and spherical stress estimates.

In other words, increasing the subdivision frequency decreases UC sizes, which in turn reduces the membrane volume required to form the UCs and the air volume required to fill them. Furthermore, balanced UC pressures (i.e., follows Eq. 76) result in W/B ratios above one given the parameters considered. Interestingly, balanced UC pressures invert the W/B ratio curve because loading area ratios increase with f (as shown in Fig. VI-9). Given the membrane parameters, $r_s \ll 0.4572 \text{ m}$ do not lead to W/B (rigid) estimates below one. Figs. VI-11b, VI-11d and VI-11f show the spherical stresses over the same r_s , P_{ca} and f . The UC size effect is observed by the reduction of the spherical stress estimate with increasing frequency. The stress reduction is not sufficient for the balanced UC pressures (those given by Eq. 76) to be below the yield limit for Mylar (approximately 100 MPa) given the h_m chosen [136].

The changes in volume and mass as a function of f are presented in Figs. VI-12 and VI-13, respectively, all shown in red with respect to the volume or mass of air displaced by the Icoron; that is when rigidity, air density at sea level, and the same membrane thickness and density as above are assumed. From a volume ratio perspective, the cumulative volume of all of the UC (air) cavities exponentially decreases with f from approximately one to 0.07 (at $f = 50$), which supports the reduction in W/B ratio as a function of f shown in Figs. VI-11a, VI-11c and VI-11e. On the other hand, the membrane volume shows to be an insignificant fraction of the displaced air when compared to the UC cavities volume; that is because it is approximately $2.3\text{E-}4$ at $f = 1$ and $6.3\text{E-}4$ at $f = 50$.

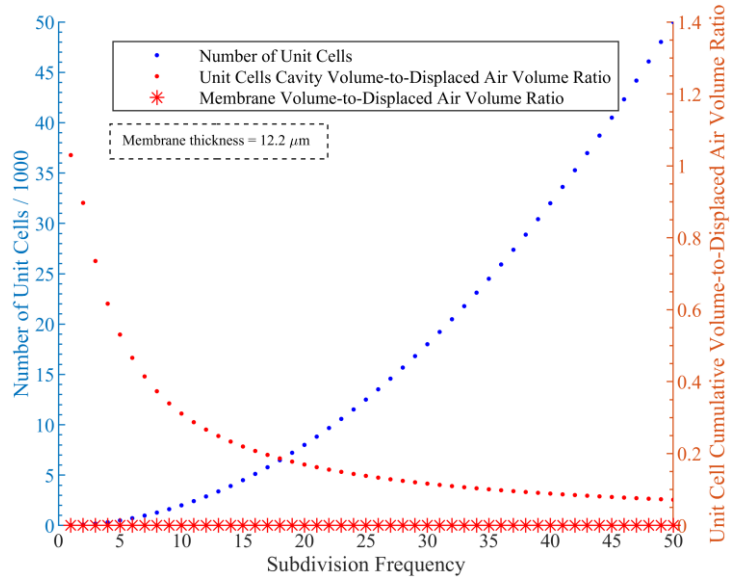


Fig. VI-12. Unit cells air cavity and membrane total volumes as ratios of the displaced air mass across subdivision frequency.

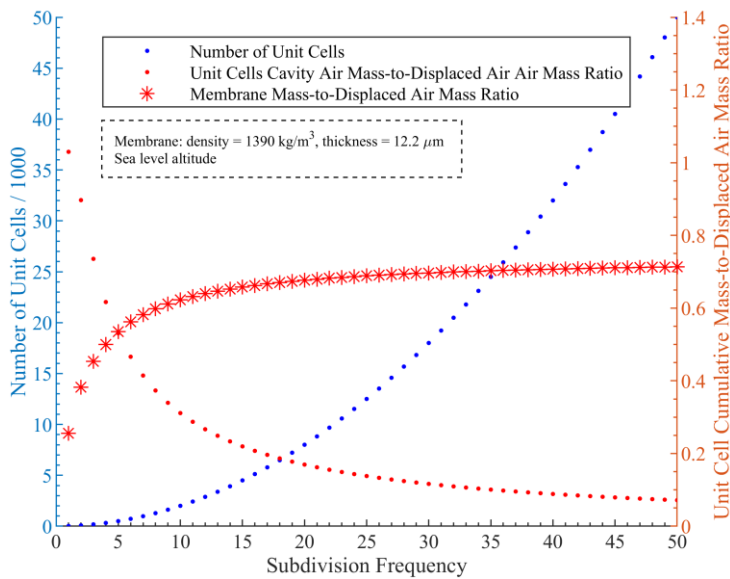


Fig. VI-13. Unit cells air cavity and membrane total masses as ratios of the displaced air mass across subdivision frequency.

The membrane contribution to the overall weight of the Icoron is best captured by considering the mass ratios shown in Fig. VI-13. Given that the membrane density is over 1,000 times the density of air, any scale disparities dissipate when considering mass ratios.

The membrane mass (and volume) increases with f , primarily driven by the added surfaces (i.e., contact surfaces on the UC) required to connect the increased number of UCs. The mass ratio changes from 0.26 to 0.6 within $1 \leq f < 9$, and from 0.61 to 0.71 within $10 \leq f < 50$. These results suggest that membrane thickness plays a critical role as f increases, particularly for subdivision frequencies above where the UC air cavity mass ratio and the membrane mass ratio are approximately the same; this happens at $f = 5$ for the mass properties used to develop Fig. VI-13.

An important characteristic of the Icoron structure is the competing objectives: for any given material, a reduction of W/B ratio promotes a higher structural radius selection while such selection constraints the minimum subdivision frequency required for a target stress level, driven by the UC size. This is observed across Figs. VI-11b, VI-11d and VI-11f, where, for example, a $f = 20$ is required to achieve 100 MPa with $r_s = 0.6096 m$, compared to $f = 15$ for $r_s = 0.4572 m$.

6.3. Case Study

Just as with the Helical Sphere, a case study provides significant structural characteristics regarding the expected response of the Icoron geometry when subjected to the loading histories described (in section 6.1.2). Following the discussion of W/B ratio and spherical stress above, a radius $r_s = 0.4572 m$ is selected⁵⁷ for the purpose of estimating the structural behavior of an Icoron structure, where Fig. VI-11b suggests a W/B ratio below one for $f \geq 8$, and a spherical stress below 100 MPa for $f \geq 15$. Membrane

⁵⁷ Given that r_s is defined from the center of the structure to any of the vertices used to shape the Icoron, the overall radius of the structure is $r_s + r_c + h_m$. In this case, $r_s + r_c \cong 0.48 m$.

material properties resemble PET membrane at room temperature (i.e., modulus of 4.36 GPa, Poisson ratio of 0.38 and density of 1390 kg/m^3) [48]. A frequency of 15, representing 4,500 UCs, is selected in the interest of minimizing the number of UCs while estimating stress levels below the yield limit of 100 MPa identified for PET membrane [136]. Such structural radius and membrane material combination indicate potential buoyancy, while minimizing the size of the spherical volume the UCs are required to fill. The latter is a conservative consideration given the potential challenges inherent to the manufacturing of considerably large Icoron structures with increased numbers of relatively small UCs.

6.3.1. Model Description

The UC, representative of a $f = 15$ Icoron of $r_s = 0.4572 \text{ m}$, is modeled following the structural description presented in section 6.1.2, starting from the assembly presented in Fig. VI-7 above. A summary of the FEM definitions is illustrated in Table IV-2⁵⁸. The structural mesh is made up of 30,778 three-node membrane elements with linear shape functions and side lengths ranging from $0.01r_c$ to $0.05r_c$. Following Fig. VI-14, this mesh arrangement favors smaller elements along the sides where membrane surfaces meet.

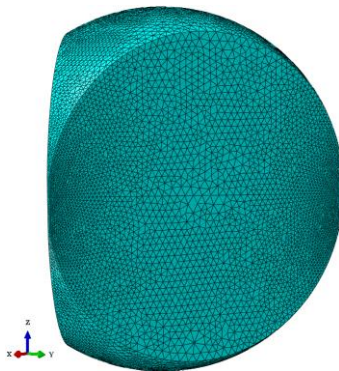


Fig. VI-14. Unit cell membrane mesh.

⁵⁸ The Python code that created the Abaqus CAE file used to produce the results discussed here is included in section C.2 of Appendix C.

Table VI-1. Unit cell finite element model definitions.

Definition	Value(s)	Comments
<i>Initial Geometries</i>	<ul style="list-style-type: none"> - Structural radius = 0.4572 m - Frequency = 15 - Unit cell radius = 0.232 mm - Membrane thickness = 12.2 μm - Contact cone representing symmetry boundary conditions 	Unit cell as representative of Icoron geometry.
<i>Material</i>	<ul style="list-style-type: none"> - Density = 1390 kg/m^3 - Linear isotropic: <ul style="list-style-type: none"> o Elastic modulus = 4.36 GPa o Poisson's ratio = 0.38 	Membrane is based on DuPont Mylar® HP 12.6 μm PET film [116] properties.
<i>Mesh</i>	<ul style="list-style-type: none"> - Membrane: 30,778 three-node membrane elements with a linear shape function and side lengths ranging from 0.01 to 0.05 the unit cell radius - Air cavity: three-noded surface facet with linear shape function (development discussed on section 5.4.2) - Contact cone: 50,909 three-node rigid elements 	Membrane element code is M3D3. Rigid element code is R3D3. Surface facet element code is F3D3.
<i>Initial Conditions</i>	<ul style="list-style-type: none"> - Unit cell cavity pressure, P_{ca}, = 101.325 kPa - Reference or atmospheric pressure, P_{oa}, = 101.325 kPa 	
<i>Boundary Conditions</i>	<ul style="list-style-type: none"> - Contact cone affixed to a reference node at (0,0,0) 	
<i>Loading</i>	<ul style="list-style-type: none"> - A total pressure of 101.325 kPa - Applied over 0.05s following a smooth load amplitude (example shown in Fig. V-13) 	
<i>Contact</i>	<ul style="list-style-type: none"> - Normal contact applied through general self-contact with penalty enforcement - Tangential contact defined as frictionless with finite sliding 	Penalty stiffness was controlled by Abaqus, defaulted as 10x the membrane modulus.
<i>Solver</i>	<ul style="list-style-type: none"> - Abaqus/Explicit - Geometric nonlinearity allowed 	

Loading is based on a one-step loading history, applying a pressure load onto its internal membrane from 0 to 101.325 kPa (complete vacuum) towards the structural center over a 0.05 s timescale through a smooth load amplitude (example shown in Fig. V-13). The initial state is therefore a stress-free static state with $P_{ca} = P_{oa} = 101.325$ kPa. The mesh density and solution timescale were selected after considering their effects on energy quantities⁵⁹. The UC response is modeled using the simulation framework detailed in

⁵⁹ Energy distribution across the time history and timescale variations were studied to judge the validity of the mesh and timescale considering the search of a quasi-static solution. Undesirable energy quantities (i.e., kinetic, contact enforcement and bulk viscosity) combined remained below 10% across 93% of the load history. Furthermore, internal energy to work done ratios remained above 90% at and above the mesh density

section 5.4 [120], which includes the hydrostatic representation of the air-membrane interaction, but replaces the implicit solver with an explicit solver and the use of Lagrangian multipliers for contact enforcement with a penalty stiffness enforcement method. Such changes allow the solution to resolve the convergence difficulties associated with membrane wrinkling and contact.

Lagrangian multipliers, a contact constraints enforcement method in Abaqus/Implicit (discussed and used in the solutions presented in Chapter V), is replaced with penalty contact in the general contact algorithm within Abaqus/Explicit. The enforcement of normal contact through the penalty method approximates hard contact conditions by using springs to resist penetration of the slave surface into the master surface. This is depicted in Fig. VI-15, where the upper surface is the slave surface and the surface being penetrated (in the diagram) is the master surface. The penalty stiffness, k_s , effectively prevents penetration of the slave node into the master surface.

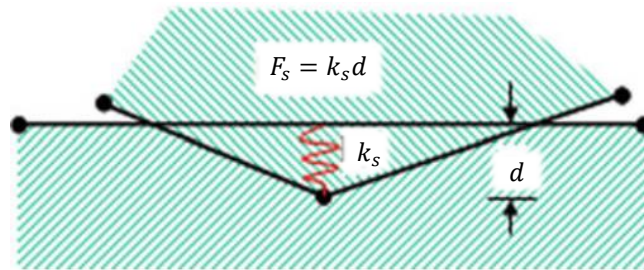


Fig. VI-15. Penalty contact enforcement diagram [61].

Numerically, the penalty method first looks for active or violated contact conditions based on a central difference prediction at each new increment, correcting overclosures. Restoring forces F_s are then applied to preserve the condition such that k_s becomes

chosen (i.e., 30,778 three-node membrane elements with a linear shape function and side lengths range from 0.01 to 0.05 of the UC radius). These effects are discussed further in Appendix C.

proportional to the penetration, d . The associated potential energy $\frac{1}{2}k_s d^2$ for each correction is incorporated in the virtual work of the structure, enforcing contact constraints and becoming part of the energy balance. Compared to the Lagrange multiplier method, the penalty method trades precise enforcement for improved convergence rates and shorter computation times as it does not add DOF to the system. While Abaqus lets users choose linear and nonlinear penalty stiffnesses, the default of ten times the element stiffness was used as the initial stiffness for the analyses presented in this chapter without notable penetration between the UC and the contact (rigid) cone [59,61].

6.3.2. Structural Response

Two components of the UC work in conjunction to develop the resulting work done by the applied loading: the air defined as an ideal gas and the membrane defined by a linear elastic constitutive model; both are coupled by a volume constraint that reflects changes in air pressure as pressure loading onto the membrane surface. Such work is confined to the path established by internal surface of the rigid cone through hard frictionless contact interactions, intended to approximate the Icoron's behavior through a symmetric response. All that said, Fig. VI-16 contains the von Mises and logarithmic maximum principal strain contours at five loading states, from 0.00116 to one of the applied load ratio (i.e., 0.117 kPa to 101.325 kPa), completing the defined loading history⁶⁰. Each state is labeled along with their associated time, applied load ratio and radial displacement (relative to the initial structural radius [$r_s + r_c + h_m = 0.4804$ m]). Keep in mind that the behavior is constrained by the rigid cone, albeit not displayed in Fig. VI-16.

⁶⁰ Results shown are based on outputs extracted at 1000 evenly-spaced time points (i.e., every 5E-5 s).

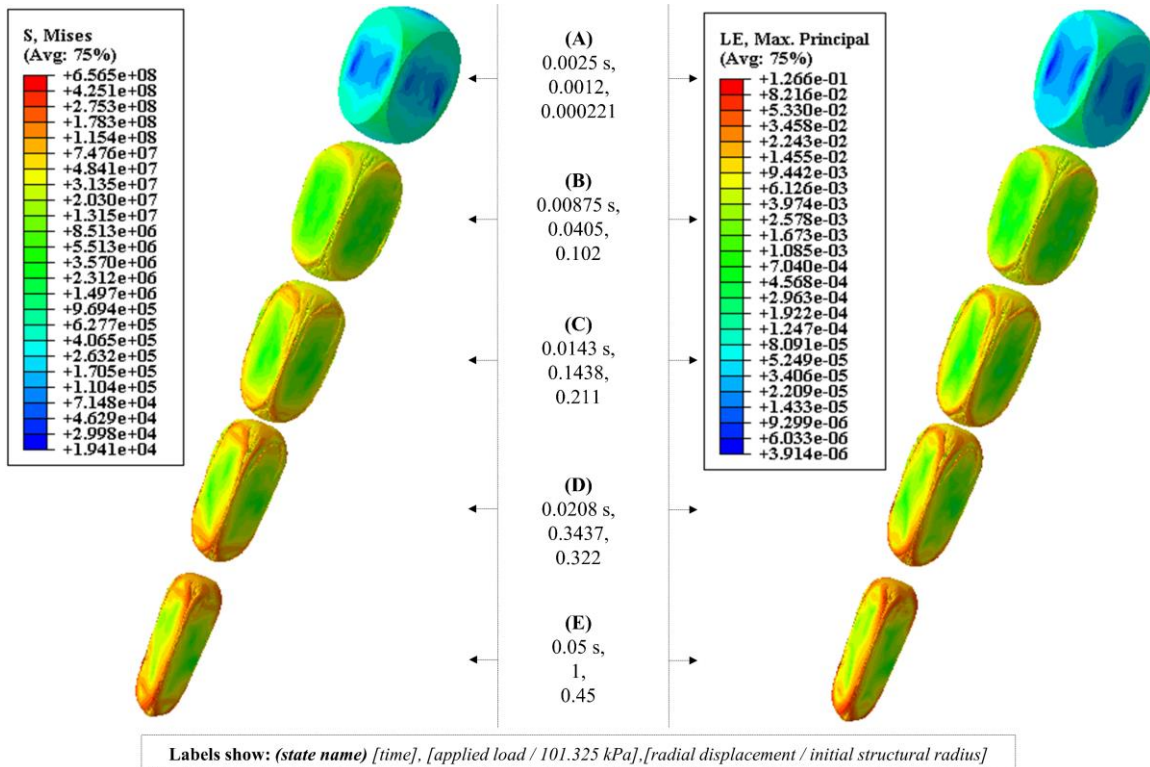


Fig. VI-16. Von Mises stress contours (left) and logarithmic maximum principal strain contours (right) overlaid for five loading states.

State A shows the UC undergoing small displacement and strains, which can be considered geometrically linear behavior. In other words, the UC at state A is geometrically similar to the initial state. From there, the UC transitions to geometrically nonlinear states dominated by wrinkling⁶¹, where high stresses and large strains are observed⁶². State B shows that wrinkling is initiated along the edges connecting the contact surfaces, developing further to all connecting edges through state E. This causes the UC to radially elongate as it is crumpled along the contact cone, inducing wrinkling on the internal (i.e.,

⁶¹ Wrinkling is characterized as a local buckling phenomenon occurring as membrane regions display out-of-plane deformation [91,162–164].

⁶² The development of contact and wrinkling during the transition from state A to B drove solution nonconvergence when the implicit solver (used in Chapter 5) was considered, ergo adjusting the methodology proposed for air-stiffened design to the use both implicit and explicit solvers, as applicable.

loading) and external (i.e., exposed to atmospheric pressure) surfaces, reducing their exposed surface areas. At maximum applied load (state E), the UC incurred a radial displacement of 0.45 the initial structural radius (i.e., $r_s + r_c + h_m$), with maximum stresses in the order of 650 MPa. It is beneficial to state, nonetheless, that stresses outside wrinkled regions remained near or below the near 100 MPa estimate provided by the spherical stress estimate (Eq. 79).

Wrinkling is primarily characterized by the progressive folding of the UC corners onto themselves past state A as the UC is being driven radially downward by the applied load and constrained to the path established by the rigid cone. As observed in Fig. VI-17 from states A-E, a UC corner folds inward and onto itself, stretching the membrane to accommodate the progressive development of the folds, subsequently causing significant changes in the stress field in and around them. Higher stresses are primarily localized to regions adjacent to those where membrane folding has occurred, which indicates that {1} those regions are being stretched to allow the corner to fold further inward in order to ‘fit’ within the volume of the rigid cone and {2} the membrane radius is locally increased and tends to be proportional to the membrane stress (Eq. 79). The development of these folds is strictly dependent on preserving the contact interactions among adjacent regions of the same surface (i.e., self-contact) and between the UC and the rigid cone, such that surface crossing is prevented.

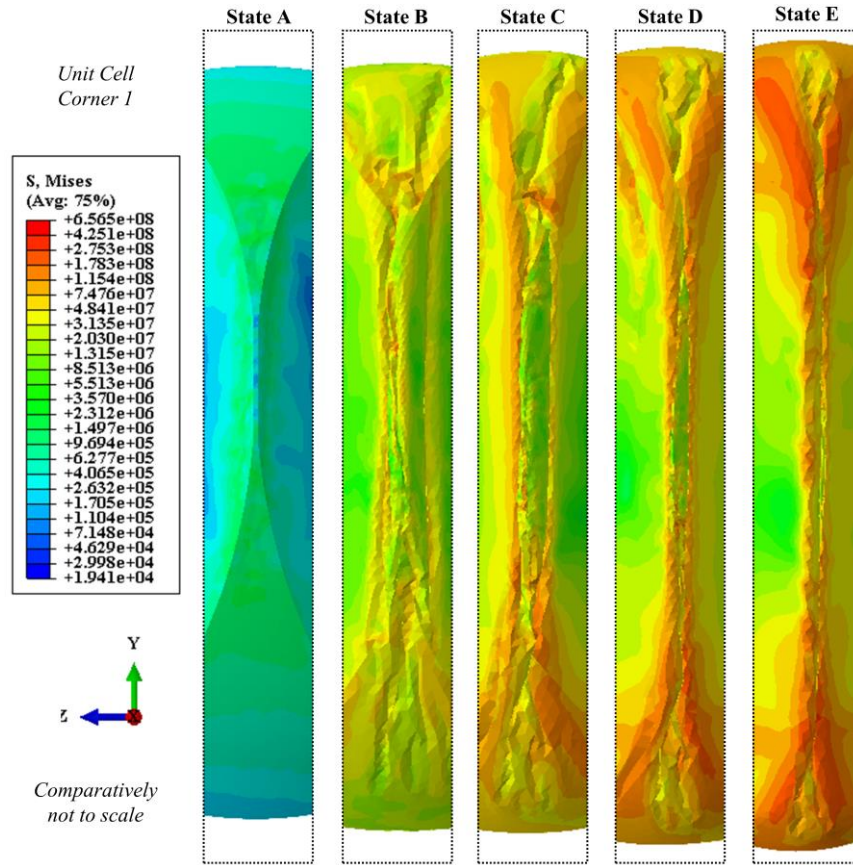


Fig. VI-17. Comparison of von Mises stress contours near a unit cell corner across loading states A-E.

Wrinkling, or the folding of the three UC corners, occurs asymmetrically even though the initial geometry is symmetrically defined. Fig. VI-18 shows the three UC corners (i.e., C1, C2, C3) at states A and B. State A shows variations in the stress field within these corners and initial development of distinct wrinkling patterns above and below each corner; refer to circled regions in Fig. VI-18. These relatively small variations develop into distinct wrinkling patterns due to the distinct strain accumulation associated with each corner⁶³. Such behavior results in an uneven stress field, where the location of the stress

⁶³ Wrinkle patterns are mesh-density and timescale dependent. Detailed wrinkle characterization of the solution in question requires a further refined mesh and a lower load rate in order to allow time-independent wrinkle development. The reasoning is detailed in the mesh-density and timescale dependencies discussion within Appendix C. Generally, wrinkling is a stiffness- and time-dependent wave propagation phenomenon, requiring appropriate timescale and mesh selection for accurate numerical representation [162].

maximum is dependent on the wrinkling pattern developed at any given load amplitude, but shown to be within the wrinkling pattern of one of these corners.

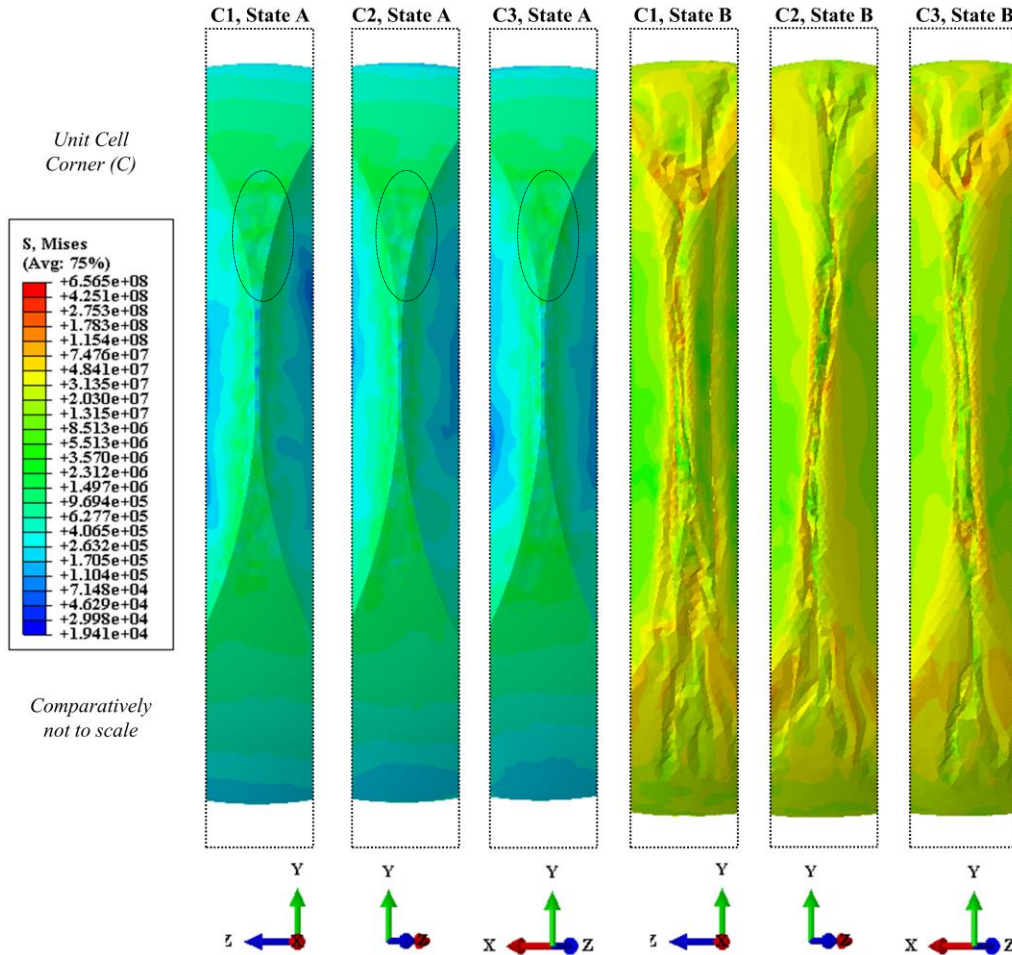


Fig. VI-18. Comparison of von Mises stress contours near the unit cell corners at states A and B.

All these UC shape changes observed throughout the load history were enabled by volume reductions and corresponding pressure increases associated with air compression within the cavity; coupled through the hydrostatic fluid-structure interaction formulation. Increases in cavity pressure (blue) and cavity surface area (red) are shown in Fig. VI-19. The cavity pressure reaches 2.9 times the applied pressure at the end of the load history; the latter is equal to the atmospheric and the initial cavity pressures for this case study.

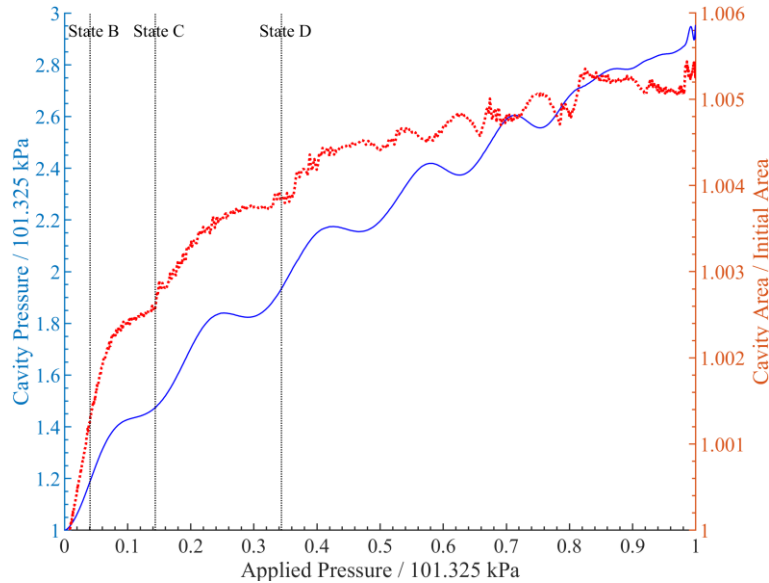


Fig. VI-19. Cavity pressure and area across the loading history.

Slope changes throughout the load history are attributed to significant fluctuations of wrinkling location and shape, where each significant fluctuation shifts the force resultant upward (away from the structural center), driving the UC away from the structural center and consequently releasing some of the fluid energy by decreasing its pressure. In other words, wrinkling effectively changes the surface direction locally, as well as the overall cavity volume. Such changes have a cumulative effect on the direction of the nodal forces resulting from the pressure distribution. The force resultant, calculated from these nodal forces, becomes directionally opposite to the structural center such that the UC temporarily displaces away from this center (i.e., changes the displacement direction). This upward displacement, in turn, exchanges fluid pressure with cavity volume, reflected as a cavity pressure reduction. The upward displacement continues until the additional load applied over new increments reverses the force resultant downward, restoring the displacement to a downward direction.

Furthermore, changes in the cavity surface area, shown in red (Fig. VI-19), provide a proxy for accumulated membrane strain as the surface facets used to represent the cavity are tied to, and therefore follow, the membrane elements (refer to section 5.4.2). Referring to Fig. VI-19, cavity surface area changes by a maximum of 0.55 % at the end of the load history. The slope changes are attributed to elements displaying large strain as wrinkling develops and fluctuates. The relatively small change in surface area suggests that membrane straining has a lesser contribution in the overall movement of the UC. In such case, the exclusion of inelastic behavior (which would otherwise cause inelastic strains in the small set of membrane elements displaying large strains) is not expected to provide a significant variation on the overall UC displacement, although it remains a valid consideration as an Icoron of specific parameters is studied further.

Energy Discussion

The energy balance augments the discussion above by providing energy distribution as a function of the work done by the applied load. Such discussion validates the use of the 0.05 s as an appropriate timescale for quasi-static responses of the UC, given that Abaqus/Explicit is a dynamic solver (i.e., explicit solution of dynamic equilibrium equations). In other words, it validates that the work done on the structure is primarily distributed as UC internal energy with minimal work reflected as kinetic energy and numerical enforcement methods, such as solution damping and contact.

With the above in mind, the energy balance, based on the model definitions (refer to Table VI-1), is given by:

$$E_w \cong E_m + E_{fc} + E_k + (E_v - E_{pw}) \quad (80)$$

where E_w is the external work performed by the applied load, E_m is the membrane (elastic) strain energy, E_{fc} is the energy accumulated by fluid cavity (i.e., air compression), E_k is the kinetic energy, E_v is the energy dissipated through linear bulk viscosity⁶⁴ and E_{pw} is the work done by the springs in the enforcement of contact (i.e., penalty contact formulation).

Fig. VI-20 shows the energy contribution of each component in Eq. 80, relative to the external work. The response is initially dominated by kinetic energy, primarily associated with the formation of contact interactions as the initial state had small deviations inherent to surface discretization. Membrane strain energy is the second component initially driving the response; this was observed in state A (Fig. VI-16), where air compressibility was insignificant. Nonetheless, fluid cavity energy readily becomes the major component driving the work done to the UC throughout the loading history, portrayed as significant increases in cavity pressure and minimal increases in cavity area (Fig. VI-19). Overall, above 90% of the work done is propagated as fluid cavity and membrane strain energies across approximately 93% of the loading history, suggesting the response found reaches a quasi-static approximation.

⁶⁴ Bulk viscosity introduces solution damping associated with volume strain rates. It is included by default in Abaqus/Explicit in an effort to improve solution performance of high-speed dynamic events and does not add to element stress as it is meant to be a numerical effect. A linear bulk viscosity is included for structural elements, such as membranes, and applied as viscosity pressure, $p_v = b\rho c_d L^e \dot{\epsilon}_{vol}$, where b is the damping coefficient (defaulted at 0.06), ρ is the material density, c_d is the current dilatational wave speed (Eq. 24 in section 2.2.2), L^e is the element's characteristic length and $\dot{\epsilon}_{vol}$ is the element's volumetric strain rate [59]. Results from otherwise equally-defined analyses of the UC considered in this case study with $b = 0, 0.03, 0.06$ were compared, finding that all quantities had insignificant variation throughout the load history. The exception to the latter were stresses and strains associated with the small set of elements showing large strains, which experienced large strain rates. The fact that these elements are a small set of the mesh justifies finding similar solutions across b values. As such, the solution presented in this chapter use the default b value as it stabilizes solutions while affecting the solution outputs insignificantly.

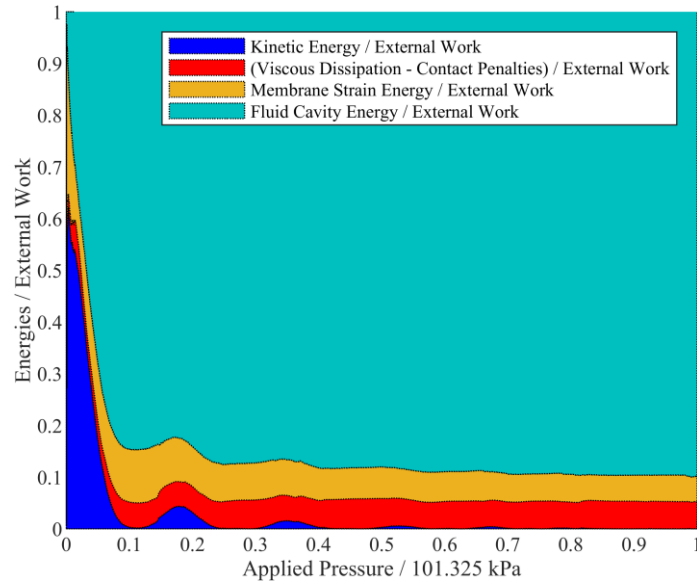


Fig. VI-20. Energy distribution across load history.

Weight-to-Buoyancy Estimate Based on Deformed State

The radial displacement evidenced in the stress and stress contours (Fig. VI-16) suggests that buoyancy cannot be achieved. To this point, the W/B ratio accounting for the air removed at any given point in the load history and the resulting radial displacement follows Eq. 78 above and is plotted in blue below (Fig. VI-21) along with the individual weight and buoyancy components (red). The W/B ratio curve is a reflection of the radial displacement, and it starts at full weight (i.e., no air evacuated). It increases from there as the mass of air removed at any given load is not sufficient to overcome the reduction in displaces air that drives buoyancy at said load. In other words, a buoyant design requires the buoyancy curve to cross the weight curve before the relative applied pressure is one.

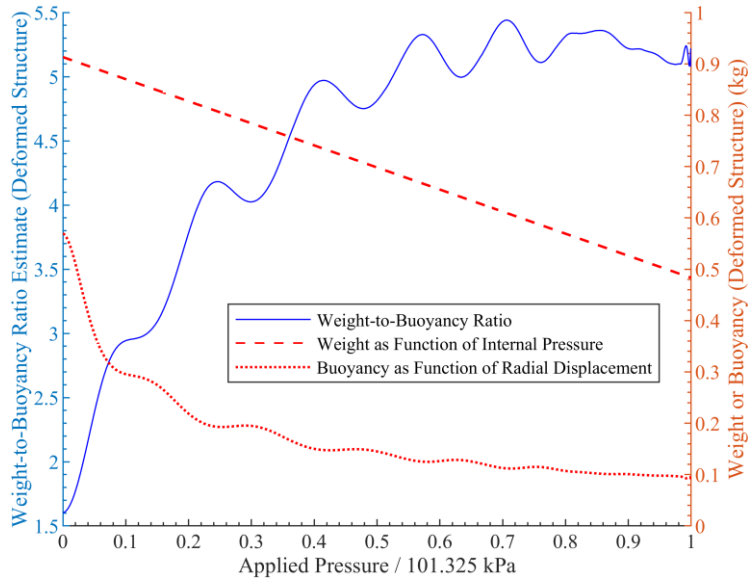


Fig. VI-21. Weight-to-buoyancy estimate (deformed state) along the load history.

Given the discussed above case study, there are several pathways one may take in regards to the design space exploration of the Icoron as a potential LTAS. The primary design variables as it concerns a quasi-static structure are {1} membrane modulus and density; {2} membrane thickness; {3} structural radius; {4} subdivision frequency; {5} UC cavity preload pressure (i.e., this refers to the two-step loading history discussed in section 6.1.2, where the UC cavity is inflated to a designated preload pressure prior to applying the internal pressure load resembling air removal); and {6} atmospheric altitude.

6.3.3. Modulus and Preload Pressure Effects

A question that may arise in the search for specific design parameters (e.g., subdivision frequency, modulus, thickness, structural radius) is the relevance of modulus and preload pressure in the overarching UC response. The following study briefly looks at these effects, limited to the geometry and model definitions used above. Solutions were obtained with the following variables modified: {1} membrane stiffness modulus $E = [1, 2, 3, 4, 5]E_{(i)}$, where $E_{(i)} = 4.36$ GPa as used above; and {2} cavity preload pressure

$P_{ca(preload)} = [1, 1.25, 1.5]P_{ca(i)}$, where $P_{ca(i)} = 101.325$ kPa, as used for the solution above. This produced 15 combinations (i.e., 14 additional analyses). Three response surfaces were plotted in Fig. VI-22, representing outputs at the end of the second (and last) step, where the applied load is equal to 101.325 kPa; those responses are air cavity pressure, cavity area and W/B ratio.

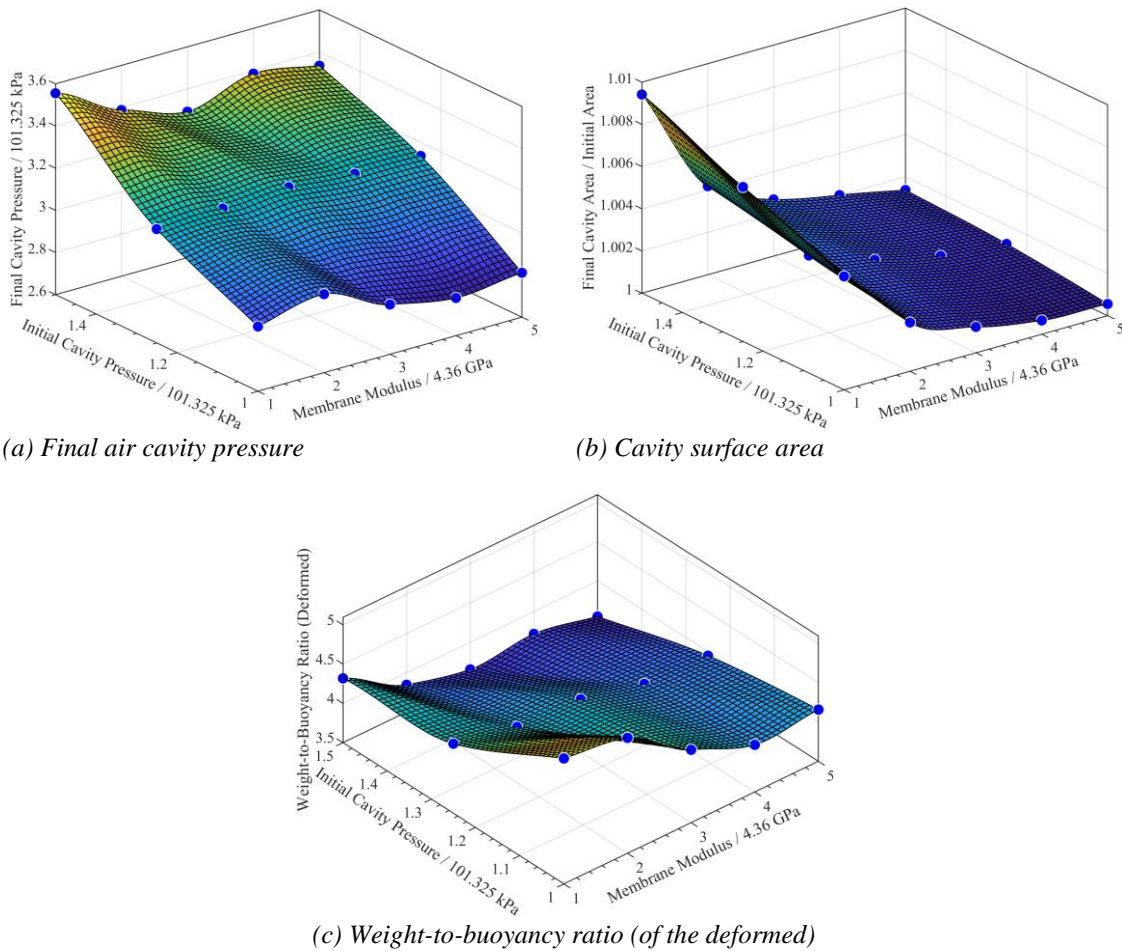


Fig. VI-22. Response surfaces representing the final deformed states as a function of modulus and preloaded cavity pressure.

The final air cavity pressure (Fig. VI-22a), $P_{ca(f)}$, shows a marked dependence on its preloaded pressure, where $P_{ca(f)}/P_{ca(preload)}$ varies from 2.9 when $P_{ca(preload)} = P_{ca(i)}$ down to 2.36 at $P_{ca(preload)} = 1.5P_{ca(i)}$. In other words, the ratio of the UC air

pressure before and after ‘vacuum loading’ (i.e., second step representing air removal from the Icoron’s internal cavity) is reduced with increases in preload pressure; that is because a higher preload pressure increases the bulk modulus of the enclosed air (i.e., ‘air-stiffening’ the UC), which in turn reduces air compressibility as a function of the ‘vacuum loading’, delaying wrinkling and membrane folding, and ergo delaying radial displacement. The final air cavity pressure dependence on membrane modulus was rather insignificant, compared to that of the preload pressure. On the contrary, the cavity surface area (Fig. VI-22b) has a strong dependence on membrane modulus with lesser dependence on the preload pressure. This is not surprising since, contrary to the air cavity pressure, the surface area is directly dependent on the membrane strains, and the inflation (first loading step) provides uniform pressure along all UC surfaces, which has an improved force balance, compared to the second step, and drives strain accumulation along the entire UC surface. Note, regardless, that surface area increases by a maximum of 10% at the maximum preload pressure and minimum modulus. This results in reduced W/B estimates (Fig. VI-22c), with both preload pressure and modulus increases. Most importantly, lesser effects related to the modulus are outperformed by those associated with preload pressure. This brief look at the effects that variation of modulus and preload pressure have on the structural response indicates that, in the search of a buoyant Icoron LTAS, priority should be given to managing the stresses associated with high preload pressures through UC radius and membrane thickness, rather than focusing on finding materials with high moduli.

An interesting association of the modulus with the thickness is what has been defined as *membrane modulus*, E^* (i.e., modulus times thickness such that $E^* = Eh_m$). The

membrane modulus is a convenient consideration because it drives displacement solutions, such that displacement fields are interchangeable if E^* remains constant across said solutions⁶⁵. This is relevant because the radial displacement u_r can be defined in terms of E^* , which in turn removes a variable from the design space in the search of a buoyant design.

6.4. Summary and Conclusions

In this chapter, the Icoron geometry is described following its development as an extension of the air-stiffened concept onto the vertices of icosahedron subdivisions. The development focuses on the geometrical description of the UC as a representative part of the overall (largely) symmetric design, where areas and volumes are characterized as sums of the UC contributions. This leads into UC radii and its variations as a function of subdivision frequency. A structural description is provided after, discussing UC load distribution, assembly representation and load balance. It is posed then that high preload pressures can minimize the radial displacement of the UC upon loading. The weight-to-buoyancy ratio (assuming rigidity) and stress estimates then support parameter selection for the case study, where preload pressures high enough to result in negligible force resultants leading to large structural radii and small subdivision frequencies; the latter detrimental to stress control.

⁶⁵ Membrane modulus has been used in literature [126] to describe displacement solutions for membranes. The validity of this concept was confirmed by considering two UC models of equal E^* but distinct E and h_m values, resulting in equivalent displacement fields. It is important to note, nonetheless, that the constitutive states reflected unique stress and strain values associated with the specific modulus and thickness.

The case study considers an Icoron of 0.48 m structural radius and 4,500 UCs of PET membrane. The model definitions are illustrated in Table VI-1. The following is said about these results, including the variation of the stiffness modulus and UC preload pressure:

1. The simulation methodology presented in Chapter 5 served to develop a quasi-static solution of the UC as a representation of an Icoron's response with one exception: an explicit solver was needed in order to characterize the structural response along the entire loading history because membrane wrinkling and contact drove nonconvergence of implicit solutions early in the time history. The use of an explicit solver in turn brought:
 - a. *The replacement of Lagrange multipliers* as an exact enforcement of the normal hard contact with penalty contact as an approximate enforcement; surface overlapping, which can be associated with approximate methods, was not observed in the solutions considered.
 - b. *An emphasis on loading rate*, or solution timescales, to ensure energy distributions result in a quasi-static approximation and maintain a feasible computing time. In this case, a timescale of 0.05 s was sufficient to develop a quasi-static solution.
 - c. Bulk viscosity effects were found to play an insignificant role on the displacement field and constitutive state, while providing numerical stabilization.

2. Membrane strains play a lesser role in the UC movement, validated by the effect of the membrane modulus on the UC displacement compared to the effect provided by the cavity preload pressure. Based on these observations, having a linear elastic approximation is a valid approach during the search of Icoron parameters that can lead to buoyant structures. Specifically, the potential accumulation of inelastic strains is expected to arise only at small portions of the regions of the UC where wrinkles manifest, insignificantly affecting the overall stability of the UC. That said, thermo- and visco-elasticity typically are important considerations when polymer membranes are being considered, particularly in time-dependent solutions.
3. Spherical stress is an appropriate estimate for the stress field away from wrinkled surfaces.
4. The potential development of wrinkles, as observed in the case study, requires mesh convergence studies based on solutions that are able to generate such wrinkles. In other words, linear solvers will not serve as proxies to establish appropriate discretization as wrinkle shape and frequency is not known a priori throughout the loading history.
5. The development of wrinkling patterns along the UC surfaces is highly mesh-density dependent. On the contrary, overarching solution outputs, such as UC cavity pressure and radial displacement, vary minimally with mesh density.

Given the analytic framework and the Icoron geometry provided, feasibility efforts primarily rely on finding the following design variable combination that allow for buoyancy after deformation while remaining consistent with targeted stress limits: {1} membrane thicknesses; {2} structural radius; {3} subdivision frequency; and {4} UC cavity preload pressure. The latter design variable plays a major role in the potential of the Icoron as an VLTAS because high preloads (up to the balanced pressure stated in Eq. 74) preserve buoyancy by minimizing radial displacement, while resulting in a high strength requirement. The opposite is also true.

VII. CONCLUSIONS

LTA systems have played a key role in the air domain as low speed, ultra-long endurance flight systems since their inception. Traced back to Francesco Lana de Terzi [137], the idea of having these systems maintain buoyancy without the use of LTA gases became an attractive notion yet to be realized outside the bounds of theoretical and analytical works [2–4,6,138–142]. Its realization has been bounded by specific stiffness and strength requirements commercially unavailable today, given the geometrical arrangements considered. Its pursue, nonetheless, has led to the development of unique concepts and design strategies that provide valuable insight on the behavior of extremely light structures. Air-stiffened designs grew out of these developments with the goal of closing the material gap by laying down an alternate path for the realization of these unique LTA structures. Accordingly, this dissertation is a compilation of the development trajectory that led to the concept, structural simulation framework and case studies of air-stiffened designs, establishing a generalizable framework that supports the search of optimal solutions within a variety of design spaces that can be constrained by present availability of material and manufacturing technology.

7.1. Summary of the Development Trajectory

Given the background on LTA systems, the development trajectory starts with (*Chapter III*) the characterization of cylindrical shell collapse solutions at increasingly smaller thicknesses of 1E-3 m down to 1E-6 m. Their comparison highlighted the changes in collapse behavior resulting from the loss of significant bending, as well as the dissipation of material inelastic effects as thickness is reduced. These findings, along with the use of

an implicit dynamic solver, started to shape the simulation framework of air-stiffened designs. (*Chapter IV*) A spherical sandwich shell with low-density cores was considered next in an effort to conserve significant bending stiffness given the light weight application, finding that stability requirements could not be met given all the low-density and commercially-available cores considered. More importantly, it led to the idea of air-stiffened designs by forcing the question of whether thin barriers shaped by air can be potential VLTAS. The third study focused on exploring spherical spaces through topology optimization, which cemented the need for a low-density material that supports or shapes the membranes displacing the atmosphere to enable buoyancy. As the idea of air-stiffened design was being formed, the fourth study validated the use of the classical hoop stress for initial design estimates, providing a measure of the approximate membrane radius and thickness required to maintain stress limits. (*Chapter V*) As the first air-stiffened design considered, the Helical is parametrized and characterized, showing geometric and contact nonlinearities as unstable responses developed. The instability observed prompted the development of the (*Chapter VI*) Icoron and its structural solutions in the context of VLTAS. The structural solutions for both designs supported the development of the simulation framework for air-stiffened designs, summarized next.

7.2. Summary of Findings

The structural characterization of air-stiffened designs, as defined, requires several analytical formulations suited to handle geometrically nonlinear behavior. The primary formulation is the air-membrane, or more generally, the fluid-structure interaction coupling the fluid state to the enclosing structure. This is indispensable because, as a compressible

fluid, the state of the air is not known a priori throughout the loading history and therefore cannot be simply represented by an equivalent pressure load. The work discussed used a hydrostatic representation of the fluid-structure interaction, in which the fluid state is uniform across the filled cavity containing it and coupled to it via a volume constraint. In the interest of quasi-static solutions, this interaction provided the necessary coupling while eliminating the need for coupled Eulerian-Lagrangian solutions where the Navier-Stokes equations would become part of the system of equations. The use of such hydrostatic interaction is adequate when the kinematics of fluid flow are negligible.

Solutions displaying large displacements drove the need for contact interactions that restricted the crossing of adjacent surfaces. From the structural perspective, this inclusion was essential in characterizing the behavior through and post-collapse for the Helical Sphere design, where normal contact drove membrane-to-membrane interactions, in turn enabling air-membrane interactions to shape the response as the load history progressed. In contrast, normal contact was used to represent symmetry conditions of the Icoron, enabling the use of a UC to represent its behavior. Furthermore, normal contact prevented surface regions from crossing as membrane wrinkling occurred.

Phenomena such as membrane wrinkling, contact and large displacements, which are enabled by the fluid-structure interaction, place an emphasis on geometrically nonlinear behavior. These drive the selection of solvers capable of handling such nonlinearities. For example, an implicit dynamic solver based on a backward Euler operator provided sufficient robustness to represent the behavior of the Helical Sphere, where membrane-to-membrane contact developed smoothly through the load history without any significant

discontinuities. On the contrary, the same implicit solver could not solve the frequent changes in surface directionality and contact regions observed in the UC (of the Icoron); the latter solved explicitly instead. The use of dynamic solvers to find quasi-static solutions required the consideration of energy ratios to ensure that energy distribution is representative of a quasi-static behavior; meaning that the majority of the work done by the applied loading becomes internal energy. This was primarily accomplished through numerical damping inherent in the Euler operator for the Helical Sphere solutions, and through loading rate and mesh density control for the Icoron solutions.

From the analytical perspective, air-stiffened designs have increased the level of complexity required to characterize the structural behavior of VLTAS designs subjected to loads consistent with air removal from their innermost cavities; that is compared to prior designs [2–4,6,138–142]. From the perspective of having potential as VLTAS, air-stiffened designs have shifted the prior requirement for considerably high specific moduli to one where stress limit becomes the primary material limitation. Such allows for an alternate search path, where limiting the accumulation of strain is secondary to limiting the accumulation of stress. The latter is controlled by the Icoron design flexibility in terms of the subdivision frequency and sizing characteristics, or more generally, by the definition of air-stiffened designs where connectivity arrangements are predicated on local membrane radii control as a proxy for stress control.

7.3. Recommendations

Given the concept of air-stiffened designs, the simulation framework, and the benefits of a highly symmetric design observed through the Icoron, there are several development paths envisioned:

- The search for optimal parameters for the Icoron design primarily given W/B, stress, and manufacturing constraints. This can be accomplished by having the optimization algorithm(s) drive FEM inputs, requiring days or weeks of computation time and limited design variable range to ensure model definitions can represent the constitutive behavior accurately, which in turn drives the FEM output-driven objectives and constraints. Or, it can be done by decoupling the FEM from the optimization algorithm and representing desired FEM output-driven objectives and constraints with response surfaces obtained and validated prior to the optimization runs.
- Regardless of the desire for an optimized solution, validation can entail higher-order modeling and experimentation. Higher-order modeling can include combinations of the following:
 - o UCs connected together to form a quadrant of the Icoron;
 - o Constitutive definitions that represent thermoviscoelastic behavior of polymer membranes forming the UCs;
 - o Time-dependent effects associated with preload and air removal rates;
 - o Whole flight simulations representing the movement of the structure through the atmosphere.

- Another path is the removal of the LTA weight constraint, leading to the generalization of air-stiffened designs as inflatable structures. Air-stiffened designs can be explored for utilization as barriers between environments with significant pressure differences, such as a pressurized vessel or vacuum chamber, where the goal can be to provide an enclosure with comparative loading capabilities at significant weight reduction.

Appendices

APPENDIX A. STABILITY AND MATERIAL NONLINEARITY SUPPORTING INFORMATION

A.1. Johnson-Cook Material Effects on Shell Stability

Contrary to the displacement-controlled results presented in section 3.3, load-controlled collapse analyses were conducted using the following material definitions: {1} linear elastic, {2} Johnson-Cook with isotropic hardening, {3} Johnson-Cook with isotropic hardening and thermal softening, and {4} Johnson-Cook with isotropic hardening, strain hardening, and thermal softening. The cylinder was discretized with 7,595 S8R elements as it resulted in a variation below 2% in and an asymmetric eigen shape. The latter is assumed by the theory that generated the critical load provided by Eq. 28 on pg. 58. The applied load, normalized by the predicted critical load, vs the axial displacement at a node located mid-length is plotted for each of material definitions in Fig. A-1.

The first notable change in slope occurs near $0.697P_{cr}$, which is when the local (element) instabilities show in the model; only possible because of the geometric imperfections added to the unloaded shell. From there, changes in slope occur between locations B and C, displayed in the curve as snapback behavior, and in the collapsed shell as new shapes of which each achieves a new state of equilibrium. After location C, the slope only changes slightly, which indicates that the shape found at location C is being progressively developed as it crushes, rather than changing shapes at new states of equilibrium are found. In other words, the shape is somewhat “locked” after location C.

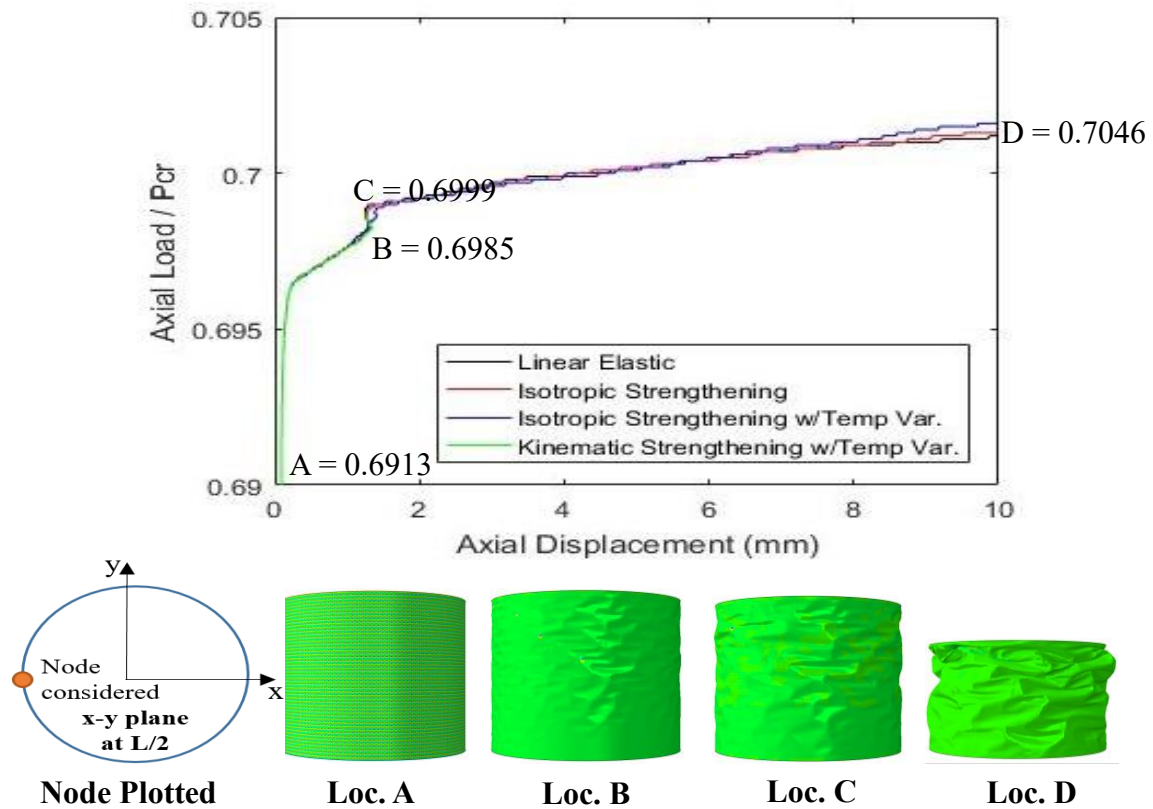


Fig. A-1. Load vs displacement with Johnson-Cook material effects.

The local instabilities that enable collapse also enables stress concentration near them. When these high stress regions reach yield stress values, the Johnson-Cook criteria “turns on” and the material stiffness of these regions changes, encouraging equilibrium around the newly found shape. Now, the model with the linear elastic material definition did not converge past location B. Given that the models are the same, with the exception of the material definition, it can be suggested that the change in material stiffness aides in convergence, as well as in defining a collapse path. Lastly, adding strain hardening and thermal softening did not significantly change the collapse path, compared to just isotropic

hardening. This a consequence of the quasi-static analysis, which does not produce strain rates and temperature⁶⁶ variations that significantly change the material stiffness.

A.2. Circular Cylindrical Shell Python Code to Generate Abaqus CAE

Using Abaqus v2016, the following Python code generates the Abaqus .cae file of the circular cylindrical shell for analysis, based on the input stated at the beginning of the code. The author suggests that Python codes are copied and pasted into a Python, or .py, interpreter for readability, and on the Abaqus command window for model generation. The author welcomes requests and inquiries regarding functions and codes used to develop the content of this dissertation.

```
## Cylindrical Shell Buckling Analysis ABAQUS-Python Script 20181023
# By Ruben Adorno

# *****
# Inputs
# *****
path      = 'I:\LTVA Research\Cylinder Continuation\Ver t6 Imp Mode 20\Scratch'      # working
directory, i.e., temp analysis files, etc.
scratch_folder = path                                # results files location
job_name = 'NLID_NLM_t6_ImpMode20'
model_name = 'Cyl NLID NLM t6 Mode20'
bucklefilename = 'CylShell_20181022_Buckle_Job_t6' # linear perturbation buckling '.fil' name, same as
Buckle odb name

# Step Info #####
step_type      = 5 # 1 = buckle, 2 = frequency, 3 = riks, 4 = NL Newton-Raphson w/Adaptive Stab..
                # 5 = Implicit Dynamics Quasi Static, 6 = Implicit Dynamics Moderate Dissipation
mat_type       = 1 # 0 = Elastic only, 1 = Elastic & Johnson-Cook Rate-Dependent Plastic
mat_rate_dependency = 0 # no Johnson-Cook rate dependency; applies for NL material only (mat_type =
1)
mat_temp_dependency = 0 # no Johnson-Cook temp effects (m = 0 in (T..)^(m)); applies for NL material
only (mat_type = 1)
inelastic_heat_frac = 0.9 # Abaqus default = 0.9
```

⁶⁶ Assumes an adiabatic process with 90% of the inelastic energy goes into temperature variation. Abaqus calculated the temperature increase at each element by obtaining the heat flux created by the inelastic strain. The heat flux $r^{inelastic} = \eta \sigma : \dot{\epsilon}^{inelastic} = \rho c(T) \dot{T}$, where η = inelastic heat fraction, σ = stress tensor, $\dot{\epsilon}^{inelastic}$ = inelastic strain rate tensor, ρ = mass density, c = inelastic heat fraction, and T = temperature.

```

BC_load      = 1 # displacement controlled instead of load controlled
load        = -1.7115611726E4      # Theoretical Pcr *multiplier to get most of Riks and NR
disp        = -10E-3                # amount of axial shortening applied for displacement-controlled analysis
Riks_set    = [1E-3,1E-15,1E-3,3000] # Riks Step Settings: initial, min & max arc increment, max # of
increments
NR_set      = [1E-3,1E-15,1E-3,3000,2E-7] # NR w/Adap Stab Set: initial, min & max load increment, max
# of increments, energy ratio for Adap Stab
ImpD_set    = [1E-3,1E-15,1E-3,3000] # Imp NL Dyn Quasi-Static Step Set: initial, min & max load
increment, max # of increments

# Constant Values #####
circum_edges_seed = 340
longit_edges_seed = 80
num_cores = 4
memory_usage = 90
num_GPUs = 0
skin_radius = 4.064e-01 # Kobayashi: 0.1
skin_length = 8.128e-01 # Kobayashi: 0.1139
skin_thickness = 1E-06 # Kobayashi: 0.247E-3
t = skin_thickness
imper = [t*0.0056, t*0.0056, t*0.0056, t*0.0056, t*0.0056, t*0.0056, \
        t*0.0056, t*0.0056, t*0.0056, t*0.0056, t*0.0056, t*0.0056, \
        t*0.0056, t*0.0056, t*0.0056, t*0.0056, t*0.0056, t*0.0056, \
        t*0, t*0] # fraction of mode 1-20 added as imperfection, previously 0.0036
skin_element_type1 = S4R # 2D Shell: S4R 3D Shell: SC8R 3D Stress: C3D8R
skin_element_type2 = S3 # 2D Shell: S3 3D Shell: SC6R 3D Stress: C3D6
BC_location = 2 # 1 = top BC off, 2 = top BC on
BC_type_bottom_edge = 2 # 1 = SS, 2 = Fixed
BC_type_top_edge = 2 # 1 = SS, 2 = Fixed
skin_poisson = 0.361 # Kobayashi: 0.3
skin_modulus = 1.1E11 # Kobayashi: 5.56E9
skin_density = 4424.000000
skin_A = 8.618000e+08/2
skin_B = 3.309000e+08
skin_n = 3.400000e-01
skin_m = 8.000000e-01
skin_C = 1.200000e-02
skin_eps0 = 1.000000e+00
skin_Tm = 1.905220e+03
skin_T0 = 2.731500e+02
skin_d1 = -9.000000e-02
skin_d2 = 2.500000e-01
skin_d3 = 5.000000e-01
skin_d4 = 1.400000e-02
skin_d5 = 3.870000e+00
skin_specific_heat = 526 # J/(kg-K), ASTM Grade 5 titanium; Ti6Al4V,
http://www.matweb.com/search/DataSheet.aspx?MatGUID=b350a789eda946c6b86a3e4d3c577b39

# *****
# CAE Creation Code
# *****

```

```

# Loads Modules' Calls
*****
import os
from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

# Sets Working Directory
*****
os.chdir(path)

# Load variables
*****
execfile(Var_file)

# Model name and creation
*****
mdb.Model(modelType=STANDARD_EXPLICIT, name=model_name)

# Creates Part *****
mdb.models[model_name].ConstrainedSketch(name='__profile__', sheetSize=2.0)
mdb.models[model_name].sketches['__profile__'].CircleByCenterPerimeter(center=(
    0.0, 0.0), point1=(skin_radius, 0.0))
mdb.models[model_name].Part(dimensionality=THREE_D, name='Part-1', type=
    DEFORMABLE_BODY)
mdb.models[model_name].parts['Part-1'].BaseShellExtrude(depth=skin_length, sketch=
    mdb.models[model_name].sketches['__profile__'])

# Creates Material
*****
mdb.models[model_name].Material(name='Material-1')
mdb.models[model_name].materials['Material-1'].Elastic(table=((skin_modulus, skin_poisson), ))
mdb.models[model_name].materials['Material-1'].Density(table=((skin_density, ), ))
Adiabatic = OFF
if mat_type == 1:
    if mat_temp_dependency == 0:
        skin_m = 0
    elif mat_temp_dependency == 1:
        mdb.models[model_name].materials['Material-1'].InelasticHeatFraction(fraction=inelastic_heat_frac)
        mdb.models[model_name].materials['Material-1'].SpecificHeat(table=((skin_specific_heat, ),))
        Adiabatic = ON

mdb.models[model_name].materials['Material-1'].Plastic(hardening=JOHNSON_COOK,

```

```

    table=((skin_A, skin_B, skin_n, skin_m, skin_Tm, skin_T0), )
if mat_rate_dependency == 1:
    mdb.models[model_name].materials['Material-1'].plastic.RateDependent(
        table=((skin_C, skin_eps0), ), type=JOHNSON_COOK)
    #mdb.models[model_name].materials['Material-1'].JohnsonCookDamageInitiation(
    # table=((skin_d1, skin_d2, skin_d3, skin_d4, skin_d5, skin_Tm, skin_T0, skin_eps0), ))

# Creates Profile, Creates Section, and Assigns Section
*****
mdb.models[model_name].parts['Part-1'].Set(faces=
    mdb.models[model_name].parts['Part-1'].faces.getSequenceFromMask(['#1'], ), name='Whole Shell')

# Composite Shell Section (1-ply) (allows for plastic strains for 2D shell mesh)
mdb.models[model_name].CompositeShellSection(idealization=NO_IDEALIZATION,
integrationRule=SIMPSON, layup=(SectionLayer(
    thickness=skin_thickness, material='Material-1', plyName=''), ), name='Section-1',
poissonDefinition=DEFAULT,
    preIntegrate=OFF, symmetric=False, temperature=GRADIENT, thicknessModulus=None,
thicknessType=UNIFORM, useDensity=OFF)

mdb.models[model_name].parts['Part-1'].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models[model_name].parts['Part-1'].sets['Whole Shell'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)

# Create Instance
*****
mdb.models[model_name].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models[model_name].rootAssembly.Instance(dependent=OFF, name='Part-1-1',
    part=mdb.models[model_name].parts['Part-1'])

# Creates Edges Sets
*****
mdb.models[model_name].rootAssembly.PartitionFaceByShortestPath(faces=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].faces.getSequenceFromMask(['#1'], ),
    point1=mdb.models[model_name].rootAssembly.instances['Part-1-1'].InterestingPoint(
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges[1], MIDDLE),
    point2= mdb.models[model_name].rootAssembly.instances['Part-1-1'].InterestingPoint(
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges[0], MIDDLE))
mdb.models[model_name].rootAssembly.PartitionEdgeByPoint(edge=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges[1], point=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].InterestingPoint(
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges[1], MIDDLE))
mdb.models[model_name].rootAssembly.Set(edges=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges.findAt(((
    -0.707106*skin_radius, -0.707106*skin_radius, 0.0), ), ((0.707106*skin_radius, 0.707106*skin_radius,
    0.0), ), ), name='Lower Edge')
mdb.models[model_name].rootAssembly.Set(edges=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges.findAt(((
    0.707106*skin_radius, -0.707106*skin_radius, skin_length), ), ((-0.707106*skin_radius,
    0.707106*skin_radius, skin_length), ), ), name='Upper Edge')
mdb.models[model_name].rootAssembly.Set(edges=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges.findAt(((

```

```

-skin_radius, 0.0, 0.875*skin_length), ), ((-skin_radius, 0.0, 0.75*skin_radius), ), ), name='Longitudinal
Edge')
mdb.models[model_name].rootAssembly.Set(vertices=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].vertices.findAt((( -skin_radius, 0.0,
skin_radius), )),name='L_2 Pt')

# Creates Mesh *****
mdb.models[model_name].rootAssembly.setElementType(
    elemTypes=(ElemType(elemCode=skin_element_type1, elemLibrary=STANDARD,
secondOrderAccuracy=ON), ElemType(elemCode=skin_element_type2, elemLibrary=STANDARD)),
    regions=(mdb.models[model_name].rootAssembly.instances['Part-1-
1'].faces.getSequenceFromMask(['#1'], ), ), )
mdb.models[model_name].rootAssembly.setMeshControls(elemShape=QUAD, regions=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].faces.getSequenceFromMask(['#1'], ), ),
technique=SWEEP)

mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER, edges=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges.findAt(((
-skin_radius, 0.0, 0.875*skin_length), ), ((-skin_radius, 0.0, 0.75*skin_radius), ), ),
number=longit_edges_seed/2)
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER, edges=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges.findAt(((
-0.707106*skin_radius, -0.707106*skin_radius, 0.0), ), ((0.707106*skin_radius, 0.707106*skin_radius,
0.0), ), ), number=circum_edges_seed/2)
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER, edges=
    mdb.models[model_name].rootAssembly.instances['Part-1-1'].edges.findAt(((
0.707106*skin_radius, -0.707106*skin_radius, skin_length), ), ((-0.707106*skin_radius,
0.707106*skin_radius, skin_length), ), ), number=circum_edges_seed/2)
mdb.models[model_name].rootAssembly.generateMesh(regions=(mdb.models[model_name].rootAssembl
y.instances['Part-1-1'], ))

# Creates Reference Point (RP) (to later constraint Edge to RP)
*****
mdb.models[model_name].rootAssembly.ReferencePoint(point=(0.0, 0.0, skin_length))
mdb.models[model_name].rootAssembly.Set(name='Upper Edge RP Set',
referencePoints=(mdb.models[model_name].rootAssembly.referencePoints.findAt((0.0, 0.0, skin_length)),
))

# Creates Constraint between Edges and RP
*****
mdb.models[model_name].RigidBody(name='Upper Edge Constraint', tieRegion=
    mdb.models[model_name].rootAssembly.sets['Upper Edge'], refPointRegion=
    Region(referencePoints=(mdb.models[model_name].rootAssembly.referencePoints.findAt((0.0, 0.0,
skin_length)), )))

# Creates Step
*****
*****
if step_type == 1:
    mdb.models[model_name].BuckleStep(maxIterations=5000, name='Buckle', numEigen=20,
previous='Initial', vectors=20)
    F_load = -1 # concentrated load applied at RP
elif step_type == 2:

```

```

mdb.models[model_name].FrequencyStep(name='Frequency', previous='Initial', numEigen=5)
elif step_type == 3:
    F_load = load # concentrated load applied at RP
    # Creates NL Riks Step
    mdb.models[model_name].StaticRiksStep(extrapolation=LINEAR,
    initialArcInc=Riks_set[0], minArcInc=Riks_set[1], maxArcInc=Riks_set[2], maxLPF=1,
    maxNumInc=int(Riks_set[3]), name='Buckle', nlgeom=ON, previous='Initial',
    useLongTermSolution=False, adiabatic=Adiabatic)
    # Creates Outputs Requests
    mdb.models[model_name].fieldOutputRequests['F-Output-1'].setValues(
    variables=('S', 'MISESMAX', 'PE', 'PEEQ', 'PEEQMAX', 'PEQC', 'PEMAG', 'E', 'LE', 'ER', 'U', 'RF', 'CF',
    'NFORC', 'COORD', 'STATUS'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='Load Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set']
    , sectionPoints=DEFAULT, variables=('U3', 'CF3','RF','TF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='L_2 Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['L_2 Pt']
    , sectionPoints=DEFAULT, variables=('U', 'CF','RF','TF'), frequency=1)
elif step_type == 4:
    F_load = load # concentrated load applied at RP
    # Creates NL NR Step
    mdb.models[model_name].StaticStep(adaptiveDampingRatio=0.05, continueDampingFactors=False,
    extrapolation=PARABOLIC, initialInc=NR_set[0], minInc=NR_set[1], maxInc=NR_set[2],
    maxNumInc=NR_set[3],
    name='Buckle', nlgeom=ON, previous='Initial', solutionTechnique=
    QUASI_NEWTON, stabilizationMagnitude=NR_set[4], stabilizationMethod=
    DISSIPATED_ENERGY_FRACTION, useLongTermSolution=False, adiabatic=Adiabatic)
    # Creates Outputs Requests
    mdb.models[model_name].fieldOutputRequests['F-Output-1'].setValues(
    variables=('S', 'MISESMAX', 'PE', 'PEEQ', 'PEEQMAX', 'PEQC', 'PEMAG', 'E', 'LE', 'ER', 'U', 'RF', 'CF',
    'NFORC', 'COORD', 'STATUS', 'THE', 'SF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='Load Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set']
    , sectionPoints=DEFAULT, variables=('U3', 'CF3','RF','TF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='L_2 Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['L_2 Pt']
    , sectionPoints=DEFAULT, variables=('U', 'CF','RF','TF'), frequency=1)
elif step_type == 5:
    F_load = load # concentrated load applied at RP
    # Creates Implicit Dynamic Quasi-Static Step
    mdb.models[model_name].ImplicitDynamicsStep(alpha=DEFAULT,
    amplitude=RAMP,extrapolation=PARABOLIC,
    application=QUASI_STATIC, initialConditions=OFF, initialInc=ImpD_set[0],
    maxInc=ImpD_set[2], maxNumInc=ImpD_set[3], minInc=ImpD_set[1],
    name='Buckle', nlgeom=ON, nohaf=OFF, previous='Initial', solutionTechnique=QUASI_NEWTON,
    adiabatic=Adiabatic)
    # Creates Outputs Requests
    mdb.models[model_name].fieldOutputRequests['F-Output-1'].setValues(

```

```

    variables=('S', 'MISESMAX', 'PE', 'PEEQ', 'PEEQMAX', 'PEQC', 'PEMAG', 'E', 'LE', 'ER', 'EE', 'IE', 'U',
    'RF', 'CF', 'NFORC', 'COORD', 'STATUS', 'THE', 'SF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='Load Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set']
    , sectionPoints=DEFAULT, variables=('U3', 'CF3', 'RF', 'TF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='L_2 Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['L_2 Pt']
    , sectionPoints=DEFAULT, variables=('U', 'CF', 'RF', 'TF'), frequency=1)
elif step_type == 6:
    F_load = load # concentrated load applied at RP
    # Creates Implicit Dynamic Quasi-Static Step
    mdb.models[model_name].ImplicitDynamicsStep(alpha=DEFAULT,
    amplitude=RAMP, extrapolation=PARABOLIC,
    application=MODERATE DISSIPATION, initialConditions=OFF, initialInc=ImpD_set[0],
    maxInc=ImpD_set[2], maxNumInc=ImpD_set[3], minInc=ImpD_set[1],
    name='Buckle', nlgeom=ON, nohaf=OFF, previous='Initial', solutionTechnique=QUASI_NEWTON,
    adiabatic=Adiabatic)
    # Creates Outputs Requests
    mdb.models[model_name].fieldOutputRequests['F-Output-1'].setValues(
    variables=('S', 'MISESMAX', 'PE', 'PEEQ', 'PEEQMAX', 'PEQC', 'PEMAG', 'E', 'LE', 'ER', 'EE', 'IE', 'U',
    'RF', 'CF', 'NFORC', 'COORD', 'STATUS', 'THE', 'SF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='Load Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set']
    , sectionPoints=DEFAULT, variables=('U3', 'CF3', 'RF', 'TF'), frequency=1)
    mdb.models[model_name].HistoryOutputRequest(createStepName='Buckle',
    name='L_2 Point History Output', rebar=EXCLUDE,
    region=mdb.models[model_name].rootAssembly.sets['L_2 Pt']
    , sectionPoints=DEFAULT, variables=('U', 'CF', 'RF', 'TF'), frequency=1)

# Creates BC *****
if BC_type_bottom_edge == 1:
    u_bottom = [0, 0, 0, UNSET, UNSET, UNSET] # Simply Supported: [0, 0, 0, UNSET, UNSET,
    UNSET]
elif BC_type_bottom_edge == 2:
    u_bottom = [0, 0, 0, 0, 0, 0] # Encastre/Fixed: [0, 0, 0, 0, 0, 0]

if BC_type_top_edge == 1: # [u1, u2, u3, ur1, ur2, ur3] of BC along bottom; 0 = no disp, UNSET is free in
that DOF
    u_top = [0, 0, UNSET, UNSET, UNSET, UNSET] # Simply Supported: [0, 0, UNSET, UNSET,
    UNSET, UNSET]
elif BC_type_top_edge == 2:
    u_top = [0, 0, UNSET, 0, 0, 0] # Encastre/Fixed: [0, 0, UNSET, 0, 0, 0]

if BC_location == 1:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET, buckleCase=
    PERTURBATION_AND_BUCKLING, createStepName='Initial', distributionType=
    UNIFORM, fieldName='', fixed=OFF, localCsys=None, name='BC Along Lower Edge', region=
    mdb.models[model_name].rootAssembly.sets['Lower Edge'], u1=u_bottom[0], u2=u_bottom[1],
    u3=u_bottom[2], ur1=u_bottom[3], ur2=u_bottom[4], ur3=u_bottom[5])
elif BC_location == 2:

```

```

mdb.models[model_name].DisplacementBC(amplitude=UNSET, buckleCase=
    PERTURBATION_AND_BUCKLING, createStepName='Initial', distributionType=
    UNIFORM, fieldName="", fixed=OFF, localCsys=None, name='BC Along Lower Edge', region=
    mdb.models[model_name].rootAssembly.sets['Lower Edge'], u1=u_bottom[0], u2=u_bottom[1],
u3=u_bottom[2], ur1=u_bottom[3], ur2=u_bottom[4], ur3=u_bottom[5])
mdb.models[model_name].DisplacementBC(amplitude=UNSET, buckleCase=
    PERTURBATION_AND_BUCKLING, createStepName='Initial', distributionType=
    UNIFORM, fieldName="", fixed=OFF, localCsys=None, name='BC Along Upper Edge', region=
    mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set'], u1=u_top[0], u2=u_top[1],
u3=u_top[2], ur1=u_top[3], ur2=u_top[4], ur3=u_top[5])

# Creates Load *****
if step_type == 3 or step_type == 4 or step_type == 5 or step_type == 6 and BC_load == 1:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET, createStepName='Buckle',
        distributionType=UNIFORM, fieldName="", fixed=OFF, localCsys=None, name=
        'Load BC', region=mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set'], u1=UNSET,
        u2=UNSET, u3=disp, ur1=UNSET, ur2=UNSET, ur3=UNSET)
elif step_type == 1 or step_type == 3 or step_type == 4 or step_type == 5 or step_type == 6:
    mdb.models[model_name].rootAssembly.Set(name='Upper Edge RP Set',
        referencePoints=(mdb.models[model_name].rootAssembly.referencePoints.findAt((0.0, 0.0,
skin_length)), ))
    mdb.models[model_name].ConcentratedForce(cf3=F_load, createStepName='Buckle',
        distributionType=UNIFORM, field="", localCsys=None, name='Upper Edge Load',
        region=mdb.models[model_name].rootAssembly.sets['Upper Edge RP Set'])

# Adds Imperfection
*****
*****
if step_type == 3 or step_type == 4 or step_type == 5 or step_type == 6:
    # Adds Imperfections from Linear Buckle Step File, as Initial Configuration
    x = '*Imperfection, File='+bucklefilename+', Step=1\n1, '+str(imper[0])+'\n2, '+str(imper[1])+'\n3,
'+str(imper[2])+'\n4, '+str(imper[3])+'\n5, '+str(imper[4])\
    +'\n6, '+str(imper[5])+'\n7, '+str(imper[6])+'\n8, '+str(imper[7])+'\n9, '+str(imper[8])+'\n10,
'+str(imper[9])+'\n11, '+str(imper[10])+'\n12, '+str(imper[11])\
    +'\n13, '+str(imper[12])+'\n14, '+str(imper[13])+'\n15, '+str(imper[14])+'\n16, '+str(imper[15])+'\n17,
'+str(imper[16])+'\n18, '+str(imper[17])\
    +'\n19, '+str(imper[18])+' \n20, '+str(imper[19])
    # Creates imperfection keyword string based on selected inputs
    mdb.models[model_name].keywordBlock.synchVersions(storeNodesAndElements=True) # Assigns the
model keyword to the 'sieBlocks' string
    try: # Removes the 'Imperfection...' string from the Keywords, if present
        l = mdb.models[model_name].keywordBlock.sieBlocks
        idx = [index for index, s in enumerate(l) if 'Imperfection' in s][0]
        mdb.models[model_name].keywordBlock.replace(idx, '\n')
    except:
        pass

    idy = mdb.models[model_name].keywordBlock.sieBlocks.index("*** -----
-----\n** \n** STEP: Buckle\n** ") # Finds location of the NL Riks Step
    mdb.models[model_name].keywordBlock.insert(idy-1, x) # Adds 'Imperfection...' keyword before the
NL Riks Step
    mdb.models[model_name].keywordBlock.synchVersions(storeNodesAndElements=True) # Syncs
keywordBlock object changes with CAE model

```

```

# Adds Displacements 'U' through '*Node File' Keyword, Hence to the Input File (keywordBlock object
definition on Abaqus Scripting Reference Manual)
*****
if step_type == 1:
    mdb.models[model_name].keywordBlock.synchVersions(storeNodesAndElements=True) # Assigns the
model keyword to the 'sieBlocks' string
    idx = mdb.models[model_name].keywordBlock.sieBlocks.index("*End Step") # Finds location of '*End
Step' in keyword array
    mdb.models[model_name].keywordBlock.insert(idx-1, "\n*EL FILE\nE, LE, ER, PE, PEQC, S') # Adds
'*Node File' to Keywords before '*End Step'; creates .fil with nodal displacements
    mdb.models[model_name].keywordBlock.insert(idx-1, "\n*Node File\nU, CF, RF, COORD') # Adds
'*Node File' to Keywords before '*End Step'; creates .fil with nodal displacements
    mdb.models[model_name].keywordBlock.insert(idx-1, "\n*Node Print\nU") # Adds '*Node Print' to
Keywords before '*End Step'; adds nodal displacements to .dat file
    mdb.models[model_name].keywordBlock.insert(idx-1, "\n*FILE FORMAT, ASCII') # Adds '*FILE
FORMAT, ASCII' to Keywords before '*End Step'; changes .fil format to ASCII (to make it readable by
MATLAB
    mdb.models[model_name].keywordBlock.synchVersions(storeNodesAndElements=True) # Syncs
keywordBlock object changes with CAE model

# Creates and Submits Job
*****
mdb.Job(atTime=None, contactPrint=OFF, description="", echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=memory_usage, memoryUnits=PERCENTAGE, model=model_name, modelPrint=OFF,
    multiprocessingMode=DEFAULT, name=job_name, nodalOutputPrecision=FULL,
    numCpus=num_cores, numDomains=num_cores, numGPUs=num_GPUs, queue=None,
resultsFormat=ODB, scratch=scratch_folder, type=
    ANALYSIS, userSubroutine="", waitHours=0, waitMinutes=0)

```

A.3. Circular Cylindrical Shell Python Code to Generate Outputs

Using Abaqus v2016, the following Python code extracts the outputs from the ODB output file created by Abaqus and generates three .txt files with vertical displacement, plastic dissipation and strain energy quantities for every time increment. The author suggests that Python codes are copied and pasted into a Python, or .py, interpreter for readability, and on the Abaqus command window for model generation. The author welcomes requests and inquiries regarding functions and codes used to develop the content of this dissertation.

Cylindrical Shell Output Request from ODB to TXT and ABAQUS Graphs

Input: the following Info Regarding the ODB File

ODBfilename = 'NLID_NLM_t6_v2'

ODBpath = 'C:\Test'

Reportpath = ODBpath

stepname = 'Buckle'

import os

from part import*

from material import*

from section import*

from assembly import*

from step import*

from interaction import*

from load import*

from mesh import*

from job import*

from sketch import*

from visualization import*

from connectorBehavior import*

from odbAccess import *

from abaqusConstants import *

from odbMaterial import *

from odbSection import *

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

session.xyReportOptions.setValues(numDigits=12)

Sets Working Directory

os.chdir(ODBpath)

Opens the ODB

odb = session.openOdb(name=ODBpath+'\\'+ODBfilename+'.odb')

session.viewports['Viewport: 1'].setValues(displayedObject=odb)

assembly = odb.rootAssembly

Writes RF3 vs U3 at RP

xy_result = session.XYDataFromHistory(name='PatRP', odb=odb,
outputVariableName='Reaction force: RF3 PI: rootAssembly Node 1 in NSET Upper Edge RP Set',)

xy_result = session.XYDataFromHistory(name='U3atRP', odb=odb,
outputVariableName='Spatial displacement: U3 PI: rootAssembly Node 1 in NSET Upper Edge RP Set',
)

x1 = session.xyDataObjects['U3atRP']

x2 = session.xyDataObjects['PatRP']

session.writeXYReport(fileName=ODBfilename+' Results PvsU.txt', xyData=(x1, x2))

Writes the Plastic Dissipation Energy versus Time for the Whole Model

f = open(ODBfilename+' Results PDE.txt','w')

energy = odb.steps[stepname].historyRegions['Assembly ASSEMBLY'].historyOutputs['ALLPD']

f.write('Plastic Dissipation Energy versus Time for the whole model\n')

f.write('Time Plastic Dissipation Energy\n')

```
for time in energy.data:
    f.write('%f %.20f\n'%(time[0],time[1],))

f.close()

# Writes the Strain Energy versus Time for the Whole Model
f = open(ODBfilename+' Results SE.txt','w')
energy = odb.steps[stepname].historyRegions['Assembly ASSEMBLY'].historyOutputs['ALLSE']
f.write('Strain Energy versus Time for the whole model\n')
f.write('Time Strain Energy\n')
for time in energy.data:
    f.write('%f %.20f\n'%(time[0],time[1],))

f.close()
```

APPENDIX B. HELICAL SPHERE AIR-STIFFENED DESIGN SUPPORTING INFORMATION

B.1. Inflated Circular Membrane Finite Element Results

The analytical and numerical solutions of an inflated circular membrane presented in Ref. [26] and Ref. [27] support the validation of the modeling components presented in Chapter V as representative of the mechanics observed in long-time solutions of gas-filled membrane structures. With this in mind, an inflated circular membrane was modeled with geometric nonlinearity and a hydrostatic representation of the air-membrane interaction, solved implicitly with the backward Euler operator. The circular membrane was initially flat with a radius of 0.1425 m, thickness of 12.2 μm , elastic modulus times thickness of 311,488 Pa-m, Poisson ratio of 0.38 and density of 1390 kg/m^3 . The membrane was first inflated to 400 kPa, followed by an external pressure load of 150 kPa or 300 kPa. All of the above follow Ref. [27]. Two cavity sizes with air at 288.15 K were considered in order to evaluate any effects associated with cavity size during loading as a result of air compression. Fig. B-1 shows the final state of the models for each of the cavity sizes, where cavities are shown in green and membrane states after inflation and loading in red. The membranes were defined with the same radius. In both cases, the membranes were defined with 1102 linear membrane elements, while the cavities used linear surface elements containing no inherent stiffness. The cavity, including the membrane's outer rim, were fixed throughout the simulation.

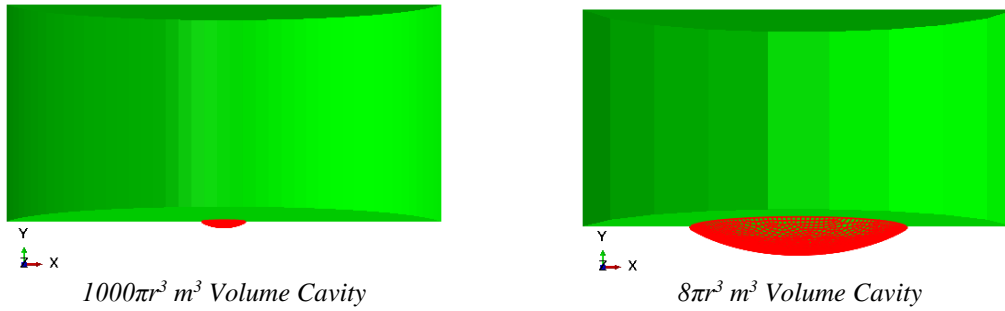


Fig. B-1. Inflated circular membrane with cavity at final state.

Commensurate with the figures presented in Ref. [26] and Ref. [27], Fig. B-2 shows the x coordinate (i.e., location of nodes along the radius) vs. the y coordinate (i.e., resulting vertical displacement at corresponding nodes) for 150 kPa and 300 kPa pressure loadings of both models. Notably, there are no appreciable differences associated between the two cavity volumes considered. Most importantly, Fig. B-2 shows the trend observed in Fig. 3 of Ref. [27], where numerical solutions are compared to Bouzidi et al. analytical solution [26]. Furthermore, the resulting maximum displacement is within approximately 8% and 6% of those in Fig. 3 of Ref. [27] for 150 kPa and 300 kPa, respectively.

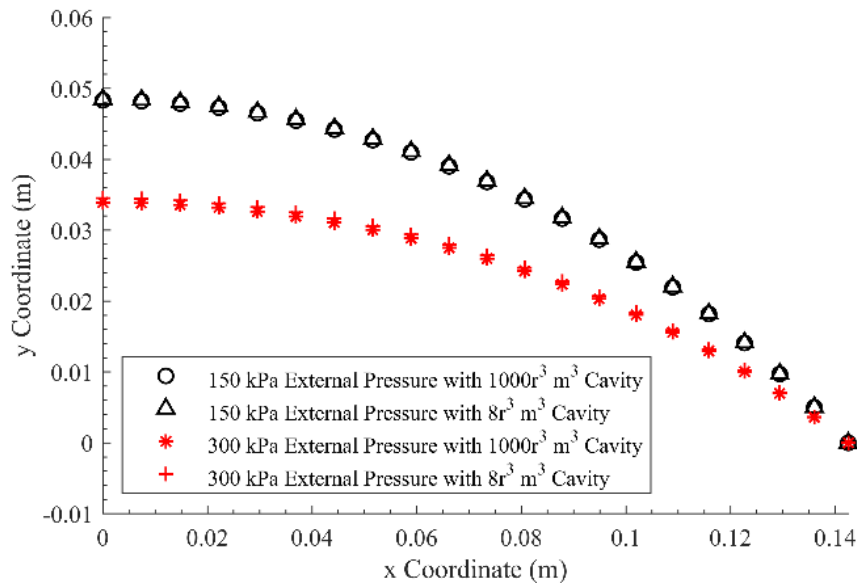


Fig. B-2. Final state of inflated circular membranes.

B.2. Convergence Study Results

The convergence study associated with the structural response presented in Chapter V was conducted with a reduced model in order to efficiently determine the mesh density required to minimize output variation. The reduced order model was based on the axisymmetric representation of the spherical membrane, referred to as T_0 and located at the north pole of the toroidal representation of $n = 41$. A rigid contact line in the axisymmetric representation was added in order to subject T_0 to contact and cavity volume changes that represent the type of conditions each of the membrane cavities are subjected to during the response of the axisymmetric model presented. The model is presented in Fig. B-3. Such model uses the same simulation techniques, material properties, and solver selection discussed in Chapter V, except for contact between the membrane and the rigid line being defined as frictionless and with separation after contact allowed. The latter allows free sliding, while contact pressures and cavity volume/pressure changes develop.

The parameters controlled in the convergence analysis are {1} the use of linear axisymmetric membrane (MAXI) or axisymmetric shell (SAX1) elements to discretize the former, and {2} a target uniform element size (i.e., distance between adjacent seeds) referred to here as ‘seed’, such that $\text{seed} = x \cdot t_m$, where x is the parameter changed and t_m is the membrane thickness. These parameters and corresponding results are presented in Table B-1; each of the results were extracted from the last increment, corresponding to an applied pressure of 101.325 kPa (i.e., sea level pressure). In other words, the results reflect analyses that went through the entire prescribed load history. Note that seed/t_m and number

of elements are inversely proportional (e.g., a $\text{seed}/t_m = 4$ corresponds to 781 elements, while a $\text{seed}/t_m = 100$ corresponds to 31 elements). A minimum $\text{seed}/t_m = 4$ was used.

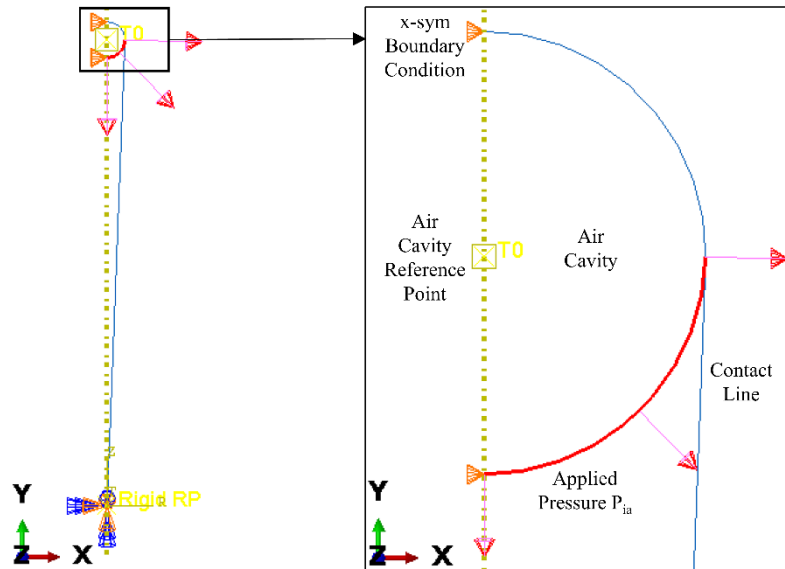


Fig. B-3. Reduced order model.

That said, Figs. B-4 to B-8 plot each of the results against seed/t_m , and include a textbox with the maximum percent differences among membrane results, among shell results, and between membrane and shell results. *Course to fine discretization goes right to left.* Percent differences remain below 1% for all outputs except contact pressure, except for the contact pressure. In essence, the maximum contact pressure continues to rise with mesh refinement when shell elements are used, while it stabilizes when membrane elements are used. A contact singularity explains the unbounded rise of contact pressure with discretization, similar to when a concentrated load is applied to a node. Such behavior observed in Fig. 6 of Ref. [143]. Furthermore, Block and Keer state that, for a curved thin beam, "when the thickness to length ratio of the beam is small, the contact stress distribution asymptotically approaches a concentrated force at the edge of contact, as illustrated in Fig. 2" [144]. This discrepancy between the shell and the membrane is

understood to be associated with the bending stiffness present in the shells, which carries the moment that presents the singularity of the curved surface contacting the rigid line; the membrane is not able to carry such moment.

Table B-1. Convergence study results.

Element Type ^a	Seed/ t_m ^b	Number of Elements	Cavity Pressure (Pa)	Max von Mises Stress (Pa)	Max Contact Pressure (Pa)	Vertical Displacement (m)	Strain Energy (J)
MAX1	4	781	113491.31	872133248	1304485.125	-0.056497108	10.124856
MAX1	5	625	113491.57	872156800	1304529.125	-0.056498263	10.12487125
MAX1	10	312	113491.18	872069184	1304324.5	-0.056495082	10.12458038
MAX1	20	156	113485.37	871169792	1302339.375	-0.056445729	10.12218857
MAX1	30	104	113529.53	872199808	1304756.75	-0.056532491	10.12354183
MAX1	40	79	113476.75	870037760	1300163.375	-0.056404181	10.11786652
MAX1	50	63	113480.86	870892672	1300147.875	-0.05644387	10.11688709
MAX1	60	52	113559.62	873369856	1302409.75	-0.056588773	10.11824131
MAX1	70	45	113537.92	870929024	1294906.25	-0.056463934	10.11087418
MAX1	80	39	113662.08	873636544	1295792.25	-0.056646302	10.11187553
MAX1	90	35	113428.01	866804032	1279817	-0.05626459	10.09673977
MAX1	100	31	113452.94	859920512	1260951.25	-0.055938851	10.08017159
SAX1	4	781	113487.34	872567296	1403614.625	-0.056500211	10.12500286
SAX1	5	625	113487.47	872573632	1349828.875	-0.056500319	10.12498283
SAX1	10	312	113483.56	872190208	1315945.75	-0.056482423	10.12421227
SAX1	20	156	113489.57	872469632	1307339.625	-0.056498799	10.12396717
SAX1	30	104	113515.32	872924096	1306098.75	-0.056549978	10.12425232
SAX1	40	79	113444.43	868062272	1299839.375	-0.056276895	10.11378574
SAX1	50	63	113461.8	870244928	1294738.125	-0.056380849	10.11488438
SAX1	60	52	113500.68	870823168	1293880.125	-0.056404602	10.11240673
SAX1	70	45	113519.52	869356736	1299518.5	-0.056362133	10.10754108
SAX1	80	39	113639.8	871962816	1282364.5	-0.056529015	10.10802078
SAX1	90	35	113428.73	867201216	1281999.375	-0.056265291	10.09676075
SAX1	100	31	113432.37	861935168	1267678.375	-0.056035966	10.08354473

^a MAX1 and SAX1 are linear axisymmetric membrane and shell elements, respectively.

^b Refers to the distance between seeds, such that $(seed/t_m) \cdot t_m$ provides the element target size for mesh creation, where $t_m = 12.2E-6$ m (i.e., membrane thickness). A smaller seed/ t_m indicates finer discretization.

^c Vertical displacement (i.e., U2) of the node aligned with the axis of symmetry.

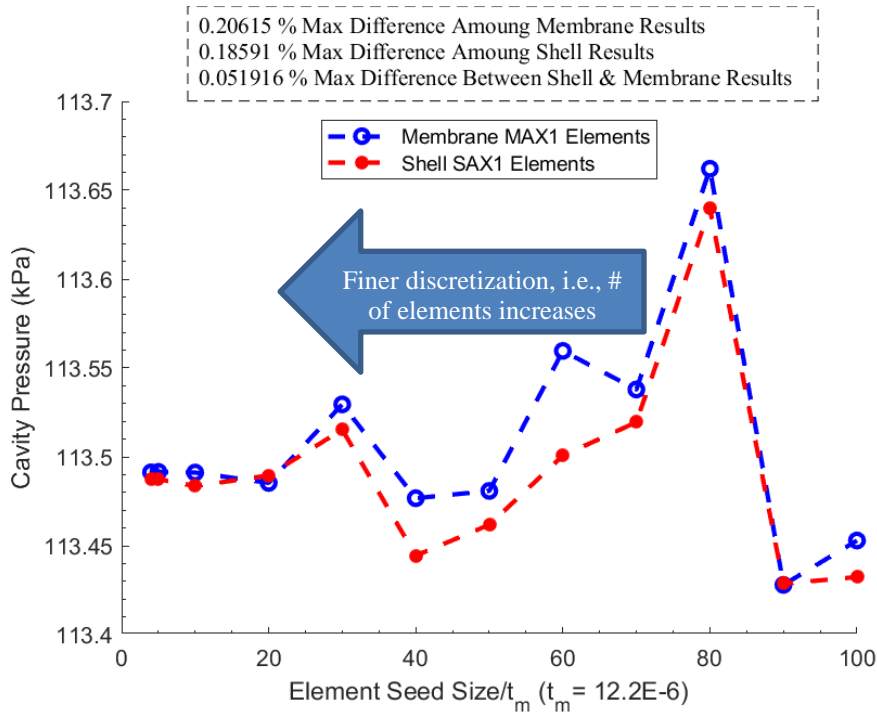


Fig. B-4. Cavity pressure versus element seed size/ t_m .

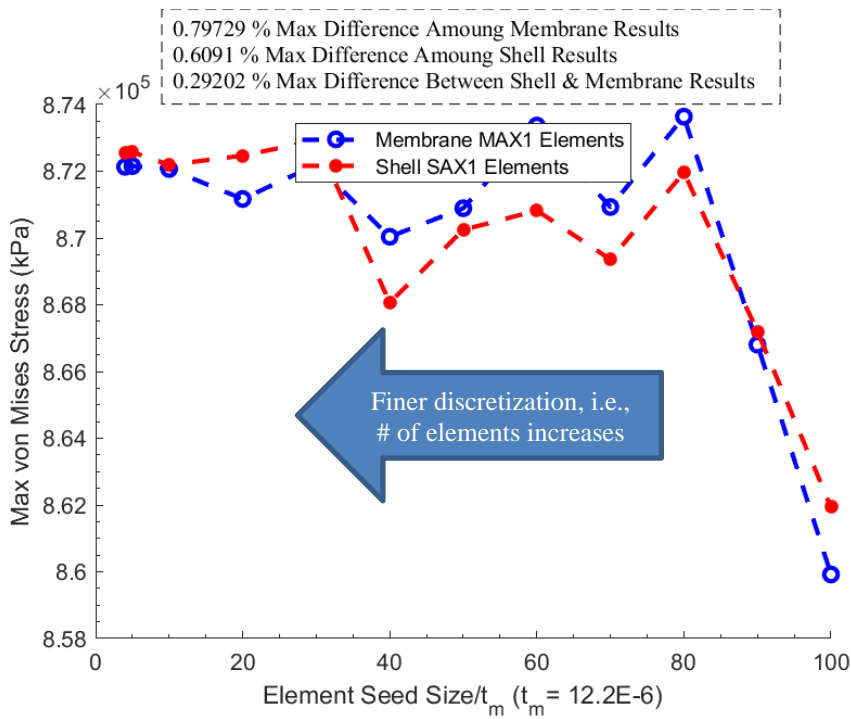


Fig. B-5. Maximum von Mises stress versus element seed size/ t_m .

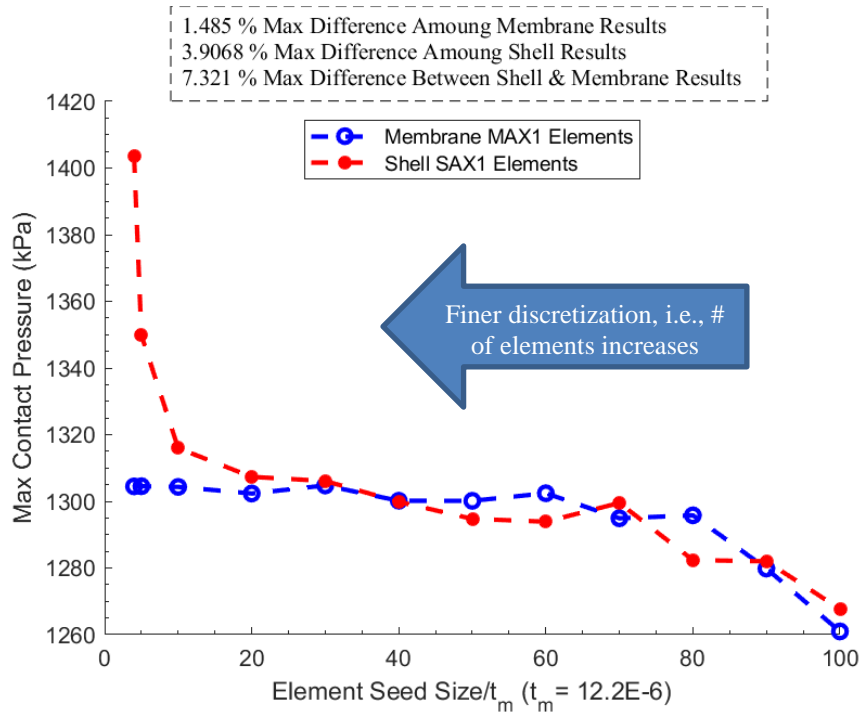


Fig. B-6. Maximum contact pressure versus element seed size/ t_m .

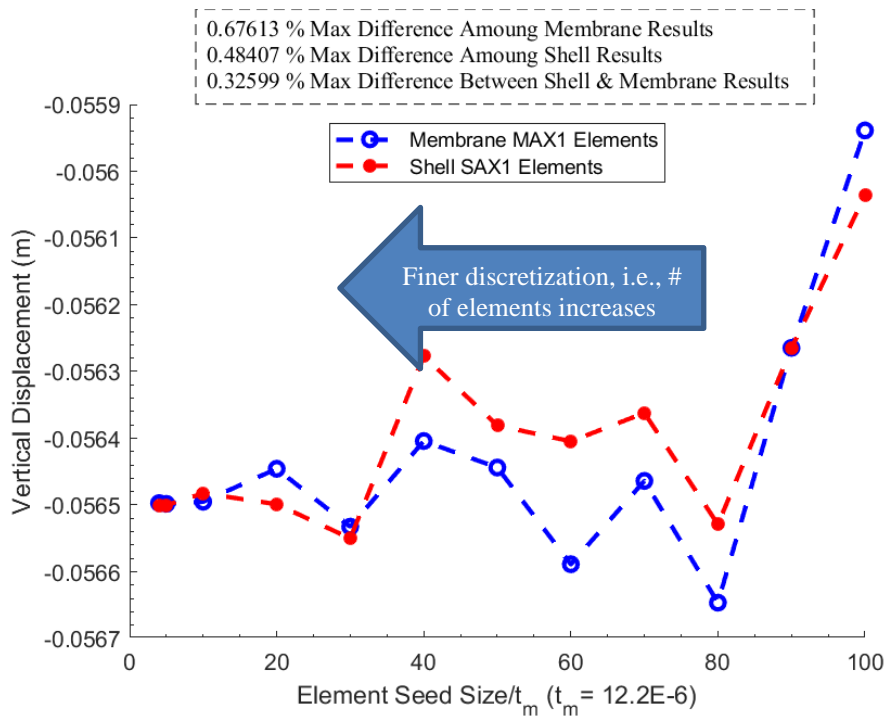


Fig. B-7. Vertical displacement (U2) at northeast node versus element seed size/ t_m .

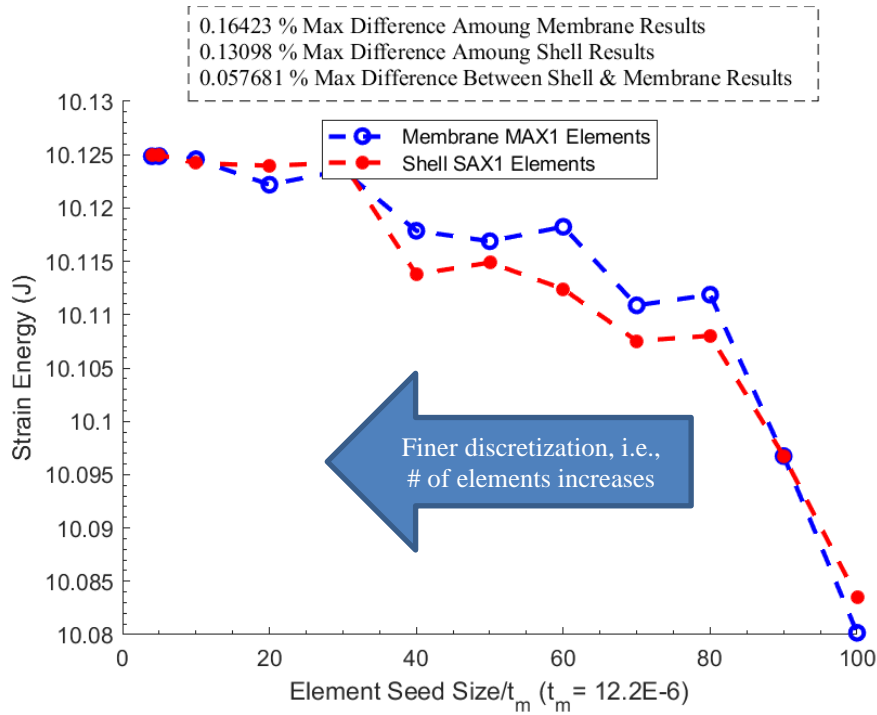


Fig. B-8. Strain energy versus element seed size/ t_m .

B.3. Frequency-Based Analysis

In terms of the modal analysis, Baker [145] provided an thorough development of the axisymmetric modes of vibration for a thin spherical shell by assuming linear deformations and neglecting bending moments and radial normal stress, and considering axisymmetric motion exclusively. His model, presented here, following Eqs. B-1 and B-2, where E is the modulus of elasticity, ν is the Poisson's ratio, ρ is the density, r is the radius, n is the mode number, c_{a_n} is related to the one of the two frequencies predicted for each n , referred to as the *upper branch*, and c_{b_n} related to the other, referred to as the *lower branch*.

$$a_n = \frac{1}{2\pi r} \left[\frac{c_{a_n} E}{2\rho(1-\nu^2)} \right]^{\frac{1}{2}} \text{ (Hz), where:} \quad (\text{B-1})$$

$$c_{a_n} = [n(n+1) + 1 + 3\nu] + \{[n(n+1) + 1 + 3\nu]^2 - 4(1-\nu^2)[n(n+1) - 2]\}^{1/2}$$

$$b_n = \frac{1}{2\pi r} \left[\frac{c_{b_n} E}{2\rho(1-\nu^2)} \right]^{\frac{1}{2}} \text{ (Hz), where:} \quad (\text{B-2})$$

$$c_{b_n} = [n(n+1) + 1 + 3\nu] - \{[n(n+1) + 1 + 3\nu]^2 - 4(1-\nu^2)[n(n+1) - 2]\}^{1/2}$$

Modes 2-5 predicted by FE analysis with shell elements are similar to those of the lower branch predicted by Eq. B-2, in frequency, and presented by [145], in shape. Fig. B-9 shows the comparison, including the mode predicted with membrane elements; the FE analysis shell frequency matches within 0.04% with b_n . The membrane mode presented is a sample of the first 100 modes showing similar characteristics, with a frequency range of 1.8079E-2 – 4,559 Hz.

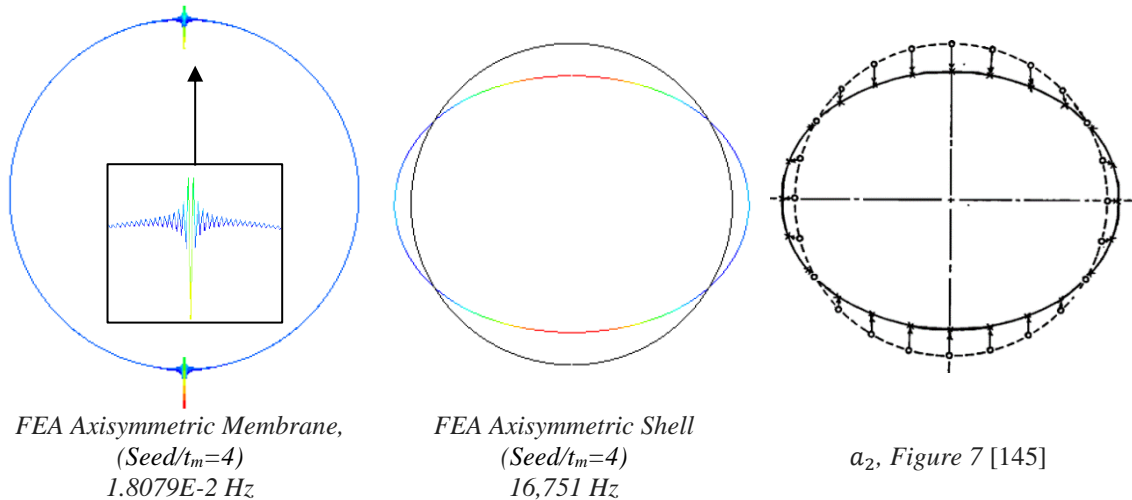


Fig. B-9. Mode 2.

While axisymmetric shell elements were not used in the solution presented in Chapter 5, they provided an estimate of the first eigenvalue while avoiding the spurious (i.e., non-physical) modes present in the eigenvalue extraction using membrane elements [133]. Said eigenvalue was taken as a lower bound for estimating the timescale that

promotes quasi-static behavior as the addition of fluid mass is expected to increase its value, reducing its period. Section 1.4.1 of Abaqus' Benchmarks Guide compares the theory to models with different shell elements within the Abaqus library, including the SAX1 used here [146].

B.4. Python Code to Generate the Abaqus CAE for the Axisymmetric Representation of the Helical Sphere

Using Abaqus v2016, the following Python code generates the Abaqus .cae file of the axisymmetric representation of the Helical Sphere for analysis, based on the input stated at the beginning of the code. The arrays “pt1” through “pt7” were built through a MATLAB function (not included) and are based on a $n = 41$ model. The author suggests that Python codes are copied and pasted into a Python, or .py, interpreter for readability, and on the Abaqus command window for model generation. The author welcomes requests and inquiries regarding functions and codes used to develop the content of this dissertation.

```

# Python code for Axisymmetric Quarter Representation of the toroidal representation, Implicit
# Used with SIMULIA Abaqus 2016, but may work with other versions
# by Ruben Adorno, Air Force Institute of Technology
# IMPORTANT: n must be odd, n = 41 in this example!!!!

#####
# Variables
#####
m_name = 'Axisym_Qtr_Tor_Sph_n=41' # Model name
job_name= m_name # Job Name, 38 chars max, no spaces
lt = 1E-0 # load time
st1 = 1E-0 # s, step time
st2 = 1E-0 # s, step time
dt1 = lt/50 # s, delta time for outputs, and max for step 1
dt2 = lt/500 # s, delta time for outputs, and max for step 2
tm = 1.22E-5 # m, T2-Tn membrane thickness
tm_T0 = 1*tm # m, T0 membrane thickness
tm_T1 = 1*tm # m, T1 membrane thickness
pl_seed_min = 2*tm # Toroids element size edge min seed

```

```

p1_seed_max = 5*tm      # Toroids element size edge min seed
p2_seed = 5E-3         # Symmetry rigid plane element size global seed
Ppress = 0.0*101325    # Pa, T1-Tend cavities pressurized gauge pressure load (before internal
pressure load)
O_freq = 2             # Output write frequency for increments, e.g. 2 = every 2 increments
E = 4.36E9             # Pa, membrane elastic modulus
rho = 1390             # kg/m3, membrane density
nu = 0.38              # Poisson's ratio
Pamb = 101325          # Pa, cavities ambient pressure
Pi = 0                 # Pa, cavities initial gauge pressure
Pvac = 101325          # Pa, max internal pressure load
p1_name = 'Toroids'    # Toroids part name
p2_name = 'Y-Sym Plane' # Y symmetry rigid plane part name
membrane_el = 1        # 0 = uses shell elements, 1 = uses membrane elements
step1 = 'Pressurize'
step2 = 'Vacuum'

```

```

#####
# Arrays defining 3-point arcs for toroid and strut circles, where n = 41
#####
pt1 =
(((0.00857541,0.00857541,0),),(0.01214697,0.31670680,0),),(0.03635500,0.31484768,0),),(0.06034968,
0.31114090,0),),(0.08399020,0.30560823,0),),(0.10713784,0.29828213,0),),(0.12965675,0.28920559,0),
),),(0.15141479,0.27843187,0),),(0.17228426,0.26602421,0),),(0.19214269,0.25205541,0),),(0.21087356
,0.23660745,0),),(0.22836694,0.21977098,0),),(0.24452017,0.20164481,0),),(0.25923846,0.18233531,0)
),),(0.27243544,0.16195579,0),),(0.28403365,0.14062585,0),),(0.29396504,0.11847066,0),),(0.3021713
3,0.09562024,0),),(0.30860436,0.07220869,0),),(0.31322637,0.04837338,0),),(0.31601025,0.02425420,
0),),)
pt2 =
(((0.01213234,0.31670736,0),),(0.03634045,0.31484936,0),),(0.06033531,0.31114369,0),),(0.08397609,
0.30561211,0),),(0.10712406,0.29828708,0),),(0.12964339,0.28921158,0),),(0.15140192,0.27843887,0),
),),(0.17227197,0.26603217,0),),(0.19213105,0.25206429,0),),(0.21086263,0.23661719,0),),(0.22835679
,0.21978153,0),),(0.24451086,0.20165611,0),),(0.25923004,0.18234728,0),),(0.27242795,0.16196838,0)
),),(0.28402715,0.14063897,0),),(0.29395957,0.11848424,0),),(0.30216692,0.09563420,0),),(0.3086010
2,0.07222294,0),),(0.31322414,0.04838785,0),),(0.31600913,0.02426879,0),),(0.31693966,0.00000732,
0),),)
pt3 =
(((0.00000000,0.30480610,0),),(0.02333268,0.30391174,0),),(0.04652843,0.30123390,0),),(0.06945114,
0.29678830,0),),(0.09196628,0.29060104,0),),(0.11394172,0.28270840,0),),(0.13524851,0.27315673,0),
),),(0.15576161,0.26200206,0),),(0.17536064,0.24930986,0),),(0.19393058,0.23515461,0),),(0.21136246
,0.21961938,0),),(0.22755398,0.20279533,0),),(0.24241012,0.18478120,0),),(0.25584370,0.16568271,0)
),),(0.26777589,0.14561191,0),),(0.27813667,0.12468662,0),),(0.28686523,0.10302961,0),),(0.2939103
5,0.08076798,0),),(0.29923068,0.05803237,0),),(0.30279502,0.03495621,0),),(0.30458243,0.01167491,
0),),)
pt4 =
(((0.00000000,0.32907321,0),),(0.02519031,0.32810764,0),),(0.05023279,0.32521661,0),),(0.07498049,
0.32041708,0),),(0.09928817,0.31373721,0),),(0.12301318,0.30521621,0),),(0.14601631,0.29490408,0),
),),(0.16816256,0.28286133,0),),(0.18932196,0.26915864,0),),(0.20937034,0.25387642,0),),(0.22819006
,0.23710435,0),),(0.24567067,0.21894086,0),),(0.26170958,0.19949254,0),),(0.27621268,0.17887352,0)
),),(0.28909485,0.15720479,0),),(0.30028049,0.13461353,0),),(0.30970398,0.11123230,0),),(0.3173099
9,0.08719831,0),),(0.32305391,0.06265261,0),),(0.32690201,0.03773924,0),),(0.32883173,0.01260440,
0),),)

```

```

pt5 =
(((0.00000000,0.30479268,0),),(0.00000000,0.28134653,0),),(0.00000000,0.25790037,0),),(0.00000000,
0.23445422,0),),(0.00000000,0.21100806,0),),(0.00000000,0.18756191,0),),(0.00000000,0.16411576,0),
),),(0.00000000,0.14066960,0),),(0.00000000,0.11722345,0),),(0.00000000,0.09377730,0),),(0.00000000
,0.07033114,0),),(0.00000000,0.04688499,0),),(0.00000000,0.02343883,0),),)
pt6 =
(((0.01171576,0.29307692,0),),(0.01171576,0.26963077,0),),(0.01171576,0.24618462,0),),(0.01171576,
0.22273846,0),),(0.01171576,0.19929231,0),),(0.01171576,0.17584615,0),),(0.01171576,0.15240000,0),
),),(0.01171576,0.12895385,0),),(0.01171576,0.10550769,0),),(0.01171576,0.08206154,0),),(0.01171576
,0.05861538,0),),(0.01171576,0.03516923,0),),(0.01171576,0.01172308,0),),)
pt7 =
(((0.00000000,0.28136117,0),),(0.00000000,0.25791501,0),),(0.00000000,0.23446886,0),),(0.00000000,
0.21102270,0),),(0.00000000,0.18757655,0),),(0.00000000,0.16413040,0),),(0.00000000,0.14068424,0),
),),(0.00000000,0.11723809,0),),(0.00000000,0.09379194,0),),(0.00000000,0.07034578,0),),(0.00000000
,0.04689963,0),),(0.00000000,0.02345347,0),),(0.00000000,0.00000732,0),),)

```

```

#####
# Model Initialization
#####

```

```

from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import os
import time
session.journalOptions.setValue(replayGeometry=COORDINATE, recoverGeometry=COORDINATE) #
Command to Output CAE Journal with Coordinates

```

```

# Creates Model
mdb.Model(modelType=STANDARD_EXPLICIT, name=m_name, absoluteZero=0,
universalGas=8.31432)

```

```

# Initialize
model = mdb.models[m_name]
assembly = model.rootAssembly
n = 2*len(pt1) - 1
n_st = 2*len(pt6)
t = range(1,n/2+1)
t = ["{:d} ".format(x) for x in t]

```

```

# Creates Material Definition
model.Material(name='Membrane')
model.materials['Membrane'].Density(table=((rho, ), ))
model.materials['Membrane'].Elastic(table=((E, nu), ))

```

```

# Creates Steps
model.ImplicitDynamicsStep(name=step1, previous='Initial',alpha=DEFAULT,amplitude=RAMP,
    application=QUASI_STATIC, convertSDI=CONVERT_SDI_ON,
    initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
    timePeriod = st1,
    maxNumInc = 500,
    initialInc = 1E-2*dt1,
    maxInc = dt1,
    minInc = 1e-15)
model.ImplicitDynamicsStep(name=step2, previous=step1,alpha=DEFAULT, amplitude=RAMP,
    application=QUASI_STATIC, convertSDI=CONVERT_SDI_ON,
    initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
    timePeriod = st2,
    maxNumInc = 5000,
    initialInc = 1E-1*dt2,
    maxInc = dt2,
    minInc = 1e-15)
model.steps[step2].control.setValues(lineSearch=(5.0, 1.0, 0.0001, 0.25, 0.1),
    allowPropagation=OFF, resetDefaultValues=OFF, discontinuous=ON)

# Creates Interactions Properties
model.ContactProperty('Contact Definitions')
model.interactionProperties['Contact Definitions'].TangentialBehavior(formulation=ROUGH)
#FRICITIONLESS)
model.interactionProperties['Contact Definitions'].NormalBehavior(allowSeparation=OFF,
    constraintEnforcementMethod=DIRECT, pressureOverclosure=HARD)
model.FluidCavityProperty(definition=PNEUMATIC, name='Pneumatic Air Cavity', useCapacity=False,
    molecularWeight=0.0289647)

# Creates Load Amplitudes
model.SmoothStepAmplitude(data=((0.0, 0.0), (1t, 1.0)), name='Smooth Cavity P', timeSpan=STEP)
model.SmoothStepAmplitude(data=((0.0, 0.0), (1t, 1.0)), name='Smooth Internal Load', timeSpan=STEP)

# Creates Sections (see section assignment)
if membrane_el == 0:
    # Shell Sections
    model.HomogeneousShellSection(idealization=NO_IDEALIZATION,
        integrationRule=SIMPSON, material='Membrane', name='Shell T2-Tn', numIntPts=5,
        poissonDefinition=DEFAULT, preIntegrate=OFF, temperature=GRADIENT,
        thickness=tm, thicknessField="", thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=OFF)
    model.HomogeneousShellSection(idealization=NO_IDEALIZATION,
        integrationRule=SIMPSON, material='Membrane', name='Shell T0', numIntPts=5,
        poissonDefinition=DEFAULT, preIntegrate=OFF, temperature=GRADIENT,
        thickness=tm_T0, thicknessField="", thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=OFF)
    model.HomogeneousShellSection(idealization=NO_IDEALIZATION,
        integrationRule=SIMPSON, material='Membrane', name='Shell T1', numIntPts=5,
        poissonDefinition=DEFAULT, preIntegrate=OFF, temperature=GRADIENT,
        thickness=tm_T1, thicknessField="", thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=OFF)
else:

```

```

# Membrane Sections
model.MembraneSection(name='Membrane T0', thickness=tm_T0, material='Membrane',
    poissonDefinition=DEFAULT, thicknessField="", thicknessType=UNIFORM)
model.MembraneSection(name='Membrane T1', thickness=tm_T1, material='Membrane',
    poissonDefinition=DEFAULT, thicknessField="", thicknessType=UNIFORM)
model.MembraneSection(name='Membrane T2-Tn', thickness=tm, material='Membrane',
    poissonDefinition=DEFAULT, thicknessField="", thicknessType=UNIFORM)

#####
# Part 1
#####
# Creates Sketch for Toroids Part 1
model.ConstrainedSketch(name=p1_name, sheetSize=2.0)
model.sketches[p1_name].sketchOptions.setValues(viewStyle=AXISYMM)
model.sketches[p1_name].ConstructionLine(point1=(0.0, -1.0), point2=(0.0, 1.0))
model.sketches[p1_name].FixedConstraint(entity=model.sketches[p1_name].geometry[2])
model.sketches[p1_name].Arc3Points(point1=(pt2[0][0][0],pt2[0][0][1]),
    point2=(pt3[0][0][0],pt3[0][0][1]),
    point3=(pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-pt3[0][0][1])/2-
pt1[0][0][1]))
model.sketches[p1_name].Arc3Points(point1=(pt2[0][0][0],pt2[0][0][1]),
    point2=(pt4[0][0][0],pt4[0][0][1]),
    point3=(pt3[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-
pt3[0][0][1])/2+pt1[0][0][1]))
for x in range(1,n/2+1):
    model.sketches[p1_name].Arc3Points(point1=(pt1[x][0][0],pt1[x][0][1]),
        point2=(pt2[x][0][0],pt2[x][0][1]),
        point3=(pt3[x][0][0],pt3[x][0][1]))
    model.sketches[p1_name].Arc3Points(point1=(pt1[x][0][0],pt1[x][0][1]),
        point2=(pt2[x][0][0],pt2[x][0][1]),
        point3=(pt4[x][0][0],pt4[x][0][1]))

# Creates Toroids Part
model.Part(dimensionality=AXISYMMETRIC, name=p1_name, type=DEFORMABLE_BODY)
part1 = mdb.models[m_name].parts[p1_name]
part1.BaseWire(sketch=model.sketches[p1_name])

# Creates Whole Part Set
Whole = part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-pt3[0][0][1])/2-
pt1[0][0][1],0),))
Whole += part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-
pt3[0][0][1])/2+pt1[0][0][1],0),))
for x in range(1,n/2+1):
    Whole += part1.edges.findAt(pt3[x])
    Whole += part1.edges.findAt(pt4[x])

part1.Set(edges = Whole, name = 'Whole')

# Creates T0 and T1 Set
part1.Set(name='Toroids T0', edges = part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0],
pt3[0][0][1]+(pt4[0][0][1]-pt3[0][0][1])/2+pt1[0][0][1],0),))+
    part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-
pt3[0][0][1])/2-pt1[0][0][1],0),))

```

```

part1.Set(name = 'Toroids T1', edges = part1.edges.findAt(pt3[1]) + part1.edges.findAt(pt4[1]))

# Creates T2-Tn Set (No T0 and T1)
Whole = part1.edges.findAt(pt3[2])
Whole += part1.edges.findAt(pt4[2])
for x in range(3,n/2+1):
    Whole += part1.edges.findAt(pt3[x])
    Whole += part1.edges.findAt(pt4[x])

part1.Set(edges = Whole, name = 'Toroids T2-Tn')

# Creates X-Sym Vertices Set
xsym = part1.vertices.findAt(pt3[0])
xsym += part1.vertices.findAt(pt4[0])
part1.Set(vertices = xsym, name = 'X-Sym')

# Creates Y-Sym Vertices Set
part1.Set(vertices = part1.vertices.findAt(pt2[n/2]), name = 'Y-Sym')

# Creates Northeast X-Sym Node Set
part1.Set(vertices = part1.vertices.findAt(pt4[0]), name = 'Northeast X-Sym Node')

# Creates Toroids N Vertices Set
vN = part1.vertices.findAt(pt2[0])
for x in range(1,n/2):
    vN += part1.vertices.findAt(pt2[x])

part1.Set(vertices = vN, name = 'Toroids N Vertices')

# Creates Toroids S Vertices Set
vS = part1.vertices.findAt(pt1[1])
for x in range(2,n/2+1):
    vS += part1.vertices.findAt(pt1[x])

part1.Set(vertices = vS, name = 'Toroids S Vertices')

# Creates Pia, Poa, and Internal Surfaces
part1.Surface(name='Toroids Pia T0', side1Edges=part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0],
pt3[0][0][1])+(pt4[0][0][1]-pt3[0][0][1])/2-pt1[0][0][1],0),))
part1.Surface(name='Toroids Poa T0', side2Edges=part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0],
pt3[0][0][1])+(pt4[0][0][1]-pt3[0][0][1])/2+pt1[0][0][1],0),))
part1.Surface(name='Toroids Internal T0', side2Edges=part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0],
pt3[0][0][1])+(pt4[0][0][1]-pt3[0][0][1])/2-pt1[0][0][1],0),),
side1Edges=part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0],
pt3[0][0][1])+(pt4[0][0][1]-pt3[0][0][1])/2+pt1[0][0][1],0),))
PiaT = part1.edges.findAt(pt3[1])
PoaT = part1.edges.findAt(pt4[1])
part1.Surface(name='Toroids Pia T1', side2Edges=part1.edges.findAt(pt3[1]))
part1.Surface(name='Toroids Poa T1', side1Edges=part1.edges.findAt(pt4[1]))
part1.Surface(name='Toroids Internal T1', side1Edges=part1.edges.findAt(pt3[1]),
side2Edges=part1.edges.findAt(pt4[1]))
for x in range(2,n/2+1):
    PiaT += part1.edges.findAt(pt3[x])

```

```

PoaT += part1.edges.findAt(pt4[x])
part1.Surface(name='Toroids Pia T' + t[x-1], side2Edges=part1.edges.findAt(pt3[x]))
part1.Surface(name='Toroids Poa T' + t[x-1], side1Edges=part1.edges.findAt(pt4[x]))
part1.Surface(name='Toroids Internal T' + t[x-1], side1Edges=part1.edges.findAt(pt3[x]),
              side2Edges=part1.edges.findAt(pt4[x]))

part1.Surface(name='Poa', side1Edges=PoaT,
              side2Edges=part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-
pt3[0][0][1])/2+pt1[0][0][1],0),)))
part1.Surface(name='Pia', side2Edges=PiaT,
              side1Edges=part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-
pt3[0][0][1])/2-pt1[0][0][1],0),)))

# Creates Additional Set to Flip Normal on Elements
part1.Set(name='Poa', edges = PoaT + part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0],
pt3[0][0][1]+(pt4[0][0][1]-pt3[0][0][1])/2+pt1[0][0][1],0),)))

# Assigns Section
if membrane_el == 0:
    # Shell
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
                           region = part1.sets['Toroids T2-Tn'], sectionName = 'Shell T2-Tn',
                           thicknessAssignment=FROM_SECTION)
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
                           region = part1.sets['Toroids T0'], sectionName = 'Shell T0',
                           thicknessAssignment=FROM_SECTION)
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
                           region = part1.sets['Toroids T1'], sectionName = 'Shell T1',
                           thicknessAssignment=FROM_SECTION)
else:
    # Membrane
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
                           region = part1.sets['Toroids T2-Tn'], sectionName = 'Membrane T2-Tn',
                           thicknessAssignment=FROM_SECTION)
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
                           region = part1.sets['Toroids T0'], sectionName = 'Membrane T0',
                           thicknessAssignment=FROM_SECTION)
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
                           region = part1.sets['Toroids T1'], sectionName = 'Membrane T1',
                           thicknessAssignment=FROM_SECTION)

# Mesh
if membrane_el == 0:
    part1.setElementType(elemTypes=(ElemType(elemCode=SAX1, elemLibrary=EXPLICIT,
secondOrderAccuracy=OFF), ),
                        regions = part1.sets['Whole'])
else:
    part1.setElementType(elemTypes=(ElemType(elemCode=MAX1, elemLibrary=STANDARD), ),
                        regions = part1.sets['Whole'])

Whole = part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-pt3[0][0][1])/2-
pt1[0][0][1],0),))

```

```

Whole += part1.edges.findAt(((pt4[0][0][0] + pt1[0][0][0], pt3[0][0][1]+(pt4[0][0][1]-
pt3[0][0][1])/2+pt1[0][0][1],0),))
for x in range(1,n/2+1):
    Whole += part1.edges.findAt(pt3[x])
    Whole += part1.edges.findAt(pt4[x])

part1.seedEdgeByBias(biasMethod=DOUBLE, constraint=FINER,
    minSize = p1_seed_min,
    maxSize = p1_seed_max,
    endEdges = Whole)
##part1.seedPart(deviationFactor=0.1, minSizeFactor=0.1, size=p1_seed)
part1.generateMesh()
part1.flipNormal(regions = part1.sets['Poa'])
part1.flipNormal(regions = part1.sets['Toroids T0'])

#####
# Part 2
#####
# Creates Sketch
model.ConstrainedSketch(name=p2_name, sheetSize=2.0)
model.sketches[p2_name].sketchOptions.setValues(viewStyle=AXISYM)
model.sketches[p2_name].ConstructionLine(point1=(0.0, -1.0), point2=(0.0, 1.0))
model.sketches[p2_name].FixedConstraint(entity=model.sketches[p1_name].geometry[2])
model.sketches[p2_name].Line(point1=(0.0, 0.0), point2=(1.2*pt2[n/2][0][0], 0.0))

# Creates Part and Sets
model.Part(dimensionality=AXISYMMETRIC, name=p2_name, type=DISCRETE_RIGID_SURFACE)
part2 = model.parts[p2_name]
part2.BaseWire(sketch=model.sketches[p2_name])
part2.Set(name='Whole', edges=part2.edges.findAt(((0, 0, 0),)))
part2.Surface(name='Contact', side1Edges=part2.edges.findAt(((0, 0, 0), )))

# Mesh
part2.seedPart(deviationFactor=0.1, minSizeFactor=0.1, size=p2_seed)
part2.setElementType(elemTypes=(ElemType(elemCode=RAX2, elemLibrary=EXPLICIT), ), regions =
part2.sets['Whole'])
part2.generateMesh()

#####
# Assembly
#####
# Creates Assembly
assembly.DatumCsysByThreePoints(coordSysType=
    CYLINDRICAL, origin=(0.0, 0.0, 0.0), point1=(1.0, 0.0, 0.0), point2=(0.0, 0.0, -1.0))
assembly.Instance(dependent=ON, name=p1_name, part=part1)
assembly.Instance(dependent=ON, name=p2_name, part=part2)

# X-Sym
vS = assembly.allInstances[p1_name].vertices.findAt(pt3[0])
assembly.Set(vertices = vS, name = 'T0 S X-Sym Vertex')
assembly.SetByBoolean(name='X-Sym', sets=(assembly.allInstances[p1_name].sets['Northeast X-Sym
Node'],
assembly.sets['T0 S X-Sym Vertex']))

```

```

# Creates Reference Points (RP)
RP = assembly.ReferencePoint(point=(0,pt2[0][0][1],0))
assembly.features.changeKey(fromName='RP-1', toName='T0')
assembly.Set(name='T0', referencePoints=(assembly.referencePoints[RP.id],))
for x in range(1,n/2+1):
    RP = assembly.ReferencePoint(point=(0,pt2[x][0][1],0))
    assembly.features.changeKey(fromName='RP-1', toName='T'+t[x-1])
    assembly.Set(name='T'+t[x-1], referencePoints=(assembly.referencePoints[RP.id],))

RP = assembly.ReferencePoint(point=(0, 0, 0))
assembly.features.changeKey(fromName='RP-1', toName='Rigid RP')
assembly.Set(name='Rigid RP', referencePoints=(assembly.referencePoints[RP.id], ))

# Creates Outputs
model.FieldOutputRequest(name = p1_name + ' Whole', frequency = O_freq, createStepName=step1,
    region = assembly.allInstances[p1_name].sets['Whole'],
    variables = ('S','LE','U','V','A','RF','P','CSTRESS','CDISP','COORD'))
model.FieldOutputRequest(name = 'Poa S Coords', frequency = O_freq, createStepName=step1,
    region = assembly.allInstances[p1_name].sets['Poa'],
    variables = ('COORD',))
model.HistoryOutputRequest(name = 'Whole', frequency = O_freq, createStepName=step1,variables =
PRESELECT)
model.HistoryOutputRequest(name = p1_name + ' Whole', frequency = O_freq, createStepName=step1,
    region = assembly.allInstances[p1_name].sets['Whole'],
    variables = PRESELECT)
model.HistoryOutputRequest(name = 'Northeast X-Sym Node', frequency = O_freq, createStepName=step1,
rebar=EXCLUDE, sectionPoints=DEFAULT,
    region = assembly.allInstances[p1_name].sets['Northeast X-Sym Node'],
    variables =('U2', 'UR3'))
model.HistoryOutputRequest(name = 'Rigid RP', frequency = O_freq, createStepName=step1,
rebar=EXCLUDE, sectionPoints=DEFAULT,
    region = assembly.sets['Rigid RP'],
    variables =('RF1', 'RF2', 'RM3'))
del model.fieldOutputRequests['F-Output-1']
del model.historyOutputRequests['H-Output-1']

#####
# Interactions
#####
# Creates Fluid Cavity for Each Tube/Circle w Initial P and T, and History Request
model.FluidCavity(createStepName='Initial', ambientPressure=Pamb, useAdiabatic=False,
    interactionProperty='Pneumatic Air Cavity',
    cavityPoint = assembly.sets['T0'],
    cavitySurface = assembly.instances[p1_name].surfaces['Toroids Internal T0'],
    name = 'T0 Cavity')
model.FluidCavityPressure(fluidPressure=Pi,
    name = 'T0 Cavity Pi',
    fluidCavity = 'T0 Cavity')
model.Temperature(createStepName='Initial',
crossSectionDistribution=CONSTANT_THROUGH_THICKNESS,distributionType=UNIFORM,
magnitudes=(288.15, ),
    name = 'T0 Cavity Ti',

```

```

    region = assembly.sets['T0'])
model.HistoryOutputRequest(createStepName=step1,rebar=EXCLUDE,sectionPoints=DEFAULT,
    variables  =(PCAV',CVOL'),
    name      = 'T0 Cavity',
    region    = assembly.sets['T0'],
    frequency = O_freq)

for x in range(0,n/2):
    model.FluidCavity(createStepName='Initial', ambientPressure=Pamb, useAdiabatic=False,
        interactionProperty='Pneumatic Air Cavity',
        cavityPoint  = assembly.sets['T' + t[x]],
        cavitySurface = assembly.instances[p1_name].surfaces['Toroids Internal T' + t[x]],
        name         = 'T' + t[x] + ' Cavity')
    model.FluidCavityPressure(fluidPressure=Pi,
        name      = 'T' + t[x] + ' Cavity Pi',
        fluidCavity = 'T' + t[x] + ' Cavity')
    model.Temperature(createStepName='Initial',
crossSectionDistribution=CONSTANT_THROUGH_THICKNESS,distributionType=UNIFORM,
magnitudes=(288.15, ),
        name  ='T' + t[x] + ' Cavity Ti',
        region = assembly.sets['T' + t[x]])
    model.HistoryOutputRequest(createStepName=step1, rebar=EXCLUDE,sectionPoints=DEFAULT,
        variables  =(PCAV',CVOL'),
        name      = 'T' + t[x] + ' Cavity',
        region    = assembly.sets['T' + t[x]],
        frequency = O_freq)

# General Contact
cont_name = 'General Self Contact'
model.ContactStd(createStepName='Initial', name=cont_name)
model.interactions[cont_name].includedPairs.setValuesInStep(stepName='Initial', useAllstar=ON)
model.interactions[cont_name].contactPropertyAssignments.appendInStep(
    assignments=((GLOBAL, SELF, 'Contact Definitions'), ), stepName='Initial')
model.interactions[cont_name].surfaceThicknessAssignments.appendInStep(stepName='Initial',
    assignments=((GLOBAL, ORIGINAL, 1.0),
        (assembly.instances[p1_name].surfaces['Toroids Pia T0'], ORIGINAL, tm/tm_T0),
        (assembly.instances[p1_name].surfaces['Toroids Poa T0'], ORIGINAL, tm/tm_T0),
        (assembly.instances[p1_name].surfaces['Toroids Pia T1'], ORIGINAL, tm/tm_T1),
        (assembly.instances[p1_name].surfaces['Toroids Poa T1'], ORIGINAL, tm/tm_T1)))

#####
# Constraints
#####
# Toroids Tie
model.Tie(name='T-T Tie', adjust=OFF, positionToleranceMethod=COMPUTED, thickness=ON,
tieRotations=OFF,
    master = assembly.instances[p1_name].sets['Toroids N Vertices'],
    slave  = assembly.instances[p1_name].sets['Toroids S Vertices'])

# Rigid Constraint for Y-Sym Plane
model.RigidBody(name='Y-Sym Rigid Plane',
    bodyRegion  = assembly.instances[p2_name].sets['Whole'],
    refPointRegion = assembly.sets['Rigid RP'])

```

```

#####
# Loads and Boundary Conditions (BC)
#####
# BCs
model.DisplacementBC(name='X-Sym',amplitude=UNSET, u1=0, u2=UNSET, ur3=UNSET,
createStepName=step1,
    distributionType=UNIFORM, fieldName="", fixed=OFF, localCsys=None,
    region = assembly.sets['X-Sym'])
model.DisplacementBC(name='Y-Sym',amplitude=UNSET, u1=UNSET, u2=0, ur3=UNSET,
createStepName=step1,
    distributionType=UNIFORM, fieldName="", fixed=OFF, localCsys=None,
    region = assembly.allInstances[p1_name].sets['Y-Sym'])
model.EncastreBC(name='Rigid RP Encastre', createStepName=step1, localCsys=None,
    region = assembly.sets['Rigid RP'])

# Loads
model.FluidCavityPressureBC(fixed=OFF, amplitude='Smooth Cavity P', createStepName=step1,
    magnitude = Ppress,
    fluidCavity = 'T0 Cavity',
    name = 'T0 Cavity P')
model.boundaryConditions['T0 Cavity P'].deactivate(step2)
for x in range(0,n/2):
    model.FluidCavityPressureBC(fixed=OFF, amplitude='Smooth Cavity P', createStepName=step1,
        magnitude = Ppress,
        fluidCavity = 'T' + t[x] + ' Cavity',
        name = 'T' + t[x] + ' Cavity P')
    model.boundaryConditions['T' + t[x] + ' Cavity P'].deactivate(step2)

model.Pressure(name='Vacuum', amplitude='Smooth Internal Load', magnitude=-Pvac,
createStepName=step2,
    distributionType=UNIFORM, field="",region = assembly.allInstances[p1_name].surfaces['Pia'])

#####
# Job
#####
# Creates Job
mdb.Job(name=job_name, model=m_name, numCpus=1, numDomains=1, numGPUs=0,
    nodalOutputPrecision=FULL, type = ANALYSIS,
    memory = 90, memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    contactPrint=OFF, echoPrint=OFF, historyPrint=OFF, modelPrint=OFF,
    explicitPrecision=SINGLE,
    resultsFormat = ODB,
    multiprocessingMode=DEFAULT,
    queue=None, atTime=None, waitHours=0, waitMinutes=0,
    scratch="",userSubroutine="", description=")

```

APPENDIX C. ICORON AIR-STIFFENED DESIGN SUPPORTING INFORMATION

C.1. Mesh Density and Timescale Selection

The mesh density and timescale (i.e., load rate) selections made to characterize the structural response of the UC (section 6.3) were based on the energy distribution across the load history given 12 combinations of three solution timescales (i.e., 0.05 s, 0.10 s and 0.25 s) and four mesh densities (i.e., element side lengths varying from $0.01r_c$ to $(x)r_c$, where x equal 0.1, 0.075, 0.05 or 0.025, and r_c is the UC membrane radius). Mesh densities favors smaller elements (i.e., $0.01r_c$ in element side length) along the corners where membrane surfaces meet and larger elements towards the center of each membrane surface forming the UC (refer to Fig. VI-14 on page 158). The intent of this study was two-fold: {1} evaluate the effect mesh density has on energy quantities; and {2} establish a minimal timescale that allows for quasi-static solutions (i.e., keep kinetic, contact enforcement and bulk viscosity energies combined below 10% of the work done by the applied load).

Fig. C-1 shows the internal energy as a fraction of the work done by the applied load for each of the 12 combinations; this entailed running and extracting results from 12 models using the definitions detailed in section 6.3.1, with changes to the timescale and mesh density, as appropriate. Overall, undesirable energy quantities (i.e., kinetic, contact enforcement and bulk viscosity) combined remained below 10% across approximately 93% of the load history for mesh densities $0.01r_c$ to $0.050r_c$ and $0.01r_c$ to $0.025r_c$. In other words, the internal energy to work done ratios remained above 90% for these two mesh densities. Furthermore, the slope variations shown in the energy ratios associated with

course meshes (e.g., $0.01r_c$ to $0.1r_c$) dissipate as the mesh is refined. This is attributed to improvement in contact enforcement with finer meshes as relatively gradual contact interactions can take place.

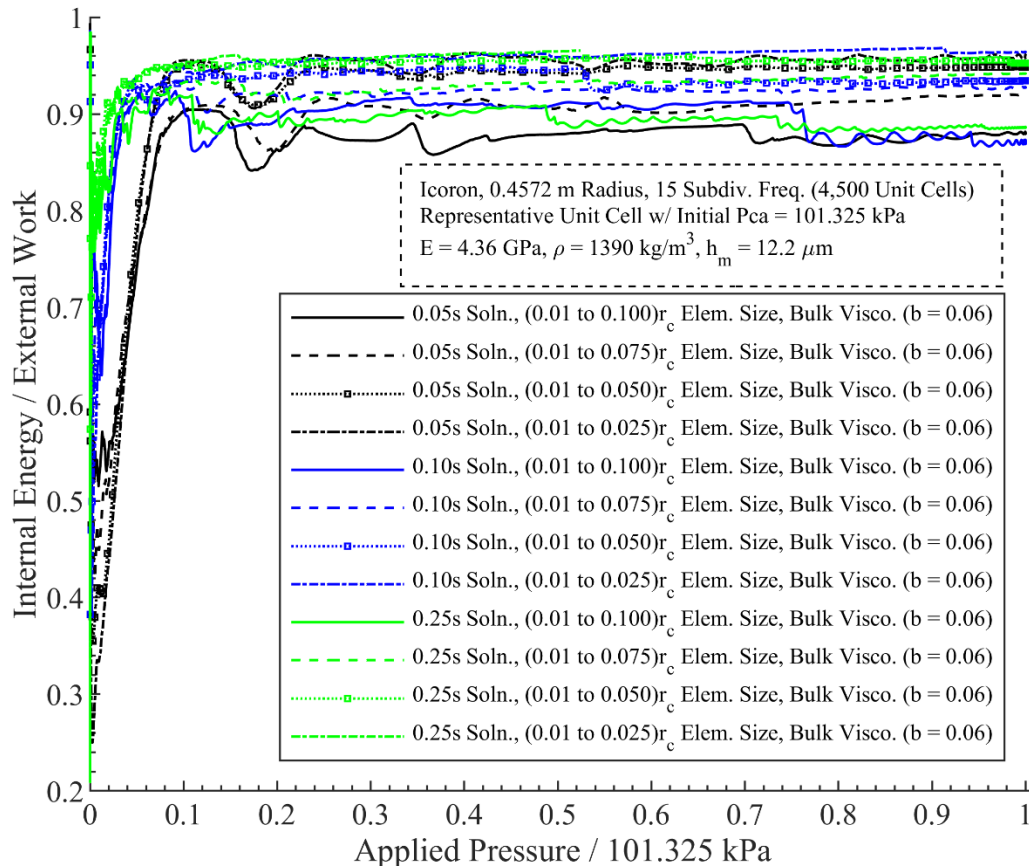


Fig. C-1. Internal energy as a fraction of the work done.

The solution computation time also played a role in the selection of a timescale and mesh density. Fig. C-2 shows the computation time across these combinations in a heatmap format, where darker tones of blue indicate lower computation times. These were calculated from the time the model was submitted to the time the solver wrote the last output into the database. Two to four analyses were ran simultaneously, each with two cores in a workstation with Intel Xeon E5-2685W v3 CPU (3.1 GHz and 10 physical cores) and 16 GB of memory.

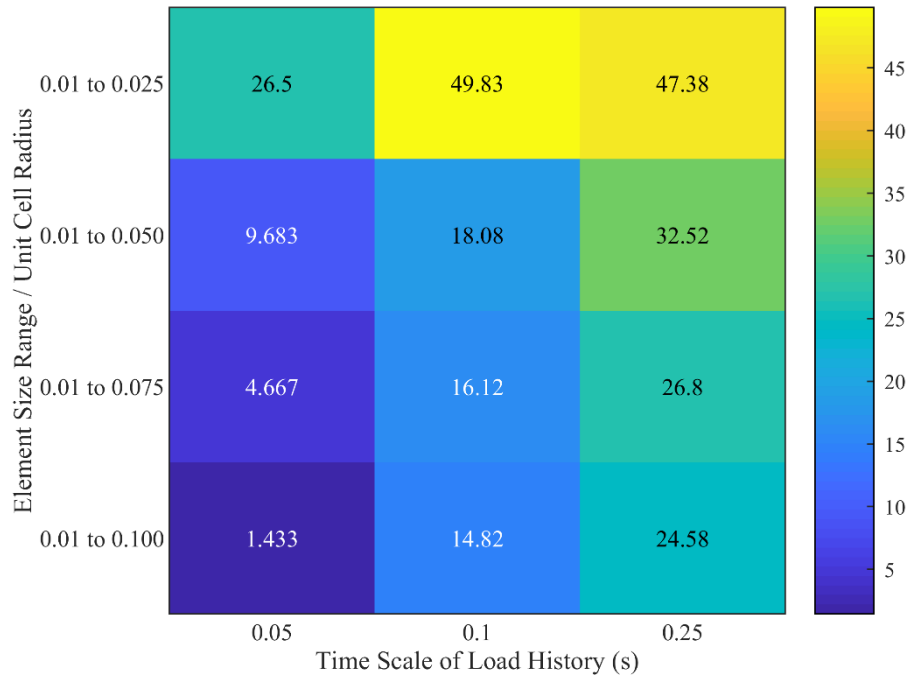


Fig. C-2. Solution computation times (hours) across mesh density and timescale.

An analysis using mesh density of $0.01r_c$ to $0.050r_c$ and a solution timescale of 0.05 s took approximately 9.7 hours to finish. This combination was selected given that it provided internal-to-work energy ratios above 90% across most of the solution while providing a solution within 12 hours. This allows a two-step loading history analysis (i.e., the UC cavity is inflated to a designated preload pressure prior to applying the internal pressure load resembling air removal) to be completed within 24 hours; a valuable consideration when further response surface and optimization studies are pursued. Other outputs considered were the stress distribution, UC cavity pressure and radial displacement. The maximum and minimum stress von Mises stresses show dependence on the mesh density, with minimal (qualitative) variation on the distribution when comparing the solutions associated with mesh densities $0.01r_c$ to $0.050r_c$ and $0.01r_c$ to $0.025r_c$. The UC cavity pressure and the radial displacement showed indistinctive variations.

Such mesh and timescale dependencies are predominant when considering local descriptions, such as the stress field and the development of the wrinkling patterns rather than global descriptions such as energy quantities and overall UC displacement. Fig. C-3 compares the von Mises stress contours at the end of the load amplitude (i.e., referred to as state E in Chapter 6) across 0.05 s and 0.01 s timescales and mesh densities $0.01r_c$ to $0.050r_c$ and $0.01r_c$ to $0.025r_c$. Comparing Figs. C-3a and C-3b, or C-3c and C-3d, a longer timescale distributes the volume reduction forced by the rigid cone by easing the wrinkles along the corners and prompting wrinkling along surfaces previously parallel to and in contact with the rigid cone. This behavior is dominant with increased mesh density, where a higher number of degrees of freedom allows new wrinkles to develop, changing the resulting pattern of the wrinkled UC.

Overall, the selected mesh density of $0.01r_c$ to $0.050r_c$ and a solution timescale of 0.05 s is said to result in a quasi-static response with converged energy and global radial displacement quantities. On the contrary, such mesh density and timescale is not representative of a converged geometrical state and stress/strain fields. As such, detailed wrinkle characterization of the solution in question requires a further refined mesh and a lower load rate in order to allow time-independent wrinkle development.

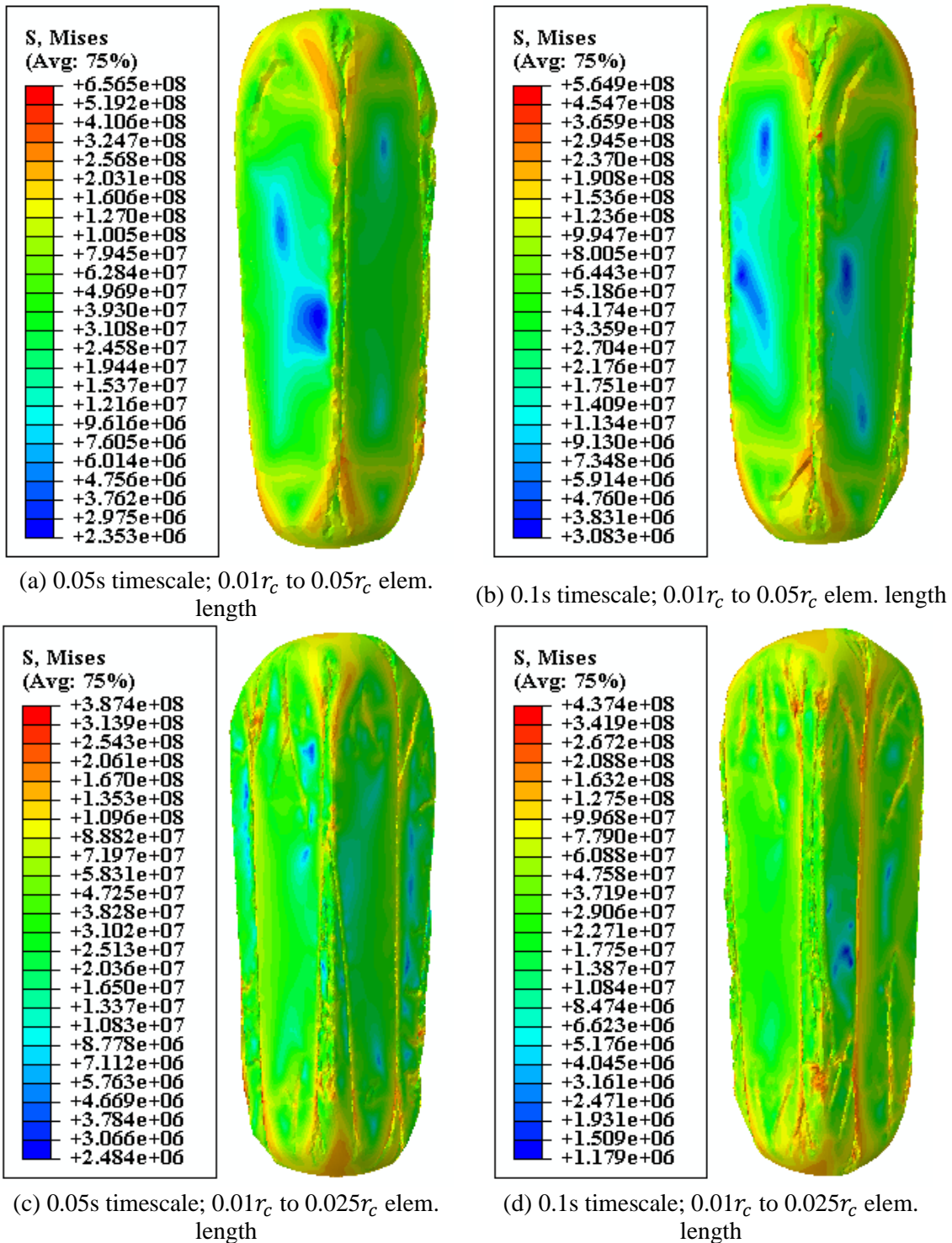


Fig. C-3. Comparison of von Mises stress contours across two mesh densities and two timescales at loading state E.

C.2. Python Code to Generate the Abaqus CAE for Unit Cell Assembly

Representation of the Icoron

Using Abaqus v2016, the following Python code generates the Abaqus .cae file of the unit cell assembly of the Icoron for analysis, based on the input stated at the beginning of the code. The unit cell radius, r_c , or “rc” in the code below, follows the maximum ratio shown in Fig. VI-5 for subdivision frequency of 15 and a radius, r_s , or “rs” in the code below, of 0.4572 m. A MATLAB function based on Mathematica Geodesate vertex location data generated r_c as a function of subdivision frequency and r_s . The author suggests that Python codes are copied and pasted into a Python, or .py, interpreter for readability, and on the Abaqus command window for model generation. The author welcomes requests and inquiries regarding functions and codes used to develop the content of this dissertation.

```
# Python code for Abaqus CAE of Air-Stiffened Geodesic Polyhedron Unit Cell (UC),
# Implicit or Explicit, Hydrostatic Surface-Based Fluid Cavity
# SIMULIA Abaqus 2016
# by Ruben Adorno, Air Force Institute of Technology
# Created: 08 June 2020
# Last revised: 16 Feb 2021

#####
#####
# Variables / Inputs
#####
#####
# Driven by Icoron subdivision frequency and size
rs = 0.45720000 # m, circumscribed sphere radius
rc = 0.02322453 # m, unit cell max internal radius

# Paths
wpath = 'C:\\Output' # Path of output files, e.g. .odb, .dat
mpath = 'C:\\' # Path of .cae and .jnl files

# Names
cae_name = 'CAE_Name' # CAE file name
```

```

m_name      = 'Model_Name' # Model name
job_name    = 'Job_Name'   # Job name, 38 chars max, no spaces

# Unit Cell
p1_name     = 'UC'        # part name
memb_shell  = 0           # 0 = uses membrane elements, 1 = uses shell elements
E           = 4.36E9      # Pa, membrane elastic modulus
hm          = 0.0001220   # membrane thickness, also preloaded in the loaded file
rho         = 1390        # kg/m3, membrane density
nu          = 0.38        # Poisson's ratio
fillet      = 0           # Fillet Corners (1)/No Fillets (0) for unit cell and contact cone
r_fillet    = 0.01*rc     # Fillet radius for unit cell and contact cone
chamfer     = 1           # Chamfer Corners (1)/No Fillets (0) for unit cell and contact cone
l_chamfer   = 0.01*rc     # Chamfer radius for unit cell and contact cone
Contact_or_BC = 0         # Contact Cone(0) or BCs(1) replacement
contact_cone_top_hat = 1   # of Contact cone is on, put a triangular surface at top to limit radial travel of
UC during loading
BCthroughRP = 0           # Do not (0)/Apply(1) BC through RP using Coupling Constraint
                                # iff Contact_or_BC == 1, unless chamfer == 0 and fillet == 0
p1_seed     = (0.01*rc,50E-3*rc) # element size min, max bias seed

# Air Properties
Ru          = 8.31432     # J/(mol-K), Universal Gas Constant
T0          = 0           # K, Absolute Zero Temperature
molW        = 0.0289647   # kg/mol, molar mass
Cp_const    = (28.11,     # a, J/(mol-K)   Cp_bar = a + b(T-T0) + c(T-T0)^2
              0.001967,   # b, J/(mol-K^2)   + d(T-T0)^3 + e/(T-T0)^2
              4.802e-06,  # c, J/(mol-K^3)
              -1.966e-09, # d, J/(mol-K^4) Cp = Cp_bar/molW = Cv - R
              0.0)        # e, (J-K)/mol

# Contact Cone
p2_name     = 'Contact Cone' # Y symmetry rigid plane part name
p2_seed     = p1_seed[1]    # Symmetry rigid plane element size global seed

# Step Info
step_type   = 'NLED'       # 'NLIS = Nonlinear implicit static, 'NLID' = NL implicit dynamic, 'NLED' =
NL explicit dynamic
int_operator = 'QS'        # Only Applies to NLID: QS = QUASI_STATIC, MD =
MODERATE DISSIPATION, TF = TRANSIENT_FIDELITY
t           = 5E-2         # s, analysis time per step
pre_pressurize = 0         # 0 = vacuum from initial cavity pressure, 1 = pre-pressurize to Ppress
adiabatic   = 'OFF'       # 'OFF'/'ON', Explicit only, changes cavity temp based on adiabatic
linearBulkVisco = 0.06    # linear bulk viscosity, explicit only, 0.06 default
quadBulkVisco = 1.2       # quadratic bulk viscosity, explicit only, 1.2 default

vacc_t      = t           # s, load time in step
vacc_amp    = 'Smooth'    # 'Linear' or 'Smooth'
vacc_step   = (t, 3e-16, 3E-2, vacc_t/5E3, 10000) # Vacuum Step Details
                                # (step time, min inc, max inc, initial inc, max # of inc)

preP_t      = t           # s, load time in step
preP_amp    = 'Smooth'    # 'Linear' or 'Smooth'

```

```

preP_step    = (t, 3e-16, 3E-2, preP_t/5E3, 10000) # Pre-pressurize Step Details
              # (step time, min inc, max inc, initial inc, max # of inc)

O_freq      = 1      # Implicit ONLY, Output write frequency for increments, e.g., 2 = every 2 increments
numInt      = 1000   # Explicit ONLY, Output number of evenly spaced time intervals

# Load Info
Pamb        = 101325 # Pa, cavities ambient pressure
Pi          = 0      # Pa, cavities initial gauge pressure
Pvac        = 101325 # Pa, max internal pressure load
Ppress      = 101325 # Pc # Pa, T1-Tend cavities pressurized gauge pressure load (before internal
pressure load)

# Job Parallelization
parallel    = 'ON'   # Parallelization 'ON' or 'OFF'
numCPUs     = 2      # Number of CPUs
numGPUs     = 0      # Number of GPUs

#####
#####
# Model Initialization
#####
#####
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
import os
import time
import math

session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE) #
Command to Output CAE Journal with Coordinates
os.chdir(wpath) # assigns working directory, 'os.getcwd()' command shows it
mdb.saveAs(pathName=mpath + '\\' + cae_name + '.cae')

# Creates Model
mdb.Model(modelType=STANDARD_EXPLICIT, name=m_name, absoluteZero = T0, universalGas = Ru)

# Initialize
model = mdb.models[m_name]
assembly = model.rootAssembly

# Creates Material Definition

```

```

model.Material(name='Membrane')
model.materials['Membrane'].Density(table=((rho, ), ))
model.materials['Membrane'].Elastic(table=((E, nu), ))

# Creates Interactions Properties
model.ContactProperty('Contact Definitions')
model.interactionProperties['Contact Definitions'].TangentialBehavior(formulation=FRICITIONLESS)
model.StdInitialization(name = 'Contact Initialization')
model.FluidCavityProperty(definition=PNEUMATIC, name='Pneumatic Air Cavity', useCapacity=False,
    molecularWeight = molW)
if step_type == 'NLED':
    model.interactionProperties['Pneumatic Air Cavity'].setValues(useCapacity=True,
        capacityTable=(Cp_const, ),)
    model.interactionProperties['Contact Definitions'].NormalBehavior(allowSeparation=ON,
        constraintEnforcementMethod=DEFAULT, pressureOverclosure=HARD)
else:
    model.interactionProperties['Contact Definitions'].NormalBehavior(allowSeparation=ON,
        constraintEnforcementMethod=DIRECT, pressureOverclosure=HARD)
    model.StdStabilization(name = 'Contact Stabilization')

# Creates Load Amplitudes
if vacc_amp == 'Smooth':
    model.SmoothStepAmplitude(data=((0.0, 0.0), (vacc_t, 1.0)), name = 'Internal Load', timeSpan=STEP)
elif vacc_amp == 'Linear':
    model.TabularAmplitude(data=((0.0, 0.0), (vacc_t, 1.0)), name = 'Internal Load',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)

if preP_amp == 'Smooth':
    model.SmoothStepAmplitude(data=((0.0, 0.0), (preP_t, 1.0)), name = 'Cavity P', timeSpan=STEP)
elif preP_amp == 'Linear':
    model.TabularAmplitude(data=((0.0, 0.0), (preP_t, 1.0)), name= 'Cavity P',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)

# Creates Sections (see section assignment)
if memb_shell == 1:
    # Shell Sections
    model.HomogeneousShellSection(idealization=NO_IDEALIZATION,
        integrationRule=SIMPSON, material='Membrane', name='Shell', numIntPts=5,
        poissonDefinition=DEFAULT, preIntegrate=OFF, temperature=GRADIENT,
        thickness=hm, thicknessField="", thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=OFF)
else:
    # Membrane Sections
    model.MembraneSection(name='Membrane', thickness=hm, material='Membrane',
        poissonDefinition=DEFAULT, thicknessField="", thicknessType=UNIFORM)

#####
#####
# Part 1
#####
#####

# Creates Sketch

```

```

model.ConstrainedSketch(name=p1_name, sheetSize=2.0)
model.sketches[p1_name].sketchOptions.setValues(viewStyle=AXISYM)
model.sketches[p1_name].ConstructionLine(point1=(-1, 0), point2=(1, 0))
model.sketches[p1_name].ArcByCenterEnds(center=(0, 0), direction=CLOCKWISE,
    point1=(-rc-hm/2, 0), point2=(rc+hm/2, 0))
model.sketches[p1_name].Line(point1=(-rc-hm/2, 0), point2=(rc+hm/2, 0))

# Creates Part
model.Part(dimensionality=THREE_D, name=p1_name, type=DEFORMABLE_BODY)
part1 = mdb.models[m_name].parts[p1_name]
part1.BaseSolidRevolve(angle=360.0, flipRevolveDirection=OFF,
    sketch= model.sketches[p1_name])

# Reference Point for Cavity
RP = part1.ReferencePoint(point=(0,0,0))
part1.features.changeKey(fromName='RP', toName='UC')
part1.Set(name='UC RP', referencePoints=(part1.referencePoints[RP.id],))

# Create Datum Planes for Partitions, and System for Assembly
dpXY = part1.DatumPlaneByPrincipalPlane(offset=0.0, principalPlane=XYPLANE)
dp1 = part1.DatumPlaneByThreePoints(
    point1 = (0, 0, -rs),
    point2 = (rc*cos(0), rc*sin(0), 0),
    point3 = (rc*cos(radians(120)), rc*sin(radians(120)), 0))
dp2= part1.DatumPlaneByThreePoints(
    point1 = (0, 0, -rs),
    point2 = (rc*cos(radians(120)), rc*sin(radians(120)), 0),
    point3 = (rc*cos(radians(240)), rc*sin(radians(240)), 0))
dp3 = part1.DatumPlaneByThreePoints(
    point1 = (0, 0, -rs),
    point2 = (rc*cos(radians(240)), rc*sin(radians(240)), 0),
    point3 = (rc*cos(0), rc*sin(0), 0))
dsys1 = part1.DatumCsysByThreePoints(coordSysType=CARTESIAN, name='Part 1 Sys',
    origin = (0.0, 0.0, 0.0),
    point1 = (rc*cos(0), rc*sin(0), 0),
    point2 = (rc*cos(radians(120)), rc*sin(radians(120)), 0))

# Partitions Along Datum Planes and Remove Excess Cells
part1.PartitionCellByDatumPlane(datumPlane = part1.datums[dp1.id],
    cells = part1.cells.findAt(((rc*cos(radians(60)), rc*sin(radians(60)), 0), )))
part1.PartitionCellByDatumPlane(datumPlane = part1.datums[dp2.id],
    cells = part1.cells.findAt(((rc*cos(radians(180)), rc*sin(radians(180)), 0), )))
part1.PartitionCellByDatumPlane(datumPlane = part1.datums[dp3.id],
    cells = part1.cells.findAt(((rc*cos(radians(300)), rc*sin(radians(300)), 0), )))
part1.RemoveFaces(deleteCells=False, faceList=(part1.faces.findAt(part1.cells.pointsOn[1][0], ),
    part1.faces.findAt(part1.cells.pointsOn[2][0], ), part1.faces.findAt(part1.cells.pointsOn[3][0],)))

# Coverts Solid Cell Into Shell
part1.RemoveCells(cellList=(part1.cells.findAt(part1.cells.pointsOn[0][0], ), ))

# Remove Small Faces Created with Partitions
part1.RepairSmallFaces(toleranceChecks=False,
    faceList = (part1.faces.findAt((part1.faces.pointsOn[4][0],),

```

```

                (part1.faces.pointsOn[5][0]),
                (part1.faces.pointsOn[6][0],,)))
part1.checkGeometry()

# Converts Imprecise Edges to Precise; may not make a difference; remove if time consuming
part1.ConvertToPrecise(method=TIGHTEN_GAPS)
part1.checkGeometry()

if (fillet == 0 and chamfer == 0):
    # Partition Shell Faces Along XY Plane
    part1.PartitionFaceByDatumPlane(datumPlane = part1.datums[dpXY.id],
        faces = (part1.faces.findAt((part1.faces.pointsOn[0][0]),
            (part1.faces.pointsOn[1][0]), (part1.faces.pointsOn[2][0]),
            (part1.faces.pointsOn[3][0]), (part1.faces.pointsOn[4][0],,)))

    # Faces ID for Sets
    whole_faces = (part1.faces.findAt((part1.faces.pointsOn[0][0]),
        (part1.faces.pointsOn[1][0]), (part1.faces.pointsOn[2][0]),
        (part1.faces.pointsOn[3][0]), (part1.faces.pointsOn[4][0]),
        (part1.faces.pointsOn[5][0]), (part1.faces.pointsOn[6][0]),
        (part1.faces.pointsOn[7][0],,)))
    Pia_faces = part1.faces.findAt((part1.faces.pointsOn[5][0],))
    Poa_faces = part1.faces.findAt((part1.faces.pointsOn[6][0],))

    # Datums and Sets for BCs to Replace Contact Plane
    D1 = part1.DatumCsysByThreePoints(name='D1', coordSysType = CARTESIAN,
        origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[8][0],),MIDDLE),
        point1 = part1.vertices.findAt(part1.vertices.pointsOn[1][0]),
        point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[12][0],),MIDDLE))
    D2 = part1.DatumCsysByThreePoints(name='D2', coordSysType = CARTESIAN,
        origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[0][0],),MIDDLE),
        point1 = part1.vertices.findAt(part1.vertices.pointsOn[0][0]),
        point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[14][0],),MIDDLE))
    D3 = part1.DatumCsysByThreePoints(name='D3', coordSysType = CARTESIAN,
        origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[4][0],),MIDDLE),
        point1 = part1.vertices.findAt(part1.vertices.pointsOn[4][0]),
        point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[6][0],),MIDDLE))

    part1.Set(name = 'F1', faces = part1.faces.findAt(
        (part1.faces.pointsOn[2][0]), (part1.faces.pointsOn[3][0],,)))
    part1.Set(name = 'F2', faces = part1.faces.findAt(
        (part1.faces.pointsOn[0][0],), (part1.faces.pointsOn[7][0],),),)
    part1.Set(name = 'F3', faces = part1.faces.findAt(
        (part1.faces.pointsOn[1][0],), (part1.faces.pointsOn[4][0],),),)
    part1.Surface(name = 'F1', side1Faces = part1.faces.findAt(
        (part1.faces.pointsOn[2][0]), (part1.faces.pointsOn[3][0],,)))
    part1.Surface(name = 'F2', side1Faces = part1.faces.findAt(
        (part1.faces.pointsOn[0][0],), (part1.faces.pointsOn[7][0],),),)
    part1.Surface(name = 'F3', side1Faces = part1.faces.findAt(
        (part1.faces.pointsOn[1][0],), (part1.faces.pointsOn[4][0],),),)
    part1.Set(name = 'E1', edges = part1.edges.findAt((part1.edges.pointsOn[8][0],,))
    part1.Set(name = 'E2', edges = part1.edges.findAt((part1.edges.pointsOn[0][0],,))
    part1.Set(name = 'E3', edges = part1.edges.findAt((part1.edges.pointsOn[4][0],,))

```

```

# Edges for Meshing Below
F123_edges = part1.edges.findAt(
    (part1.edges.pointsOn[0][0]), (part1.edges.pointsOn[2][0]),
    (part1.edges.pointsOn[4][0]), (part1.edges.pointsOn[6][0]),
    (part1.edges.pointsOn[8][0]), (part1.edges.pointsOn[10][0]),
    (part1.edges.pointsOn[12][0]),(part1.edges.pointsOn[13][0]),
    (part1.edges.pointsOn[14][0],))
V123_edges = part1.edges.findAt(
    (part1.edges.pointsOn[1][0]), (part1.edges.pointsOn[3][0]),
    (part1.edges.pointsOn[5][0]), (part1.edges.pointsOn[7][0]),
    (part1.edges.pointsOn[9][0]), (part1.edges.pointsOn[11][0],))
part1.Set(name = 'F123_edges', edges = F123_edges)
part1.Set(name = 'V123_edges', edges = V123_edges)

elif (fillet == 1 and chamfer == 0):
# Creates Fillets on Contact Corners
part1.Round(radius = r_fillet, edgeList = (
    part1.edges.findAt(part1.edges.pointsOn[0][0], ),
    part1.edges.findAt(part1.edges.pointsOn[2][0], ),
    part1.edges.findAt(part1.edges.pointsOn[5][0], )))

# Partition Shell Faces Along XY Plane
part1.PartitionFaceByDatumPlane(datumPlane = part1.datums[dpXY.id],
    faces = (part1.faces.findAt((part1.faces.pointsOn[0][0]),
    (part1.faces.pointsOn[1][0]), (part1.faces.pointsOn[2][0]),
    (part1.faces.pointsOn[3][0]), (part1.faces.pointsOn[4][0]),
    (part1.faces.pointsOn[5][0]), (part1.faces.pointsOn[6][0]),
    (part1.faces.pointsOn[7][0],)))

# Faces ID
whole_faces = (part1.faces.findAt((part1.faces.pointsOn[0][0]),
    (part1.faces.pointsOn[1][0]), (part1.faces.pointsOn[2][0]),
    (part1.faces.pointsOn[3][0]), (part1.faces.pointsOn[4][0]),
    (part1.faces.pointsOn[5][0]), (part1.faces.pointsOn[6][0]),
    (part1.faces.pointsOn[7][0]), (part1.faces.pointsOn[8][0]),
    (part1.faces.pointsOn[9][0]), (part1.faces.pointsOn[10][0]),
    (part1.faces.pointsOn[11][0]), (part1.faces.pointsOn[12][0]),
    (part1.faces.pointsOn[13][0],))
Pia_faces = part1.faces.findAt((part1.faces.pointsOn[8][0],))
Poa_faces = part1.faces.findAt((part1.faces.pointsOn[13][0],))

# Datums and Sets for BCs to Replace Contact Plane
D1 = part1.DatumCsysByThreePoints(name='D1', coordSysType = CARTESIAN,
    origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[14][0]),MIDDLE),
    point1 = part1.vertices.findAt(part1.vertices.pointsOn[11][0]),
    point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[16][0]),MIDDLE))
D2 = part1.DatumCsysByThreePoints(name='D2', coordSysType = CARTESIAN,
    origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[11][0]),MIDDLE),
    point1 = part1.vertices.findAt(part1.vertices.pointsOn[4][0]),
    point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[23][0]),MIDDLE))
D3 = part1.DatumCsysByThreePoints(name='D3', coordSysType = CARTESIAN,
    origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[0][0]),MIDDLE),

```

```

point1 = part1.vertices.findAt(part1.vertices.pointsOn[1][0]),
point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[2][0]),MIDDLE))

part1.Set(name='F1', faces = part1.faces.findAt(
    (part1.faces.pointsOn[4][0]), (part1.faces.pointsOn[12][0]),))
part1.Set(name='F2', faces = part1.faces.findAt(
    (part1.faces.pointsOn[3][0], ), (part1.faces.pointsOn[7][0], ), )
part1.Set(name='F3', faces = part1.faces.findAt(
    (part1.faces.pointsOn[0][0], ), (part1.faces.pointsOn[10][0], ), )
part1.Surface(name='F1', side1Faces = part1.faces.findAt(
    (part1.faces.pointsOn[4][0]), (part1.faces.pointsOn[12][0]),))
part1.Surface(name='F2', side1Faces = part1.faces.findAt(
    (part1.faces.pointsOn[3][0], ), (part1.faces.pointsOn[7][0], ), )
part1.Surface(name='F3', side1Faces = part1.faces.findAt(
    (part1.faces.pointsOn[0][0], ), (part1.faces.pointsOn[10][0], ), )
part1.Set(name='E1', edges = part1.edges.findAt((part1.edges.pointsOn[14][0]),))
part1.Set(name='E2', edges = part1.edges.findAt((part1.edges.pointsOn[11][0]),))
part1.Set(name='E3', edges = part1.edges.findAt((part1.edges.pointsOn[0][0]),))

elif (fillet == 0 and chamfer == 1):
    # Creates Chamfer on Contact Corners
    part1.Chamfer(length = l_chamfer, edgeList = (
        part1.edges.findAt(part1.edges.pointsOn[0][0], ),
        part1.edges.findAt(part1.edges.pointsOn[2][0], ),
        part1.edges.findAt(part1.edges.pointsOn[5][0], )))

    # Partition Shell Faces Along XY Plane
    part1.PartitionFaceByDatumPlane(datumPlane = part1.datums[dpXY.id],
        faces = (part1.faces.findAt((part1.faces.pointsOn[0][0]),
            (part1.faces.pointsOn[1][0]), (part1.faces.pointsOn[2][0]),
            (part1.faces.pointsOn[3][0]), (part1.faces.pointsOn[4][0]),
            (part1.faces.pointsOn[5][0]), (part1.faces.pointsOn[6][0]),
            (part1.faces.pointsOn[7][0]),)))

    # Faces ID
    whole_faces = (part1.faces.findAt((part1.faces.pointsOn[0][0]),
        (part1.faces.pointsOn[1][0]), (part1.faces.pointsOn[2][0]),
        (part1.faces.pointsOn[3][0]), (part1.faces.pointsOn[4][0]),
        (part1.faces.pointsOn[5][0]), (part1.faces.pointsOn[6][0]),
        (part1.faces.pointsOn[7][0]), (part1.faces.pointsOn[8][0]),
        (part1.faces.pointsOn[9][0]), (part1.faces.pointsOn[10][0]),
        (part1.faces.pointsOn[11][0]), (part1.faces.pointsOn[12][0]),
        (part1.faces.pointsOn[13][0]),))
    Pia_faces = part1.faces.findAt((part1.faces.pointsOn[9][0],))
    Poa_faces = part1.faces.findAt((part1.faces.pointsOn[12][0],))

    # Datums and Sets for BCs to Replace Contact Plane
    D1 = part1.DatumCsysByThreePoints(name='D1', coordSysType = CARTESIAN,
        origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[0][0]),MIDDLE),
        point1 = part1.vertices.findAt(part1.vertices.pointsOn[1][0]),
        point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[2][0]),MIDDLE))
    D2 = part1.DatumCsysByThreePoints(name='D2', coordSysType = CARTESIAN,
        origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[14][0]),MIDDLE),

```

```

point1 = part1.vertices.findAt(part1.vertices.pointsOn[11][0],)
point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[16][0],MIDDLE))
D3 = part1.DatumCsysByThreePoints(name='D3', coordSysType = CARTESIAN,
origin = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[11][0],MIDDLE),
point1 = part1.vertices.findAt(part1.vertices.pointsOn[4][0],)
point2 = part1.InterestingPoint(part1.edges.findAt(part1.edges.pointsOn[29][0],MIDDLE))

part1.Set(name='F1', faces = part1.faces.findAt(
(part1.faces.pointsOn[0][0],) , (part1.faces.pointsOn[11][0],))
part1.Set(name='F2', faces = part1.faces.findAt(
(part1.faces.pointsOn[4][0],) , (part1.faces.pointsOn[13][0],) , ) , )
part1.Set(name='F3', faces = part1.faces.findAt(
(part1.faces.pointsOn[3][0],) , (part1.faces.pointsOn[10][0],) , ) , )
part1.Surface(name='F1', side1Faces = part1.faces.findAt(
(part1.faces.pointsOn[0][0],) , (part1.faces.pointsOn[11][0],))
part1.Surface(name='F2', side1Faces = part1.faces.findAt(
(part1.faces.pointsOn[4][0],) , (part1.faces.pointsOn[13][0],) , ) , )
part1.Surface(name='F3', side1Faces = part1.faces.findAt(
(part1.faces.pointsOn[3][0],) , (part1.faces.pointsOn[10][0],) , ) , )

part1.Set(name='E1', edges = part1.edges.findAt((part1.edges.pointsOn[0][0],))
part1.Set(name='E2', edges = part1.edges.findAt((part1.edges.pointsOn[14][0],))
part1.Set(name='E3', edges = part1.edges.findAt((part1.edges.pointsOn[11][0],))

part1.Set(name='V1', edges = part1.edges.findAt(
(part1.edges.pointsOn[3][0],) , (part1.edges.pointsOn[7][0],) ,
(part1.edges.pointsOn[11][0],) , (part1.edges.pointsOn[22][0],))
part1.Set(name='V2', edges = part1.edges.findAt(
(part1.edges.pointsOn[9][0],) , (part1.edges.pointsOn[24][0],) ,
(part1.edges.pointsOn[15][0],) , (part1.edges.pointsOn[18][0],))
part1.Set(name='V3', edges = part1.edges.findAt(
(part1.edges.pointsOn[12][0],) , (part1.edges.pointsOn[25][0],) ,
(part1.edges.pointsOn[5][0],) , (part1.edges.pointsOn[21][0],))

# Edges for Meshing Below
F123_edges = part1.edges.findAt(
(part1.edges.pointsOn[2][0],) , (part1.edges.pointsOn[13][0],) ,
(part1.edges.pointsOn[16][0],) , (part1.edges.pointsOn[27][0],) ,
(part1.edges.pointsOn[28][0],) , (part1.edges.pointsOn[29][0],) ,
(part1.edges.pointsOn[0][0],) , (part1.edges.pointsOn[11][0],) ,
(part1.edges.pointsOn[14][0],)
V123_edges = part1.edges.findAt(
(part1.edges.pointsOn[3][0],) , (part1.edges.pointsOn[7][0],) ,
(part1.edges.pointsOn[11][0],) , (part1.edges.pointsOn[22][0],) ,
(part1.edges.pointsOn[9][0],) , (part1.edges.pointsOn[24][0],) ,
(part1.edges.pointsOn[15][0],) , (part1.edges.pointsOn[18][0],) ,
(part1.edges.pointsOn[12][0],) , (part1.edges.pointsOn[25][0],) ,
(part1.edges.pointsOn[5][0],) , (part1.edges.pointsOn[21][0],)
part1.Set(name = 'F123_edges', edges = F123_edges)
part1.Set(name = 'V123_edges', edges = V123_edges)

# Whole Set

```

```

part1.Set(name='Whole', faces = whole_faces)

# Pressure Sets
part1.Set(name = 'Pia', faces = Pia_faces)
part1.Set(name = 'Poa', faces = Poa_faces)
part1.Set(name = 'Pta', faces = whole_faces)

# Pressure Surfaces
part1.Surface(name = 'Pia', side1Faces = Pia_faces)
part1.Surface(name = 'Poa', side1Faces = Poa_faces)
part1.Surface(name = 'Pta', side2Faces = whole_faces)
part1.Surface(name = 'Contact', side1Faces = whole_faces)

# Assigns Section
if memb_shell == 1:
    # Shell
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
        region = part1.sets['Whole'], sectionName = 'Shell', thicknessAssignment=FROM_SECTION)
else:
    # Membrane
    part1.SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE,
        region = part1.sets['Whole'], sectionName = 'Membrane', thicknessAssignment=FROM_SECTION)

# Mesh
if memb_shell == 1:
    part1.setElementType(regions = part1.sets['Whole'], elemTypes=(
        ElemType(elemCode=S4, elemLibrary=EXPLICIT, secondOrderAccuracy=OFF),
        ElemType(elemCode=S3R, elemLibrary=EXPLICIT, secondOrderAccuracy=OFF)),)
else:
    part1.setElementType(regions = part1.sets['Whole'], elemTypes=(
        ElemType(elemCode=M3D4, elemLibrary=EXPLICIT, secondOrderAccuracy=OFF),
        ElemType(elemCode=M3D3, elemLibrary=EXPLICIT, secondOrderAccuracy=OFF)),)

# part1.seedPart(deviationFactor=0.1, minSizeFactor=0.1, size=p1_seed)
part1.seedEdgeByBias(biasMethod = DOUBLE, constraint = FINER, endEdges = F123_edges,
    minSize = p1_seed[0], maxSize = p1_seed[1])
part1.seedEdgeBySize(constraint = FINER, deviationFactor = 0.1, edges = V123_edges,
    size = p1_seed[0])
part1.setMeshControls(elemShape=TRI, regions = whole_faces)
part1.generateMesh()
part1.flipNormal(regions = part1.sets['Whole'])

#####
#####
# Part 2
#####
#####

if Contact_or_BC == 0:
    # Create Part
    model.Part(name = p2_name, dimensionality=THREE_D, type=DISCRETE_RIGID_SURFACE)
    part2 = model.parts[p2_name]

```

```

# Datum System for Assembly
dsys2 = part2.DatumCsysByThreePoints(coordSysType=CARTESIAN, name='Part 2 Sys',
    origin = (0.0, 0.0, 0.0),
    point1 = (rc*cos(0), rc*sin(0), 0),
    point2 = (rc*cos(radians(120)), rc*sin(radians(120)), 0))
dpXY = part2.DatumPlaneByPrincipalPlane(offset=0.0, principalPlane=XYPLANE)

# Wire for Smallest X-Section
rs_new = 0.10*rs
rcs = rs_new*(rc+0.90*hm)/rs
part2.WirePolyLine(mergeType=IMPRINT, meshable=ON, points = (
    ((rcs*cos(0), rcs*sin(0), -rs + rs_new), (rcs*cos(radians(120)), rcs*sin(radians(120)), -rs + rs_new)),
    ((rcs*cos(radians(120)), rcs*sin(radians(120)), -rs + rs_new), (rcs*cos(radians(240)),
rcs*sin(radians(240)), -rs + rs_new)),
    ((rcs*cos(radians(240)), rcs*sin(radians(240)), -rs + rs_new), (rcs*cos(0), rcs*sin(0), -rs + rs_new))))

# Wire for Largest X-Section
rs_new = 1.5*rs
rcs = rs_new*(rc+0.90*hm)/rs
part2.WirePolyLine(mergeType=IMPRINT, meshable=ON, points = (
    ((rcs*cos(0), rcs*sin(0), -rs + rs_new), (rcs*cos(radians(120)), rcs*sin(radians(120)), -rs + rs_new)),
    ((rcs*cos(radians(120)), rcs*sin(radians(120)), -rs + rs_new), (rcs*cos(radians(240)),
rcs*sin(radians(240)), -rs + rs_new)),
    ((rcs*cos(radians(240)), rcs*sin(radians(240)), -rs + rs_new), (rcs*cos(0), rcs*sin(0), -rs + rs_new))))

# Loft to Form Cone
part2.ShellLoft(startCondition = NONE, endCondition = NONE, loftsections = (
    (part2.edges.findAt(part2.edges.pointsOn[3][0], ),
    part2.edges.findAt(part2.edges.pointsOn[4][0], ),
    part2.edges.findAt(part2.edges.pointsOn[5][0], )),
    (part2.edges.findAt(part2.edges.pointsOn[0][0], ),
    part2.edges.findAt(part2.edges.pointsOn[1][0], ),
    part2.edges.findAt(part2.edges.pointsOn[2][0],))),)

## if (fillet == 0 and chamfer == 0):
# Partition Shell Faces Along XY Plane
part2.PartitionFaceByDatumPlane(datumPlane = part2.datums[dpXY.id],
    faces = (part2.faces.findAt((part2.faces.pointsOn[0][0],),
    (part2.faces.pointsOn[1][0],), (part2.faces.pointsOn[2][0],),)))

# Faces ID
whole_faces = (part2.faces.findAt((part2.faces.pointsOn[0][0],),
    (part2.faces.pointsOn[1][0],), (part2.faces.pointsOn[2][0],),
    (part2.faces.pointsOn[3][0],), (part2.faces.pointsOn[4][0],),
    (part2.faces.pointsOn[5][0],),))

if contact_cone_top_hat == 1:
    part2.CoverEdges(edgeList=(
        part2.edges.findAt(part2.edges.pointsOn[6][0], ),
        part2.edges.findAt(part2.edges.pointsOn[12][0], ),
        part2.edges.findAt(part2.edges.pointsOn[14][0], ),),tryAnalytical=False)

# Faces ID

```

```

whole_faces = (part2.faces.findAt((part2.faces.pointsOn[0][0]),
    (part2.faces.pointsOn[1][0]), (part2.faces.pointsOn[2][0]),
    (part2.faces.pointsOn[3][0]), (part2.faces.pointsOn[4][0]),
    (part2.faces.pointsOn[5][0]), (part2.faces.pointsOn[6][0]),))

# Creates Set
part2.Set(name='Whole', faces = whole_faces)

# Create Surface
part2.Surface(name='Contact', side2Faces = whole_faces)

# Mesh
part2.seedPart(deviationFactor=0.1, minSizeFactor=0.1, size=p2_seed)
part2.setElementType(regions = part2.sets['Whole'],
    elemTypes=(ElemType(elemCode=R3D4, elemLibrary=EXPLICIT),
        ElemType(elemCode=R3D3, elemLibrary=EXPLICIT)),)
part2.setMeshControls(elemShape=TRI, regions = whole_faces)
part2.generateMesh()
part2.flipNormal(regions = part2.sets['Whole'])

#####
#####
# Assembly
#####
#####

# Creates Instances
assembly.Instance(dependent=ON, name=p1_name, part=part1)
inst1 = assembly.instances[p1_name]

if Contact_or_BC == 0:
    assembly.Instance(dependent=ON, name=p2_name, part=part2)
    inst2 = assembly.instances[p2_name]

# Moves and Co-Locates Parts to Center Point Per MATLAB-Obtained Center Points
dsysUC = assembly.DatumCsysByThreePoints(coordSysType=CARTESIAN, name='UC1 Sys',
    origin = (0.0, 0.0, rs),
    point1 = (rc*cos(0), rc*sin(0), rs),
    point2 = (rc*cos(radians(120)), rc*sin(radians(120)), rs))

assembly.ParallelCsys(fixedCsys = assembly.datums[dsysUC.id],
    movableCsys = inst1.datums[dsys1.id])
assembly.CoincidentPoint(fixedPoint = assembly.datums[dsysUC.id].origin,
    movablePoint = inst1.datums[dsys1.id].origin)

if Contact_or_BC == 0:
    assembly.ParallelCsys(fixedCsys = assembly.datums[dsysUC.id],
        movableCsys = inst2.datums[dsys2.id])
    assembly.CoincidentPoint(fixedPoint = assembly.datums[dsysUC.id].origin,
        movablePoint = inst2.datums[dsys2.id].origin)

# Reference Point at Origin for Rigid Constraint
RP = assembly.ReferencePoint(point=(0,0,0))

```

```

assembly.features.changeKey(fromName='RP-1', toName='Origin')
assembly.Set(name='Origin RP', referencePoints=(assembly.referencePoints[RP.id],))

#####
#####
# Interactions
#####
#####

# Creates Fluid Cavity w Initial P and T, and History Request
model.FluidCavity(createStepName='Initial', ambientPressure=Pamb, useAdiabatic=False,
    interactionProperty='Pneumatic Air Cavity',
    cavityPoint = assembly.instances[p1_name].sets['UC RP'],
    cavitySurface = assembly.instances[p1_name].surfaces['Pta'],
    name = 'UC Cavity')
model.FluidCavityPressure(fluidPressure=Pi,
    name = 'UC Cavity Pi',
    fluidCavity = 'UC Cavity')
model.Temperature(createStepName='Initial',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS,distributionType=UNIFORM,
    magnitudes=(288.15, ),
    name = 'UC Cavity Ti',
    region = assembly.instances[p1_name].sets['UC RP'])

# General Contact
if Contact_or_BC == 0:
    cont_name = 'General Self Contact'
    if step_type == 'NLED':
        model.ContactExp(createStepName='Initial', name = cont_name)
        model.interactions[cont_name].includedPairs.setValuesInStep(stepName='Initial', useAllstar=ON)
        model.interactions[cont_name].contactPropertyAssignments.appendInStep(
            assignments=((GLOBAL, SELF, 'Contact Definitions'), ), stepName='Initial')
        model.interactions[cont_name].masterSlaveAssignments.appendInStep(stepName='Initial',
            assignments = ((inst2.surfaces['Contact'], inst1.surfaces['Contact'], MASTER), ))
    else:
        model.ContactStd(createStepName='Initial', name=cont_name)
        model.interactions[cont_name].includedPairs.setValuesInStep(stepName='Initial', useAllstar=ON)
        model.interactions[cont_name].contactPropertyAssignments.appendInStep(
            assignments=((GLOBAL, SELF, 'Contact Definitions'), ), stepName='Initial')
        model.interactions[cont_name].initializationAssignments.appendInStep(stepName='Initial',
            assignments=((inst2.surfaces['Contact'], inst1.surfaces['Contact'],'Contact Initialization'), ))
        model.interactions[cont_name].masterSlaveAssignments.appendInStep(stepName='Initial',
            assignments = ((inst2.surfaces['Contact'], inst1.surfaces['Contact'], MASTER), ))

#####
#####
# Constraints
#####
#####

# Rigid Constraint for Y-Sym Plane
if Contact_or_BC == 0:
    model.RigidBody(name= p2_name + ' Constraint',

```

```

bodyRegion = assembly.instances[p2_name].sets['Whole'],
refPointRegion = assembly.sets['Origin RP'])

# Circular Face Coupling to RP for BC Application
if ((Contact_or_BC == 1 and BCthroughRP == 1) or (fillet == 0 and chamfer == 0)):
    if (fillet == 0 and chamfer == 0):
        p1 = inst1.edges.pointsOn[8][0]
        p2 = inst1.edges.pointsOn[0][0]
        p3 = inst1.edges.pointsOn[4][0]

    elif (fillet == 1 and chamfer == 0):
        p1 = inst1.edges.pointsOn[14][0]
        p2 = inst1.edges.pointsOn[11][0]
        p3 = inst1.edges.pointsOn[0][0]

    elif (fillet == 0 and chamfer == 1):
        p1 = inst1.edges.pointsOn[0][0]
        p2 = inst1.edges.pointsOn[14][0]
        p3 = inst1.edges.pointsOn[11][0]

F1_RP = assembly.ReferencePoint(point =
    inst1.InterestingPoint(inst1.edges.findAt(p1), MIDDLE))
assembly.features.changeKey(fromName='RP-1', toName='F1 RP')
assembly.Set(name='F1 RP', referencePoints=(assembly.referencePoints[F1_RP.id],))
F2_RP = assembly.ReferencePoint(point =
    inst1.InterestingPoint(inst1.edges.findAt(p2), MIDDLE))
assembly.features.changeKey(fromName='RP-1', toName='F2 RP')
assembly.Set(name='F2 RP', referencePoints=(assembly.referencePoints[F2_RP.id],))
F3_RP = assembly.ReferencePoint(point =
    inst1.InterestingPoint(inst1.edges.findAt(p3), MIDDLE))
assembly.features.changeKey(fromName='RP-1', toName='F3 RP')
assembly.Set(name='F3 RP', referencePoints=(assembly.referencePoints[F3_RP.id],))

model.Coupling(couplingType = KINEMATIC, influenceRadius = WHOLE_SURFACE, #
KINEMATIC, DISTRIBUTING, STRUCTURAL
    u1 = ON, u2 = ON, u3 = ON, ur1 = OFF, ur2 = OFF, ur3 = OFF,
    name = 'F1',
    controlPoint = assembly.sets['F1 RP'],
    surface = inst1.sets['F1'],
    localCsys = inst1.datums[D1.id])
model.Coupling(couplingType = KINEMATIC, influenceRadius = WHOLE_SURFACE,
    u1 = ON, u2 = ON, u3 = ON, ur1 = OFF, ur2 = OFF, ur3 = OFF,
    name = 'F2',
    controlPoint = assembly.sets['F2 RP'],
    surface = inst1.sets['F2'],
    localCsys = inst1.datums[D2.id])
model.Coupling(couplingType = KINEMATIC, influenceRadius = WHOLE_SURFACE,
    u1 = ON, u2 = ON, u3 = ON, ur1 = OFF, ur2 = OFF, ur3 = OFF,
    name = 'F3',
    controlPoint = assembly.sets['F3 RP'],
    surface = inst1.sets['F3'],
    localCsys = inst1.datums[D3.id])

```

```

if (fillet == 0 and chamfer == 0):
    p1 = part1.vertices.pointsOn[1][0]
    p2 = part1.vertices.pointsOn[4][0]
    d = sqrt((p2[0] - p1[0])**2 + (p2[1] - p1[1])**2 + (p2[2] - p1[2])**2)/2
    model.constraints['F1'].setValues(influenceRadius = 0.999*d)
    model.constraints['F2'].setValues(influenceRadius = 0.999*d)
    model.constraints['F3'].setValues(influenceRadius = 0.999*d)

#####
#####
# Steps, Loads, Boundary Conditions (BC), and Outputs
#####
#####

step1 = 'Pressurize'
step2 = 'Vacuum'

# Creates Step(s)
if pre_pressurize == 0:
    step1 = step2
    if step_type == 'NLIS':
        model.StaticStep(name=step2, previous='Initial', nlgeom=ON,
            minInc = vacc_step[1], maxInc = vacc_step[2], initialInc = vacc_step[3], maxNumInc = vacc_step[4],
            stabilizationMethod=DISSIPATED_ENERGY_FRACTION, continueDampingFactors=False,
            adaptiveDampingRatio=0.05, stabilizationMagnitude=0.0002,
            matrixSolver=DIRECT, matrixStorage=UNSYMMETRIC,convertSDI=CONVERT_SDI_ON)
    elif step_type == 'NLID':
        if int_operator == 'QS':
            model.ImplicitDynamicsStep(name=step2, previous='Initial',alpha=DEFAULT, amplitude=RAMP,
                application=QUASI_STATIC, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = vacc_step[0], minInc = vacc_step[1], maxInc = vacc_step[2],
                initialInc = vacc_step[3], maxNumInc = vacc_step[4])
        elif int_operator == 'MD':
            model.ImplicitDynamicsStep(name=step2, previous='Initial',alpha=DEFAULT, amplitude=RAMP,
                application=MODERATE DISSIPATION, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = vacc_step[0], minInc = vacc_step[1], maxInc = vacc_step[2],
                initialInc = vacc_step[3], maxNumInc = vacc_step[4])
        elif int_operator == 'TF':
            model.ImplicitDynamicsStep(name=step2, previous='Initial',alpha=DEFAULT, amplitude=RAMP,
                application=TRANSIENT_FIDELITY, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = vacc_step[0], minInc = vacc_step[1], maxInc = vacc_step[2],
                initialInc = vacc_step[3], maxNumInc = vacc_step[4])
    elif (step_type == 'NLED' and adiabatic == 'ON'):
        model.ExplicitDynamicsStep(name = step2, previous = 'Initial',
            timePeriod = vacc_step[0], adiabatic = ON,
            linearBulkViscosity = linearBulkVisco, quadBulkViscosity = quadBulkVisco)
    elif (step_type == 'NLED' and adiabatic == 'OFF'):
        model.ExplicitDynamicsStep(name = step2, previous = 'Initial',
            timePeriod = vacc_step[0],
            linearBulkViscosity = linearBulkVisco, quadBulkViscosity = quadBulkVisco)

```

End of If Statement~::~::~::~::~

```
if pre_pressurize == 1:
    if step_type == 'NLIS':
        model.StaticStep(name=step1, previous='Initial', nlgeom=ON,
            minInc = preP_step[1], maxInc = preP_step[2], initialInc = preP_step[3], maxNumInc = preP_step[4],
            stabilizationMethod=DISSIPATED_ENERGY_FRACTION, continueDampingFactors=False,
            adaptiveDampingRatio=0.05, stabilizationMagnitude=0.0002,
            matrixSolver=DIRECT, matrixStorage=UNSYMMETRIC,convertSDI=CONVERT_SDI_ON)
        model.StaticStep(name=step2, previous=step1, nlgeom=ON,
            minInc = vacc_step[1], maxInc = vacc_step[2], initialInc = vacc_step[3], maxNumInc = vacc_step[4],
            stabilizationMethod=DISSIPATED_ENERGY_FRACTION, continueDampingFactors=False,
            adaptiveDampingRatio=0.05, stabilizationMagnitude=0.0002,
            matrixSolver=DIRECT, matrixStorage=UNSYMMETRIC,convertSDI=CONVERT_SDI_ON)
    elif step_type == 'NLID':
        if int_operator == 'QS':
            model.ImplicitDynamicsStep(name=step1, previous='Initial',alpha=DEFAULT,amplitude=RAMP,
                application=QUASI_STATIC, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = preP_step[0], minInc = preP_step[1], maxInc = preP_step[2],
                initialInc = preP_step[3], maxNumInc = preP_step[4])
            model.ImplicitDynamicsStep(name=step2, previous=step1,alpha=DEFAULT, amplitude=RAMP,
                application=QUASI_STATIC, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = vacc_step[0], minInc = vacc_step[1], maxInc = vacc_step[2],
                initialInc = vacc_step[3], maxNumInc = vacc_step[4])
        elif int_operator == 'MD':
            model.ImplicitDynamicsStep(name=step1, previous='Initial',alpha=DEFAULT,amplitude=RAMP,
                application=MODERATE DISSIPATION, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = preP_step[0], minInc = preP_step[1], maxInc = preP_step[2],
                initialInc = preP_step[3], maxNumInc = preP_step[4])
            model.ImplicitDynamicsStep(name=step2, previous=step1,alpha=DEFAULT, amplitude=RAMP,
                application=MODERATE DISSIPATION, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = vacc_step[0], minInc = vacc_step[1], maxInc = vacc_step[2],
                initialInc = vacc_step[3], maxNumInc = vacc_step[4])
        elif int_operator == 'TF':
            model.ImplicitDynamicsStep(name=step1, previous='Initial',alpha=DEFAULT,amplitude=RAMP,
                application=TRANSIENT_FIDELITY, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = preP_step[0], minInc = preP_step[1], maxInc = preP_step[2],
                initialInc = preP_step[3], maxNumInc = preP_step[4])
            model.ImplicitDynamicsStep(name=step2, previous=step1,alpha=DEFAULT, amplitude=RAMP,
                application=TRANSIENT_FIDELITY, convertSDI=CONVERT_SDI_ON,
                initialConditions=ON, matrixStorage=UNSYMMETRIC, nlgeom=ON, nohaf=OFF,
                timePeriod = vacc_step[0], minInc = vacc_step[1], maxInc = vacc_step[2],
                initialInc = vacc_step[3], maxNumInc = vacc_step[4])
    elif (step_type == 'NLED' and adiabatic == 'ON'):
        model.ExplicitDynamicsStep(name=step1, previous='Initial',
            timePeriod = preP_step[0], adiabatic = ON,
            linearBulkViscosity = linearBulkVisco, quadBulkViscosity = quadBulkVisco)
```

```

model.ExplicitDynamicsStep(name=step2, previous=step1,
    timePeriod = vacc_step[0], adiabatic = ON,
    linearBulkViscosity = linearBulkVisco, quadBulkViscosity = quadBulkVisco)
elif (step_type == 'NLED' and adiabatic == 'OFF'):
    model.ExplicitDynamicsStep(name=step1, previous='Initial',
        timePeriod = preP_step[0],
        linearBulkViscosity = linearBulkVisco, quadBulkViscosity = quadBulkVisco)
model.ExplicitDynamicsStep(name=step2, previous=step1,
    timePeriod = vacc_step[0],
    linearBulkViscosity = linearBulkVisco, quadBulkViscosity = quadBulkVisco)

# End of If Statement~~~~~

# Step Control Edits
if (step_type == 'NLIS' or step_type == 'NLID'):
    model.steps[step1].control.setValues(allowPropagation=OFF,                resetDefaultValues=OFF,
    discontinuous=ON,
        displacementField = (0.005, 0.01, 0.0, 0.0, 0.02, 1e-06, 0.001, 1e-08, 1.0, 1e-06, 1e-08),
        rotationField    = (0.005, 0.01, 0.0, 0.0, 0.02, 1e-06, 0.001, 1e-08, 1.0, 1e-06),
        hydrostaticFluidPressureField = (0.005, 0.01, 0.0, 0.0, 0.02, 1e-13, 0.001, 1e-08, 1.0, 1e-13),
        timeIncrementation = (8.0, 10.0, 9.0, 16.0, 10.0, 4.0, 12.0, 30.0, 6.0, 3.0,50.0))
    if pre_pressurize == 1:
        model.steps[step2].control.setValues(allowPropagation=OFF,                resetDefaultValues=OFF,
        discontinuous=ON,
            displacementField = (0.005, 0.01, 0.0, 0.0, 0.02, 1e-06, 0.001, 1e-08, 1.0, 1e-06, 1e-08),
            rotationField    = (0.005, 0.01, 0.0, 0.0, 0.02, 1e-06, 0.001, 1e-08, 1.0, 1e-06),
            hydrostaticFluidPressureField = (0.005, 0.01, 0.0, 0.0, 0.02, 1e-13, 0.001, 1e-08, 1.0, 1e-13),
            timeIncrementation = (8.0, 10.0, 9.0, 16.0, 10.0, 4.0, 12.0, 30.0, 6.0, 3.0,50.0))
        model.steps[step2].control.setValues(lineSearch=(5.0, 1.0, 0.0001, 0.25, 0.1))

# Vacuum Load
if step_type == 'NLIS':
    model.Pressure(name='Vacuum', amplitude=UNSET, magnitude=-Pvac, createStepName=step2,
        distributionType=UNIFORM, field="",region = assembly.allInstances[p1_name].surfaces['Pia'])
else:
    model.Pressure(name='Vacuum', amplitude='Internal Load', magnitude=-Pvac, createStepName=step2,
        distributionType=UNIFORM, field="",region = assembly.allInstances[p1_name].surfaces['Pia'])

# Cavity Pressure BC Load
if pre_pressurize == 1:
    if step_type == 'NLIS':
        model.FluidCavityPressureBC(fixed=OFF, amplitude=UNSET, createStepName=step1,
            magnitude = Ppress,
            fluidCavity = 'UC Cavity',
            name      = 'UC Cavity P')
        model.boundaryConditions['UC Cavity P'].deactivate(step2)
    else:
        model.FluidCavityPressureBC(fixed=OFF, amplitude='Cavity P', createStepName=step1,
            magnitude = Ppress,
            fluidCavity = 'UC Cavity',
            name      = 'UC Cavity P')
        model.boundaryConditions['UC Cavity P'].deactivate(step2)

```

```

# BCs
if Contact_or_BC == 0:
    model.EncastreBC(name = p2_name + ' Encastre', createStepName=step1, localCsys=None,
        region = assembly.sets['Origin RP'])

if ((Contact_or_BC == 1 and BCthroughRP == 1) or (fillet == 0 and chamfer == 0)):
    region1 = assembly.sets['F1 RP']
    region2 = assembly.sets['F2 RP']
    region3 = assembly.sets['F3 RP']
else:
    region1 = inst1.sets['F1']
    region2 = inst1.sets['F2']
    region3 = inst1.sets['F3']

if pre_pressurize == 1:
    model.DisplacementBC(name='F1 Pre-P', amplitude=UNSET, createStepName = step1,
        localCsys = inst1.datums[D1.id],
        region = region1, # p1_name + '.F1'
        distributionType = UNIFORM, fieldName="", fixed = OFF,
        u1=UNSET, u2=UNSET, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET)
    model.DisplacementBC(name='F2 Pre-P', amplitude=UNSET, createStepName = step1,
        localCsys = inst1.datums[D2.id],
        region = region2, # p1_name + '.F2'
        distributionType = UNIFORM, fieldName="", fixed = OFF,
        u1=UNSET, u2=UNSET, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET)
    model.DisplacementBC(name='F3 Pre-P', amplitude=UNSET, createStepName = step1,
        localCsys = inst1.datums[D3.id],
        region = region3, # p1_name + '.F3'
        distributionType = UNIFORM, fieldName="", fixed = OFF,
        u1=UNSET, u2=UNSET, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET)
    model.boundaryConditions['F1 Pre-P'].deactivate(step2)
    model.boundaryConditions['F2 Pre-P'].deactivate(step2)
    model.boundaryConditions['F3 Pre-P'].deactivate(step2)

if Contact_or_BC == 1:
    model.DisplacementBC(name='F1 Vacuum', amplitude=UNSET, createStepName = step2,
        localCsys = inst1.datums[D1.id],
        region = region1, # p1_name + '.F1'
        distributionType = UNIFORM, fieldName="", fixed = OFF,
        u1=UNSET, u2=UNSET, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET)
    model.DisplacementBC(name='F2 Vacuum', amplitude=UNSET, createStepName = step2,
        localCsys = inst1.datums[D2.id],
        region = region2, # p1_name + '.F2'
        distributionType = UNIFORM, fieldName="", fixed = OFF,
        u1=UNSET, u2=UNSET, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET)
    model.DisplacementBC(name='F3 Vacuum', amplitude=UNSET, createStepName = step2,
        localCsys = inst1.datums[D3.id],
        region = region3, # p1_name + '.F3'
        distributionType = UNIFORM, fieldName="", fixed = OFF,
        u1=UNSET, u2=UNSET, u3=0.0, ur1=UNSET, ur2=UNSET, ur3=UNSET)

# Field Outputs
model.FieldOutputRequest(name = p1_name + ' Poa S Coords', frequency = O_freq, createStepName=step1,

```

```

    region = assembly.allInstances[p1_name].sets['Poa'],
    variables = ('COORD',))
model.FieldOutputRequest(name = p1_name + ' Pia Load Input', frequency = O_freq,
createStepName=step1,
    region = assembly.allInstances[p1_name].sets['Pia'],
    variables = ('P',))

if step_type == 'NLIS':
    model.FieldOutputRequest(name = p1_name + ' Whole', frequency = O_freq, createStepName=step1,
        region = assembly.allInstances[p1_name].sets['Whole'],
        variables = ('S','LE','U','RF','P','CSTRESS','CDISP','COORD'))
    model.FieldOutputRequest(name = 'Whole', frequency = O_freq, createStepName=step1,
        region = MODEL,
        variables = ('S','LE','U','RF','P','CSTRESS','CDISP','COORD'))
else:
    model.FieldOutputRequest(name = p1_name + ' Whole', frequency = O_freq, createStepName=step1,
        region = assembly.allInstances[p1_name].sets['Whole'],
        variables = ('S','LE','U','V','A','RF','P','CSTRESS','CDISP','COORD','ERV','VP'))
    model.FieldOutputRequest(name = 'Whole', frequency = O_freq, createStepName=step1,
        region = MODEL,
        variables = ('S','LE','U','V','A','RF','P','CSTRESS','CDISP','COORD','ERV','VP'))

# History Outputs
if Contact_or_BC == 0:
    model.HistoryOutputRequest(name = p2_name + ' RP', frequency = O_freq, createStepName = step1,
        rebar = EXCLUDE, sectionPoints = DEFAULT,
        region = assembly.sets['Origin RP'],
        variables =('RF1', 'RF2', 'RF3', 'RM1', 'RM2', 'RM3'))

if (Contact_or_BC == 1 and BCthroughRP == 1):
    model.HistoryOutputRequest(name = 'F1 RP', frequency = O_freq, createStepName = step1,
        rebar = EXCLUDE, sectionPoints = DEFAULT, useGlobal=OFF,
        region = assembly.sets['F1 RP'],
        variables =('RF1', 'RF2', 'RF3'))
    model.HistoryOutputRequest(name = 'F2 RP', frequency = O_freq, createStepName = step1,
        rebar = EXCLUDE, sectionPoints = DEFAULT, useGlobal=OFF,
        region = assembly.sets['F2 RP'],
        variables =('RF1', 'RF2', 'RF3'))
    model.HistoryOutputRequest(name = 'F3 RP', frequency = O_freq, createStepName = step1,
        rebar = EXCLUDE, sectionPoints = DEFAULT, useGlobal=OFF,
        region = assembly.sets['F3 RP'],
        variables =('RF1', 'RF2', 'RF3'))

if step_type == 'NLED':
    model.HistoryOutputRequest(name = 'Whole', frequency = O_freq, createStepName=step1,
        variables = ('ALLAE', 'ALLCD', 'ALLDC', 'ALLDMD', 'ALLFD', 'ALLIE', 'ALLKE',
            'ALLPD', 'ALLSE', 'ALLVD', 'ALLWK', 'ALLCW', 'ALLMW', 'ALLPW', 'ETOTAL'))
    model.HistoryOutputRequest(createStepName=step1, rebar=EXCLUDE, sectionPoints=DEFAULT,
        variables = ('PCAV', 'CVOL', 'CTEMP', 'CSAREA', 'CMASS'),
        name = 'UC Cavity',
        region = assembly.instances[p1_name].sets['UC RP'],
        frequency = O_freq)
else:

```

```

model.HistoryOutputRequest(name = 'Whole', frequency = O_freq, createStepName=step1,
    variables = ('CFTM','CFT1','CFT2','CFT3','CAREA','XT1','XT2','XT3',
        'ALLAE','ALLCD','ALLDMD','ALLEE','ALLFD','ALLIE',
        'ALLJD','ALLKE','ALLKL','ALLPD','ALLQB','ALLSE',
        'ALLSD','ALLVD','ALLWK','ETOTAL','MASS'))
model.HistoryOutputRequest(createStepName=step1,rebar=EXCLUDE,sectionPoints=DEFAULT,
    variables = ('PCAV','CVOL'),
    name = 'UC Cavity',
    region = assembly.instances[p1_name].sets['UC RP'],
    frequency = O_freq)

# Modify Output Frequency for Explicit to Evenly-Spaced Intervals
if step_type == 'NLED':
    model.fieldOutputRequests[p1_name + ' Poa S Coords'].setValues(numIntervals = numInt,
        timeMarks=OFF)
    model.fieldOutputRequests[p1_name + ' Pia Load Input'].setValues(numIntervals = numInt,
        timeMarks=OFF)
    model.fieldOutputRequests[p1_name + ' Whole'].setValues(numIntervals = numInt, timeMarks=OFF)
    model.fieldOutputRequests['Whole'].setValues(numIntervals = numInt, timeMarks=OFF)
    model.historyOutputRequests['Whole'].setValues(numIntervals = numInt)
    model.historyOutputRequests['UC Cavity'].setValues(numIntervals = numInt)
    if Contact_or_BC == 0:
        model.historyOutputRequests[p2_name + ' RP'].setValues(numIntervals = numInt)
    elif (Contact_or_BC == 1 and BCthroughRP == 1):
        model.historyOutputRequests['F1 RP'].setValues(numIntervals = numInt)
        model.historyOutputRequests['F2 RP'].setValues(numIntervals = numInt)
        model.historyOutputRequests['F3 RP'].setValues(numIntervals = numInt)

# Deletes Auto Created Output Requests
del model.fieldOutputRequests['F-Output-1']
del model.historyOutputRequests['H-Output-1']

# Contact Stabilization
if (Contact_or_BC == 0 and (step_type == 'NLIS' or step_type == 'NLID')):
    model.interactions[cont_name].stabilizationAssignments.appendInStep(stepName = step2,
        assignments=(inst2-surfaces['Contact'], inst1-surfaces['Contact'], 'Contact Stabilization'), )

#####
#####
# Job
#####
#####

# Creates Job
mdb.Job(name=job_name, model=m_name,
    type = ANALYSIS, memory = 90, memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    queue=None, atTime=None, waitHours=0, waitMinutes=0, scratch="",userSubroutine="", description="",
    contactPrint=ON, echoPrint=ON, historyPrint=ON, modelPrint=ON,
    explicitPrecision = DOUBLE_PLUS_PACK,
    resultsFormat = ODB,
    multiprocessingMode = DEFAULT)

if parallel == 'ON':

```

```
    mdb.jobs[job_name].setValues(numGPUs = numGPUs, numCpus = numCPUs, numDomains =  
numCPUs)
```

```
# Submits Job
```

```
#~ mdb.jobs[job_name].submit(consistencyChecking=OFF)
```

```
#~ mdb.jobs[job_name].waitForCompletion()
```

APPENDIX D. LIST OF ABBREVIATIONS

The following is an alphabetized list of the abbreviations used throughout this document.

2D	two-dimensional	W/B	weight-to-buoyancy
3D	three-dimensional	ZPB	zero pressure balloons
BC	boundary conditions		
CNT	carbon nanotubes		
DoD	Department of Defense		
DOF	degrees of freedom		
FE	finite element		
FEM	finite element model		
HHT	Hilber-Hughes-Taylor		
HTA	heavier-than-air		
IE	internal energy		
ISR	intelligence, surveillance and reconnaissance		
JC	Johnson-Cook (material model)		
KE	kinetic energy		
LLDPE	linear low-density Polyethylene		
LTA	lighter-than-air		
LTAS	lighter-than-air structures		
M&S	modeling and simulation		
MD	machine direction		
MMA	Method of Moving Asymptotes		
NASA	National Aeronautics and Space Administration		
ODB	output database		
PEN	Polyethylene Naphthalate		
PET	Polyethylene Terephthalate		
RAMP	Rational Approximations of Material Properties		
SF	StratoFilm		
SIMP	Isotropic Material with Penalization		
SPB	super pressure balloons		
TD	transverse direction		
UC	unit cell		
UAV	unmanned air vehicle		
ULDB	ultra-long duration balloons		
U.S.	United States		
UVI	ultraviolet inhibitor		
VLAS	vacuum lighter-than-air structures		

Bibliography

- [1] Morris, B. A. “Strength, Stiffness, and Abuse Resistance.” *The Science and Technology of Flexible Packaging*, Elsevier, Oxford, UK, 2017, pp. 309–350.
- [2] Cranston, B., AlGhofaily, M., and Palazotto, A. “Design and Structural Analysis of Unique Structures under an Internal Vacuum.” *Aerospace Science and Technology*, Vol. 68, 2017, pp. 68–76. <https://doi.org/10.1016/j.ast.2017.04.028>.
- [3] Schwemmer, J. R., Palazotto, A. N., and Chrissis, J. “Optimal Design of a Hexakis Icosahedron Vacuum-Based Lighter-Than-Air Vehicle.” *AIAA Journal*, Vol. 56, No. 6, 2018, pp. 2131–2137. <https://doi.org/10.2514/1.J057043>.
- [4] Graves, D., Palazotto, A. N., and Moore, K. “Analysis of a Celestial Shell like Structure.” Presented at the *Scitech 2019 Forum*, AIAA, San Diego, CA, 2019.
- [5] Adorno-Rodriguez, R., and Palazotto, A. N. “Nonlinear Structural Analysis of an Icosahedron under an Internal Vacuum.” *Journal of Aircraft*, Vol. 52, No. 3, 2015, pp. 878–883. <https://doi.org/10.2514/1.C033284>.
- [6] Snyder, J. W., and Palazotto, A. N. “Finite Element Design and Modal Analysis of a Hexakis Icosahedron Frame for Use in a Vacuum Lighter-Than-Air Vehicle.” *Journal of Engineering Mechanics*, Vol. 144, No. 6, 2018. [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001462](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001462).
- [7] Cranston, B. C. *Conceptual Design, Structural Analysis, and Design Space Exploration of a Vacuum Lighter than Air Vehicle*. Ph.D. Dissertation, Aeronautics and Astronautics Dept., Air Force Institute of Technology, Wright-Patterson AFB, OH, 2016.
- [8] Jamison, L., Sommer, G., and Porche, I. R. *High-Altitude Airships for the Future Force Army*. RAND Corporation, Santa Monica, CA, 2005.
- [9] Tenenbaum, N. A Beginner’s Guide to Aerostats. *Mongabay Wiltech*. <https://wildtech.mongabay.com/2016/03/a-beginners-guide-to-aerostats/>. Accessed Mar. 27, 2019.
- [10] Zhai, H., and Euler, A. “Material Challenges for Lighter-Than-Air Systems in High Altitude Applications.” Presented at the *5th Annual Aviation Technology, Integration, and Operations, 16th Lighter-Than-Air System Technical and Balloon Systems Conferences*, AIAA, Arlington, VA, 2005.
- [11] National Research Council. *The Impact of Selling the Federal Helium Reserve*. National Academies Press, Washington, D.C., 2000.

- [12] Desjardins, J. Helium: A Valuable Gas Not To Be Taken Lightly. *Visual Capitalist*. <https://www.visualcapitalist.com/helium-a-valuable-gas-not-to-be-taken-lightly/>. Accessed Apr. 1, 2019.
- [13] US Department of the Interior. *Final List of Critical Minerals 2018*. 83 FR 23295, 2018.
- [14] Anderson, S. T. “Economics, Helium, and the U.S. Federal Helium Reserve: Summary and Outlook.” *Natural Resources Research*, Vol. 27, No. 4, 2018, pp. 455–477. <https://doi.org/10.1007/s11053-017-9359-y>.
- [15] U.S. Committee on Extension to the Standard Atmosphere. *U.S. Standard Atmosphere*. NASA, NOAA and USAF, Washington, DC, 1976.
- [16] Wakefield, D., and Bown, A. “Non-Linear Analysis of the NASA Super Pressure Balloons: Whole Flight Simulations.” Presented at the *Balloon Systems Conference*, AIAA, Denver, CO, 2017.
- [17] Sterling, J. “Material Properties Analysis of Structural Members in Pumpkin Balloons.” Presented at the *3rd Annual Aviation Technology, Integration, and Operations Forum*, AIAA, Denver, CO, 2003.
- [18] Rapid Reaction Technology Office. Summary Report of DoD Funded Lighter-Than-Air Vehicles. <https://apps.dtic.mil/dtic/tr/fulltext/u2/a568211.pdf>.
- [19] Department of Homeland Security. CBP [Customs and Border Protection]’s Eyes in the Sky. <https://www.cbp.gov/frontline/frontline-november-aerostats>. Accessed Mar. 30, 2019.
- [20] Burgess, R. Navy Retires MZ-3A Blimp. *Seapower*. <http://seapowermagazine.org/stories/20170726-MZ3A.html>. Accessed Mar. 30, 2019.
- [21] Rees, M. MQ-9 Reaper UAS Selected by Royal Netherlands Defence Force. *Unmanned Systems Technology*. <https://www.unmannedsystemstechnology.com/2018/07/mq-9-reaper-uas-selected-by-royal-netherlands-defence-force/>. Accessed Mar. 31, 2019.
- [22] US Air Force. MQ-9 Reaper. *Air Force Fact Sheets*. <https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104470/mq-9-reaper/>.
- [23] Zell, H. Types of Balloons. <https://www.nasa.gov/scientific-balloons/types-of-balloons>. Accessed Apr. 1, 2019.
- [24] Loon LLC. Journey. <https://loon.co/journey/>. Accessed Apr. 8, 2019.

- [25] Cathey, H. “Evolution of the NASA Ultra Long Duration Balloon.” Presented at the *AIAA Balloon Systems Conference, Williamsburg, VA*, AIAA, Williamsburg, VA, 2007.
- [26] Cathey, H. M. “Development Overview of the Revised NASA Ultra Long Duration Balloon.” *Advances in Space Research*, Vol. 42, No. 10, 2008, pp. 1624–1632. <https://doi.org/10.1016/j.asr.2007.03.026>.
- [27] Pagitz, M., and Pellegrino, S. “Shape Optimization of ‘Pumpkin’ Balloons.” Presented at the *AIAA Balloon Systems Conference, Williamsburg, VA*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2007.
- [28] Gerngross, T., and Pellegrino, S. “Implementation and Validation of Schapery-Rand Anisotropic Viscoelasticity Model for Super-Pressure Balloons.” Presented at the *AIAA Balloon Systems Conference, Williamsburg, VA*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2007.
- [29] Rand, J. “A Constitutive Equation for Balloon Film.” Presented at the *AIAA Balloon Systems Conference, Williamsburg, VA*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2007.
- [30] Kwok, K., and Pellegrino, S. “Large Strain Viscoelastic Model for Balloon Film.” Presented at the *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Virginia Beach, VA*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2011.
- [31] Young, L., Garde, G., and Sterling, W. “A Practical Approach for Scientific Balloon Film Strain Measurement Using Photogrammetry.” Presented at the *AIAA Balloon Systems Conference, Williamsburg, VA*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2007.
- [32] Wakefield, D., and Bown, A. “Non-Linear Analysis of the NASA Super Pressure Balloons: Refined Modelling.” Presented at the *AIAA Balloon Systems (BAL) Conference, Daytona Beach, FL*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2013.
- [33] Bown, A., and Wakefield, D. “Non-Linear Analysis of the NASA Super Pressure Balloons: Implementation of a Large Strain Material Model.” Presented at the *AIAA Balloon Systems Conference*, American Institute of Aeronautics and Astronautics, Dallas, TX, 2015.
- [34] Cathey, H. M., Tuttle, J. W., Fairbrother, D. A., and Said, M. A. “Qualification of the NASA Super Pressure Balloon.” Presented at the *AIAA Balloon Systems Conference, Dallas, TX*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2015.

- [35] Cathey, H. M., and L. Pierce, D. “Development of the NASA Ultra-Long Duration Balloon.” Presented at the *NASA Science Technology Conference, Adelphi, MD*, Adelphi, Maryland, 2007.
- [36] Bestaoui, Y. *Lighter than Air Robots*. Springer, 2012.
- [37] Crowe, C. T., Schwarzkopf, J. D., Sommerfeld, M., and Tsuji, Y. *Multiphase Flows with Droplets and Particles*. CRC Press, Boca Raton, 2011.
- [38] Congressional Budget Office. Recent Development Efforts for Military Airships. <https://fas.org/irp/program/collect/cbo-airship.pdf>.
- [39] The MathWorks Inc. Atmoscoesa Function. <https://www.mathworks.com/help/aerotbx/ug/atmoscoesa.html#br4ztgw-9>.
- [40] Fairbrother, D. A. “The NASA Balloon Program - Positioning For the Future.” Presented at the *AIAA Balloon Systems Conference, Dallas, TX*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2015.
- [41] Fairbrother, D. Super Pressure Balloons. *National Aeronautics and Space Administration*. <https://sites.wff.nasa.gov/code820/spb.html>. Accessed Apr. 7, 2019.
- [42] Jones, J., and Yavrouian, A. Meshed-Pumpkin Super-Pressure Balloon Design. <https://www.techbriefs.com/component/content/article/tb/techbriefs/materials/1197>.
- [43] Nakashino, K., Saito, Y., Goto, K., Akita, D., Matsuo, T., Matsushima, K., Hashimoto, H., and Shimadu, S. “Super Pressure Balloon with Diamond-Shaped Net: A Numerical Study of Its Structural Characteristics.” Presented at the *4th AIAA Spacecraft Structures Conference, Grapevine, TX*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2017.
- [44] Gerngross, T., Xu, Y., and Pellegrino, S. “Viscoelastic Behaviour of Pumpkin Balloons.” *Advances in Space Research*, Vol. 42, No. 10, 2008, pp. 1683–1690. <https://doi.org/10.1016/j.asr.2007.03.093>.
- [45] Bosi, F., and Pellegrino, S. “Nonlinear Thermomechanical Response and Constitutive Modeling of Viscoelastic Polyethylene Membranes.” *Mechanics of Materials*, Vol. 117, 2018, pp. 9–21. <https://doi.org/10.1016/j.mechmat.2017.10.004>.
- [46] Li, J., Kwok, K., and Pellegrino, S. “Thermoviscoelastic Models for Polyethylene Thin Films.” *Mechanics of Time-Dependent Materials*, Vol. 20, No. 1, 2016, pp. 13–43. <https://doi.org/10.1007/s11043-015-9282-8>.

- [47] MatWeb LLC. Overview of Materials for Linear Low Density Polyethylene (LLDPE), Film Grade.
- [48] Sterling, J., and Fairbrother, D. "Laboratory Analysis of Polymer Thin Films for Planetary Balloons and Gossamer Structures." Presented at the *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Palm Springs, CA, 2004.
- [49] Shames, I., and Cozzarelli, F. *Elastic and Inelastic Stress Analysis*. Taylor & Francis, Philadelphia, PA, 1997.
- [50] MatWeb LLC. Overview of Materials for PEN-PET.
- [51] Han, Y., Zhang, X., Yu, X., Zhao, J., Li, S., Liu, F., Gao, P., Zhang, Y., Zhao, T., and Li, Q. "Bio-Inspired Aggregation Control of Carbon Nanotubes for Ultra-Strong Composites." *Scientific Reports*, Vol. 5, No. 1, 2015, p. 11533. <https://doi.org/10.1038/srep11533>.
- [52] Metlen, T. T. *Design of a Lighter Than Air Vehicle That Achieves Positive Buoyancy in Air Using a Vacuum*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2013.
- [53] Adorno-Rodriguez, R. *Nonlinear Structural Analysis of an Icosahedron and Its Application to Lighter Than Air Vehicles Under a Vacuum*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2014.
- [54] Moore, K. *Quasi-Static Nonlinear Analysis of a Celestial Icosahedron Shaped Vacuum Lighter Than Air Vehicle*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2018.
- [55] AlGhofaily, M. *Finite Element Analysis and Experimentation of an Icosahedron Frame under Compression*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2015.
- [56] Just, L. *Dynamic Response Analysis of an Icosahedron Shaped Lighter Than Air Vehicle*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2015.
- [57] Castello, A. *A Study of the Hexakis Vacuum Lighter Than Air Vehicle and the Effects of Air Evacuation on the Structural Integrity*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2018.
- [58] Simulia Corp. *Abaqus Theory Guide*. Dassault Systèmes, Providence, RI, 2016.
- [59] Simulia Corp. *Abaqus Analysis User's Guide*. Dassault Systèmes, Providence, RI, 2016.

- [60] Simulia Corp. *Abaqus User Subroutines Reference Guide*. Dassault Systèmes, Providence, RI, 2016.
- [61] Boulbes, R. J. *Troubleshooting Finite-Element Modeling with Abaqus*. Springer International Publishing, Cham, Switzerland, 2020.
- [62] Belytschko, T., Liu, W. K., Moran, B., and Elkhodary, K. I. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, West Sussex, UK, 2014.
- [63] Miller, K., Joldes, G., Lance, D., and Wittek, A. “Total Lagrangian Explicit Dynamics Finite Element Algorithm for Computing Soft Tissue Deformation.” *Communications in Numerical Methods in Engineering*, Vol. 23, No. 2, 2007, pp. 121–134. <https://doi.org/https://doi.org/10.1002/cnm.887>.
- [64] Bažant, Z. P., and Vorel, J. “Energy-Conservation Error Due to Use of Green–Naghdi Objective Stress Rate in Commercial Finite-Element Codes and Its Compensation.” *Journal of Applied Mechanics*, Vol. 81, No. 2, 2013. <https://doi.org/10.1115/1.4024411>.
- [65] Ji, W., Waas, A. M., and Bažant, Z. P. “Errors Caused by Non-Work-Conjugate Stress and Strain Measures and Necessary Corrections in Finite Element Programs.” *Journal of Applied Mechanics*, Vol. 77, No. 4, 2010. <https://doi.org/10.1115/1.4000916>.
- [66] Ji, W., Waas, A. M., and Bažant, Z. P. “On the Importance of Work-Conjugacy and Objective Stress Rates in Finite Deformation Incremental Finite Element Analysis.” *Journal of Applied Mechanics*, Vol. 80, No. 4, 2013. <https://doi.org/10.1115/1.4007828>.
- [67] Bažant, Z. P., Gattu, M., and Vorel, J. “Work Conjugacy Error in Commercial Finite-Element Codes: Its Magnitude and How to Compensate for It.” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 468, No. 2146, 2012, pp. 3047–3058. <https://doi.org/10.1098/rspa.2012.0167>.
- [68] Vorel, J., and Bažant, Z. P. “Review of Energy Conservation Errors in Finite Element Softwares Caused by Using Energy-Inconsistent Objective Stress Rates.” *Advances in Engineering Software*, Vol. 72, 2014, pp. 3–7. <https://doi.org/https://doi.org/10.1016/j.advengsoft.2013.06.005>.
- [69] Negrut, D., Rampalli, R., Ottarsson, G., and Sajdak, A. “On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential-Algebraic Equations of Multibody Dynamics (DETC2005-85096).” *Journal of Computational and Nonlinear Dynamics*, Vol. 2, No. 1, 2006, pp. 73–85.

- [70] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L. “Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics.” *Earthquake Engineering & Structural Dynamics*, Vol. 5, No. 3, 1977, pp. 283-292. <https://doi.org/10.1002/eqe.4290050306>.
- [71] Novak, K. *Numerical Methods for Scientific Computing*. 2017.
- [72] Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 2002.
- [73] Bendsoe, M., and Sigmund, O. *Topology Optimization - Theory, Methods, and Applications*. Springer, 2003.
- [74] SIMULIA. Topology and Shape Optimization with Abaqus. <http://www.simulia.com/download/rum11/GL/Sandeep-Urankar-ATOM-SGL-RUM-2011.pdf>.
- [75] Sigmund, O., and Clausen, P. M. “Topology Optimization Using a Mixed Formulation: An Alternative Way to Solve Pressure Load Problems.” *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, Nos. 13–16, 2007, pp. 1874–1889. <https://doi.org/10.1016/j.cma.2006.09.021>.
- [76] Bakhtiary, Allinger, Sauter, Muller, Puchinger, Friedrich, and Mulfinger. “A New Approach for Sizing, Shape and Topology Optimization.” Detroit, 1996.
- [77] Bažant, Z. P. “Structural Stability.” *International Journal of Solids and Structures*, Vol. 37, Nos. 1–2, 2000, pp. 55–67. [https://doi.org/10.1016/S0020-7683\(99\)00078-5](https://doi.org/10.1016/S0020-7683(99)00078-5).
- [78] J. Hoff, N. *The Perplexing Behavior of Thin Circular Cylindrical Shells in Axial Compression*. 1966.
- [79] Sumirin, S., Nuroji, N., and Besari, S. *Snap-Through Buckling Problem of Spherical Shell Structure*. 2014.
- [80] M, R., and H, S. “Field Study and Evaluation of Buckling Behaviour of Steel Tanks under Geometric Imperfections.” *Journal of Applied Mechanical Engineering*, Vol. 06, No. 03, 2017. <https://doi.org/10.4172/2168-9873.1000268>.
- [81] Simitses, G. J. “Buckling and Postbuckling of Imperfect Cylindrical Shells: A Review.” *Applied Mechanics Reviews*, Vol. 39, No. 10, 1986, pp. 1517-1524. <https://doi.org/10.1115/1.3149506>.
- [82] Kobayashi, T., and Mihara, Y. “Application of Abaqus for Practical Postbuckling Analyses of Cylindrical Shells under Axial Compression.” Presented at the *In Proc., SIMULIA Customer Conference*, Dassault Systemes, Providence, RI, 2010.

- [83] Hernandez-Buentello, R. G. *3D Finite Element Modeling of Sliding Wear*. PhD Dissertation, Aeronautics and Astronautics, Air Force Institute of Technology, 2013.
- [84] Eringen, A. C., and Suhubi, E. S. “Nonlinear Theory of Simple Micro-Elastic Solids—I.” *International Journal of Engineering Science*, Vol. 2, No. 2, 1964, pp. 189–203. [https://doi.org/10.1016/0020-7225\(64\)90004-7](https://doi.org/10.1016/0020-7225(64)90004-7).
- [85] Raghu, P., Preethi, K., Rajagopal, A., and Reddy, J. N. “Nonlocal Third-Order Shear Deformation Theory for Analysis of Laminated Plates Considering Surface Stress Effects.” *Composite Structures*, Vol. 139, 2016, pp. 13–29. <https://doi.org/10.1016/j.compstruct.2015.11.068>.
- [86] Hütter, G. “Homogenization of a Cauchy Continuum towards a Micromorphic Continuum.” *Journal of the Mechanics and Physics of Solids*, Vol. 99, 2017, pp. 394–408. <https://doi.org/10.1016/j.jmps.2016.09.010>.
- [87] Eringen, A. C., and Suhubi, E. S. “Nonlinear Theory of Simple Micro-Elastic Solids—II.” *International Journal of Engineering Science*, Vol. 2, No. 2, 1964, pp. 189–203. [https://doi.org/10.1016/0020-7225\(64\)90004-7](https://doi.org/10.1016/0020-7225(64)90004-7).
- [88] L. B. Freund, S. S. *Thin Film Materials: Stress , Defect Formation and Surface Evolution*. 2003.
- [89] Wang, G., and Li, X. “Predicting Young’s Modulus of Nanowires from First-Principles Calculations on Their Surface and Bulk Materials.” *Journal of Applied Physics*, Vol. 104, No. 11, 2008, p. 113517. <https://doi.org/10.1063/1.3033634>.
- [90] Abadias, G., Chason, E., Keckes, J., Sebastiani, M., Thompson, G. B., Barthel, E., Doll, G. L., Murray, C. E., Stoessel, C. H., and Martinu, L. “Review Article: Stress in Thin Films and Coatings: Current Status, Challenges, and Prospects.” *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, Vol. 36, No. 2, 2018, p. 020801. <https://doi.org/10.1116/1.5011790>.
- [91] Huang, R., Stafford, C. M., and Vogt, B. D. “Wrinkling of Ultrathin Polymer Films.” *MRS Proceedings*, Vol. 924, 2006, pp. 0924-Z04-10. <https://doi.org/10.1557/PROC-0924-Z04-10>.
- [92] Patil, A., Kolhe, A., and Shaikh, A. S. A. “Review of Buckling in Various Structures Like Plates and Shells.” *International Journal of Research in Engineering and Technology*, Vol. 3, No. 4, 2014, pp. 396–402. <https://doi.org/10.15623/ijret.2014.0304072>.

- [93] Amabili, M. “Nonlinear Stability of Circular Cylindrical Shells under Static and Dynamic Axial Loads.” *Nonlinear Mechanics of Shells and Plates in Composite, Soft and Biological Materials*, Cambridge University Press, New York, NY, 2018, pp. 542–555.
- [94] Teng, J. G., and Rotter, J. M. “Buckling of Imperfect Cylinders.” *Buckling of Thin Metal Shells*, Taylor & Francis, Abingdon, UK, 2004, pp. 52–57.
- [95] Murugesan, M., and Jung, D. “Johnson Cook Material and Failure Model Parameters Estimation of AISI-1045 Medium Carbon Steel for Metal Forming Applications.” *Materials*, Vol. 12, No. 4, 2019, p. 609. <https://doi.org/10.3390/ma12040609>.
- [96] Johnson, G. R., and Cook, W. H. “Fracture Characteristics of Three Metals Subjected to Various Strains, Strain Rates, Temperatures and Pressures.” *Engineering Fracture Mechanics*, Vol. 21, No. 1, 1985, pp. 31–48. [https://doi.org/10.1016/0013-7944\(85\)90052-9](https://doi.org/10.1016/0013-7944(85)90052-9).
- [97] He, A., Xie, G., Zhang, H., and Wang, X. “A Comparative Study on Johnson–Cook, Modified Johnson–Cook and Arrhenius-Type Constitutive Models to Predict the High Temperature Flow Stress in 20CrMo Alloy Steel.” *Materials & Design*, Vol. 52, 2013, pp. 677–685. <https://doi.org/10.1016/j.matdes.2013.06.010>.
- [98] Huang, Z., Gao, L., Wang, Y., and Wang, F. “Determination of the Johnson-Cook Constitutive Model Parameters of Materials by Cluster Global Optimization Algorithm.” *Journal of Materials Engineering and Performance*, Vol. 25, No. 9, 2016, pp. 4099–4107. <https://doi.org/10.1007/s11665-016-2178-1>.
- [99] Poliotti, M., Möller, O., Ascheri, J. P., and Sierra, P. L. “Implementation of a Unified Library of Solvers for Nonlinear Structural Analysis.” *Mecánica Computacional*, Vol. 35, 2017, pp. 2287–2299.
- [100] Hughes, T. J. R. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY, 2000.
- [101] Fung, T. C. “Numerical Dissipation in Time-Step Integration Algorithms for Structural Dynamic Analysis.” *Progress in Structural Engineering and Materials*, Vol. 5, No. 3, 2003, pp. 167–180. <https://doi.org/10.1002/pse.149>.
- [102] Dinev, D., Liu, T., and Kavan, L. “Stabilizing Integrators for Real-Time Physics.” *ACM Transactions on Graphics*, Vol. 37, No. 1, 2018, pp. 9:1–9:19. <https://doi.org/10.1145/3153420>.

- [103] Morgan, D., and Qiao, S. “Analysis of Damped Mass-Spring Systems for Sound Synthesis.” *EURASIP Journal on Audio, Speech, and Music Processing*, Vol. 2009, No. 1, 2009, p. 947823. <https://doi.org/10.1155/2009/947823>.
<https://doi.org/10.1155/2009/947823>.
- [104] Katopodes, N. D. “Basic Concepts.” *Free-Surface Flow*, Elsevier, Cambridge, MA, 2019, pp. 22–29.
- [105] Leondes, C. T. “Techniques in Control-Oriented Reduction of Finite Element Modeling in Structural Dynamic Systems.” *Structural Dynamic Systems Computational Techniques and Optimization: Finite Element Analysis Techniques, Volume 2*, Amsterdam, Netherlands: CRC Press, 1999, p. 29.
- [106] Knight, N. F., Macy, S. C., and McCleary, S. L. “Assessment of Structural Analysis Technology for Static Collapse of Elastic Cylindrical Shells.” *Finite Elements in Analysis and Design*, Vol. 18, No. 4, 1995, pp. 403–431. [https://doi.org/10.1016/0168-874X\(94\)00066-O](https://doi.org/10.1016/0168-874X(94)00066-O).
- [107] Yoshimura, Y. “On the Mechanism of Buckling of a Circular Cylindrical Shell under Axial Compression.” *National Advisory Committee for Aeronautics, Technical Memorandum 1390*, 1955.
- [108] Gurtin, M. E., and Ian Murdoch, A. “A Continuum Theory of Elastic Material Surfaces.” *Archive for Rational Mechanics and Analysis*, Vol. 57, No. 4, 1975, pp. 291–323. <https://doi.org/10.1007/BF00261375>.
- [109] Gurtin, M. E., and Ian Murdoch, A. “Surface Stress in Solids.” *International Journal of Solids and Structures*, Vol. 14, No. 6, 1978, pp. 431–440. [https://doi.org/10.1016/0020-7683\(78\)90008-2](https://doi.org/10.1016/0020-7683(78)90008-2).
- [110] Sheydakov, D. N. “Effect of Surface Stresses on Stability of Elastic Circular Cylinder.” *Advances in Mechanics of Microstructured Media and Structures*, Springer International Publishing, Cham, Switzerland, 2018, pp. 343–356.
- [111] Altenbach, H., Eremeyev, V. A., and Lebedev, L. P. “Mathematical Study of Boundary-Value Problems of Linear Elasticity with Surface Stresses.” *Surface Effects in Solid Mechanics*, Springer, Berlin/Heidelberg, Germany, 2013, pp. 1–19.
- [112] Chen, S., and Yao, Y. “Elastic Theory of Nanomaterials Based on Surface-Energy Density.” *Journal of Applied Mechanics*, Vol. 81, No. 12, 2014, p. 121002. <https://doi.org/10.1115/1.4028780>.
- [113] Müller, P., Saül, A., and Leroy, F. “Simple Views on Surface Stress and Surface Energy Concepts.” *Advances in Natural Sciences: Nanoscience and Nanotechnology*, Vol. 5, No. 1, 2013, p. 013002. <https://doi.org/10.1088/2043-6262/5/1/013002>.

- [114] Schwemmer, J. *Optimal Design of a Hexakis Icosahedron Vacuum Based Lighter Than Air Vehicle*. MS Thesis, Aeronautics and Astronautics, Air Force Institute of Technology, 2017.
- [115] Pilkey, W. D. “Thin Shells.” *Formulas for Stress, Strain, and Structural Matrices*, 2004, pp. 1185–1318.
- [116] MatWeb LLC. DuPont Teijin Films Mylar HP Polyester Film, 48 Gauge.
- [117] MatWeb LLC. DIAB ProBalsa PB Standard Balsa Wood.
<http://www.matweb.com/search/DataSheet.aspx?MatGUID=f871f55a8fc7460db37ad8cb39bd3945&ckck=1>.
- [118] Wakefield, D. “Numerical Investigations of Pumpkin Balloon Deployment Stability.” Presented at the *AIAA Balloon Systems Conference*, AIAA, Williamsburg, VA, 2007.
- [119] Hüttig, C., and Stemmer, K. “The Spiral Grid: A New Approach to Discretize the Sphere and Its Application to Mantle Convection.” *Geochemistry, Geophysics, Geosystems*, Vol. 9, No. 2, 2008, p. n/a-n/a.
<https://doi.org/10.1029/2007GC001581>.
- [120] Adorno, R., and Palazotto, A. N. “Evaluation of an Air-Stiffened Toroidal Sphere Subjected to Pressure.” *AIAA Journal*, 2020, pp. 1–6.
<https://doi.org/10.2514/1.J059893>.
- [121] Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J. “Nonlinearity: An Introduction.” *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, West Sussex, UK, 2002, pp. 595–629.
- [122] Adorno, R., and Palazotto, A. N. “Towards Feasible Near-Vacuum Lighter-than-Air Envelopes via Gossamer Structures.” Presented at the *Scitech 2020 Forum*, AIAA, Orlando, FL, 2020.
- [123] Munson, B., Young, D., and Okiishi, T. “Compressibility of Fluids.” *Fundamentals Of Fluid Mechanics*, John Wiley & Sons, Hoboken, NJ, 2006, p. 20.
- [124] Smith, J. M. “Volumetric Properties of Pure Fluids.” *Introduction to Chemical Engineering Thermodynamics*, McGraw-Hill, New York, NY, 2005, pp. 64–104.
- [125] Perry, R. H. G., Green, D. W., and Maloney, J. O. “Physical and Chemical Data.” *Perry’s Chemical Engineers’ Handbook*, McGraw-Hill, New York, NY, 1997, pp. 2–140.

- [126] Bouzidi, R., Ravaut, Y., and Wielgosz, C. “Finite Elements for 2D Problems of Pressurized Membranes.” *Computers & Structures*, Vol. 81, No. 26, 2003, pp. 2479–2490. [https://doi.org/https://doi.org/10.1016/S0045-7949\(03\)00308-0](https://doi.org/https://doi.org/10.1016/S0045-7949(03)00308-0).
- [127] Coelho, M., Roehl, D., and Bletzinger, K.-U. “Numerical and Analytical Solutions with Finite Strains for Circular Inflated Membranes Considering Pressure–Volume Coupling.” *International Journal of Mechanical Sciences*, Vol. 82, 2014, pp. 122–130. <https://doi.org/https://doi.org/10.1016/j.ijmecsci.2014.03.012>.
- [128] Bonet, J., Wood, R. D., Mahaney, J., and Heywood, P. “Finite Element Analysis of Air Supported Membrane Structures.” *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 5, 2000, pp. 579–595. [https://doi.org/10.1016/S0045-7825\(99\)00428-4](https://doi.org/10.1016/S0045-7825(99)00428-4).
- [129] Rumpel, T., and Schweizerhof, K. “Volume-Dependent Pressure Loading and Its Influence on the Stability of Structures.” *International Journal for Numerical Methods in Engineering*, Vol. 56, No. 2, 2003, pp. 211–238. <https://doi.org/10.1002/nme.561>.
- [130] Rumpel, T., and Schweizerhof, K. “Hydrostatic Fluid Loading in Non-Linear Finite Element Analysis.” *International Journal for Numerical Methods in Engineering*, Vol. 59, No. 6, 2004, pp. 849–870. <https://doi.org/10.1002/nme.892>.
- [131] Zielke, G. “Inversion of Modified Symmetric Matrices.” *Journal of the Association for Computing Machinery*, Vol. 15, No. 3, 1968, pp. 402–408. <https://doi.org/10.1145/321466.321472>. <https://doi.org/10.1145/321466.321472>.
- [132] Weisstein, E. Torus. <http://mathworld.wolfram.com/Torus.html>.
- [133] Sharma, A. K., and Bhattacharya, B. “Parameter Estimation of Butyl Rubber Aided with Dynamic Mechanical Analysis for Numerical Modelling of an Air-Inflated Torus and Experimental Validation Using 3D-Laser Doppler Vibrometer.” *Journal of Low Frequency Noise, Vibration and Active Control*, Vol. 38, No. 2, 2019, pp. 296–311. <https://doi.org/10.1177/1461348419825685>. <https://doi.org/10.1177/1461348419825685>.
- [134] Pina, J., Costa, A., and Appleton, J. “Formal Solution of Quasi-Static Problems.” *International Journal of Non-Linear Mechanics*, Vol. 45, No. 5, 2010, pp. 525–534. <https://doi.org/10.1016/j.ijnonlinmec.2010.02.002>.
- [135] Harris, J. W., and Stöcker, H. “Spherical Segment (Spherical Cap).” *Handbook of Mathematics and Computational Science*, Springer, 1998, p. 107.

- [136] VAC AERO International. An Introduction to Vacuum Pumps. <https://vacaero.com/information-resources/vacuum-pump-technology-education-and-training/1039-an-introduction-to-vacuum-pumps.html>. Accessed Apr. 22, 2019.
- [137] Hooke, R., Martyn, J., Pitt, M., and Chiswell, R. *Philosophical Collections: Containing an Account of Such Physical, Anatomical, Chymical, Mechanical, Astronomical, Optical, Or Other Mathematical and Philosophical Experiments and Observations as Have Lately Come to the Publishers Hands, as Also an Accoun.* The Royal Society, 1679.
- [138] Adorno-Rodriguez, R., and Palazotto, A. N. “Nonlinear Structural Analysis of an Icosahedron Under an Internal Vacuum.” *Journal of Aircraft*, Vol. 52, No. 3, 2015, pp. 878–883. <https://doi.org/10.2514/1.C033284>.
- [139] Metlen, T. T., Palazotto, A. N., and Cranston, B. “Economic Optimization of Cargo Airships.” *CEAS Aeronautical Journal*, Vol. 7, No. 2, 2016, pp. 287–298. <https://doi.org/10.1007/s13272-016-0188-1>.
- [140] Castello, A., and Palazotto, A. N. “Structural Analysis of the Hexakis Icosahedron Lighter Than Air Vehicle and the Air Evacuation System.” Presented at the *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, Kissimmee, FL, 2018.
- [141] Andrey, A. V. G., and Akhmeteli, M. Layered Shell Vacuum Balloons, 11/127,613, 2005.
- [142] Luciano, D., Anthony, P., and Enrico, S. “Starred Polyhedral Shell Reinforced with Internal Pockets Considering an Internal Vacuum.” *Journal of Engineering Mechanics*, Vol. 145, No. 9, 2019, p. 4019064. [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001637](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001637).
- [143] El-sayed, H. M., Lotfy, M., El-din Zohny, H. N., and Riad, H. S. “Prediction of Fatigue Crack Initiation Life in Railheads Using Finite Element Analysis.” *Ain Shams Engineering Journal*, Vol. 9, No. 4, 2018, pp. 2329–2342. <https://doi.org/https://doi.org/10.1016/j.asej.2017.06.003>.
- [144] Block, J. M., and Keer, L. M. “Partial Plane Contact of an Elastic Curved Beam Pressed by a Flat Surface.” *Journal of Tribology*, Vol. 129, No. 1, 2006, pp. 60–64. <https://doi.org/10.1115/1.2401212>.
- [145] Baker, W. E. “Axisymmetric Modes of Vibration of Thin Spherical Shell.” *The Journal of the Acoustical Society of America*, Vol. 33, No. 12, 1961, pp. 1749–1758. <https://doi.org/10.1121/1.1908562>.

- [146] Simulia Corp. *Abaqus Benchmarks Guide*. Dassault Systèmes, Providence, RI, 2016.
- [147] Moore, E. Three Types of Buoyancy. *Sciencing*. <https://sciencing.com/three-types-buoyancy-10036718.html>. Accessed Mar. 20, 2019.
- [148] ASTM International. ASTM D4850-13(2017)E1 Standard Terminology Relating to Fabrics and Fabric Test Methods. <https://doi.org/10.1520/D4850-13R17E01>.
- [149] ANSYS GRANTA. Material and Process Selection Charts GRANTA CES 2010 Edupack. 5. https://downloadfiles.grantadesign.com/pdf/teaching_resource_books/2-Materials-Charts-2010.pdf. Accessed Mar. 12, 2019.
- [150] NASA. Earth Atmosphere Model. <https://www.grc.nasa.gov/www/k-12/airplane/atmosmet.html>. Accessed Mar. 12, 2019.
- [151] Han, Y., Zhang, X., Yu, X., Zhao, J., Li, S., Liu, F., Gao, P., Zhang, Y., Zhao, T., and Li, Q. “Bio-Inspired Aggregation Control of Carbon Nanotubes for Ultra-Strong Composites.” *Scientific Reports*, Vol. 5, No. 1, 2015, p. 11533. <https://doi.org/10.1038/srep11533>.
- [152] Rainwater, E. ., and Smith, M. . “Ultra High Altitude Balloons for Medium-to-Large Payloads.” *Advances in Space Research*, Vol. 33, No. 10, 2004, pp. 1648–1652. <https://doi.org/10.1016/j.asr.2003.07.037>.
- [153] Bally Ribbon Mills. Spectra® and Dyneema®. <https://www.ballyribbon.com/fibers/spectra-dyneema/>. Accessed Apr. 10, 2019.
- [154] Blåsjö, V. “The Evolution of the Isoperimetric Problem.” *The American Mathematical Monthly*, Vol. 112, 2006, pp. 526–566.
- [155] American Vacuum Society. Technical Resources: Glossary. <https://web.archive.org/web/20060304160701/http://www.aip.org/avsguide/refguide/glossary.html#v>. Accessed Apr. 22, 2019.
- [156] Cole-Parmer Instrument Company. Gast High-Capacity Vacuum/Pressure Pumps. <https://www.coleparmer.com/p/gast-high-capacity-vacuum-pressure-pumps/6919>. Accessed Apr. 22, 2019.
- [157] Wriggers, P., and Zavarise, G. “Computational Contact Mechanics.” *Encyclopedia of Computational Mechanics*, John Wiley & Sons, Chichester, UK, 2004, pp. 196–226.

- [158] Popko, E. S. “Subdivision Schemas.” *Divided Spheres: Geodesics and the Orderly Subdivision of the Sphere*, A K Peters/CRC Press, Boca Raton, FL, 2012, pp. 191–247.
- [159] Wenninger, M. “Geodesic Domes.” *Spherical Models*, Dover Publications, Mineola, NY, 1999, pp. 80–124.
- [160] Wolfram Research Inc. “Geodesate Function.” *Wolfram Language and System Documentation Center*, Wolfram Research Inc., 2020.
- [161] Grosshans, H., Gopireddy, S. R., Humza, R. M., and Gutheil, E. “Modeling and Simulation of Single Particle and Spray Drying of PVP- and Mannitol-Water in Hot Air BT - Process-Spray: Functional Particles Produced in Spray Processes.” (U. Fritsching, ed.), Springer International Publishing, Cham, 2016, pp. 309–339.
- [162] Lecieux, Y., and Bouzidi, R. “Experimental Analysis on Membrane Wrinkling under Biaxial Load – Comparison with Bifurcation Analysis.” *International Journal of Solids and Structures*, Vol. 47, No. 18, 2010, pp. 2459–2475.
<https://doi.org/https://doi.org/10.1016/j.ijsolstr.2010.05.005>.
- [163] Wielgosz, C., and Thomas, J.-C. “Deflections of Inflatable Fabric Panels at High Pressure.” *Thin-Walled Structures*, Vol. 40, No. 6, 2002, pp. 523–536.
[https://doi.org/https://doi.org/10.1016/S0263-8231\(02\)00010-1](https://doi.org/https://doi.org/10.1016/S0263-8231(02)00010-1).
- [164] Liu, M., Huang, J., and Liu, M. “Wrinkling Reduction of Membrane Structure by Trimming Edges.” *AIP Advances*, Vol. 7, No. 5, 2017, p. 55116.
<https://doi.org/10.1063/1.4984289>.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 25-03-2021		2. REPORT TYPE Dissertation		3. DATES COVERED (From - To) September 2017 - March 2021	
4. TITLE AND SUBTITLE Design and Analysis of Air-Stiffened Vacuum Lighter-Than-Air Structures				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Captain Ruben Adorno-Rodriguez				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENY-DS-21-M-285	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 875 N. Randolph, Ste.325 Arlington Virginia, 22203				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A: Approved for Public Release; Distribution Unlimited.					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT Airships, aerostats and balloons are all part of the lighter-than-air (LTA) family of systems, which uses Archimedes principle to achieve buoyancy by replacing an air volume with LTA gases. These gases stiffen the otherwise compliant structures, allowing them to sustain the pressure difference brought by the displaced air. The compliance of these structures is a byproduct of the weight requirement, materials and geometry of which these are built from, typically resulting in dimensionalities that exhibit low or virtually non-existent bending stiffness. The former has constrained the development of LTA structures that utilize a partial vacuum, rather than a lifting gas, to achieve buoyancy. This research presents the development and structural characterization of air stiffened designs, which utilize air to shape a set of envelopes. The development established a framework that enables the structural characterization of these designs under a variety of geometric and loading conditions. Such framework allowed the development of finite element solutions that included geometric, fluid-structure and contact nonlinearities. The structural characterization of the Helical Sphere and Icoron air-stiffened designs demonstrated a reduction of material modulus and strength requirements compared to membrane-over-frame designs, and showed the capability of air-stiffened designs to be tailored for					
15. SUBJECT TERMS Lighter-than-air; fluid-structure interactions; structural simulation; structural stability					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)
U	U	U	UU	275	Dr. Anthony Palazotto, AFIT/ENY (937)255-3636 x4599, anthony.palazotto@afit.edu

Left Blank