High-Fidelity Grid Generation Investigations of a Low-Density Supersonic Decelerator

a project presented to
The Faculty of the Department of Aerospace Engineering San José State University

in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Bruno C. Sotelo

December 2017

approved by

Dr. Periklis Papadopoulos
Faculty Advisor

The Designated Project Advisor(s) Approves the Thesis Titled

HIGH-FIDELITY GRID GENERATION INVESTIGATIONS OF A LOW-DENSITY SUPERSONIC DECELERATOR

By

Bruno C. Sotelo

APPROVED FOR THE DEPARTMENT OF AEROSPACE ENGIENERING

SAN JOSE STATE UNIVERSITY

DECEMBER 2017

Dr. P. Papadopoulos	Department of Aerospace Engineering Advisor

ABSTRACT

HIGH-FIDELITY GRID GENERATION INVESTIGATIONS OF A LOW-DENSITY SUPERSONIC DECELERATOR

By Bruno Sotelo

Multi-block grid generation for Computational Fluid Dynamics (CFD) is a prevalent tool for compressible flows. Over the last several decades, unstructured methods have been more adopted by commercial CFD solvers, however multi-block grids are still the higher fidelity choice, and the one that requires a significant amount of user intervention and technique. This paper motivates this by pursuing a methodical procedure to generate a structured grid. Proof of concept tasks are carried out to solve domain-specific and shape-specific aerospace problems. Then re-use the same techniques learned to address a difficult problem: entry, descent, and landing of supersonic decelerators. A meshing technique is determined and a high quality multi-block grid is computed. The result in this paper is a successful multi-block model to continue and perform compressible flow simulations.

ACKNOWLEDGMENTS

I would like to take the opportunity to thank Dr. Papadopoulos for providing me with vast resources to complete this project, as well as his guidance that led me to pursue a career in CFD. I also would like to extensively thank the engineers at Program Development Company, the creators of GridPro, for sponsoring the software license and proving me with great user support.

Personally, I want to thank my father, Jose Sotelo, and my mother, Blanca Bautista, for their unconditional love and support. To all the friends and educators throughout my K-12, college, and graduate school, this is dedicated to all of you.

Table of Contents

TABLE OF FIGURES

Chapter 1: Introduction

1.1 Background

Human kind is living a time in which Mars and Space exploration is a reality. The advancements in technology, communications, science, and engineering have all contributed to its development. Within Mars exploration itself, is a very broad subject, which contains multiple areas of research and development. Mars Entry, Descent, and Landing (EDL) is a subject of its own. Unlike Earth, Mars' atmosphere is lower in density. In 1997, the Mars Pathfinder Meteorology Experiment (MET) instruments recorded Mars's density from the upper atmosphere at 160 km above landing site in the range of 10^{-10} kg/m^3, and at the surface at approximately 1.76 X 10^{-2} kg/m^3 [SBC97], see **Figure 1.** As shown, Mar's atmosphere profile is indeed much thinner than Earth's atmosphere, and it presents a unique physical environment in all matters for Mars' Entry, Descent, and Landing.

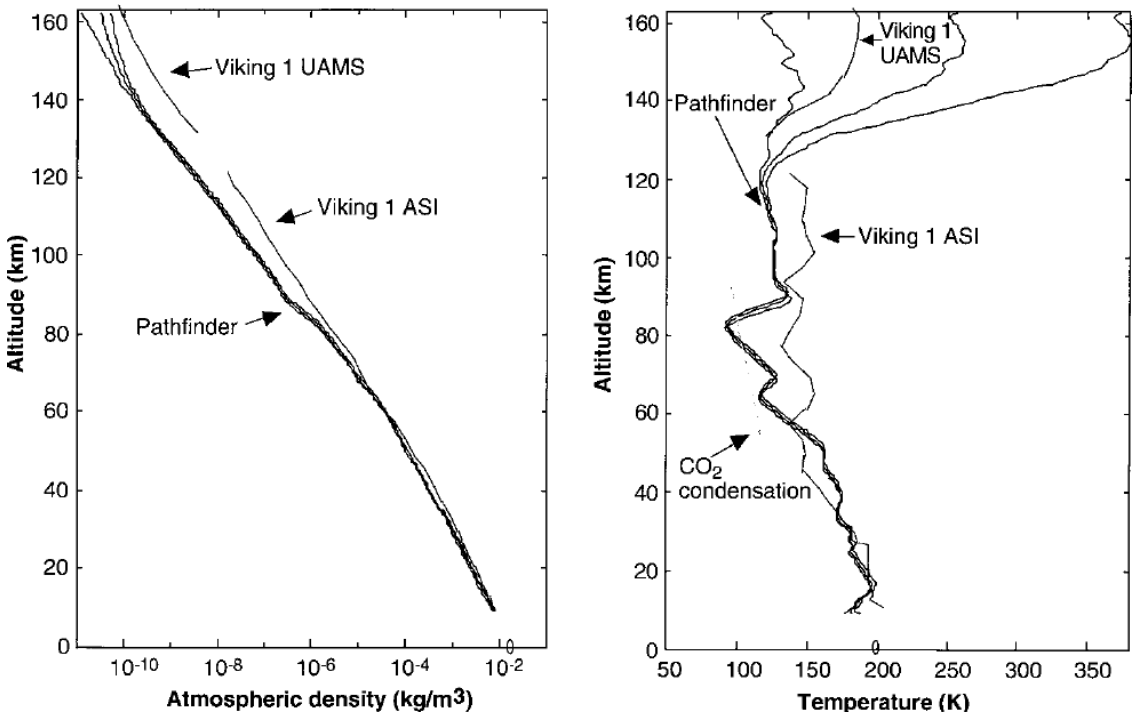


Figure 1 – Mars Atmospheric density and temperature profile

The Mars Entry, Descent, and Landing (EDL) program has the mission to bring spacecraft from a hypersonic speed in Mars' orbit to a zero-rest position on the surface. It begins when the spacecraft has completed its cruise trip to Mars' top of the atmosphere at speeds around Mach 20-25. The spacecraft then begins to feel Mars' dense atmosphere and the friction generated heats up the spacecraft heat shield to degrees Celsius temperatures in the thousands. As it descends, the outstanding aerodynamic drag in its path brings the speed to about Mach 2-2.5 before a parachute is released. This supersonic parachute further decelerates the spacecraft to a subsonic region. The heat shield then separates and finally rocket motors bring the spacecraft's speed to zero on the ground.

One of the most difficult events in this mission is the spacecraft supersonic deceleration using a parachute. A lightweight parachute which can interact with Mars' atmosphere and produce a resisting drag force that will slow down the spacecraft from supersonic to subsonic. This phenomenon is the focus of this study. As Mars exploration plans continue to expand, the need for a bigger spacecraft and heavier payload is required. This need is particularly crucial for the successful deployment, inflation, and performance of a low density supersonic parachute. Thankfully, due to major efforts in understanding the physics behind this phenomenon, scientists and engineers have achieved great success, and will continue to perform extensive research for higher amounts of payload such as more robotic rovers, and eventually humans.

Since NASA's Viking program in the 1980s, the Low Density Supersonic Decelerator has been successful. However, most of its success has been constrained to its heritance. In other words, many of the characteristics of all missions such as from Mars Pathfinder, Mars Exploration Rover programs have been kept to meet system qualifications similar to those achieved by the Viking program. This is due to the very expensive 1:1 scale experiments in order to qualify all the systems involved in EDL. Moreover, the need to first attain scientific analytical solutions to this problem has evolved such as in the development of computational physics including Computational Fluid Dynamics, Finite Element Analysis, and its coupling Fluid Structure Interaction.

This study will investigate the techniques in creating a domain similar to the capsule-parachute system in low density supersonic deceleration. This analytical and computational study will let create a high-fidelity model that will help understand the physics that describe the flow field in Mars' atmosphere and the compressible flow in deceleration from supersonic to subsonic.

1.2 Previous Work

Previous scientific investigations have developed the structural and flow physics of the low density supersonic decelerator system, more directly using computational engineering software. The results have agreed with experimental data to validate and qualify spacecraft systems prior to various missions.

The MSL supersonic parachute's specifications and features expands from previous EDL technologies flown by the Viking, Mars pathfinder, and Mars Exploration Rover [SSWRC10], see **Figure 2.**

Parameter	Value
Disk Diameter	$0.72D_o$
Reference Area (S_o)	$¼ \pi D_o^2$
Geometric Porosity (Area)	12.5%
Vent Diameter	$0.07 D_o$
Band Height	$0.121 D_o$
Gap Height	$0.042 D_o$
x/d	10
Suspension Line Length	$1.7 D_o$

Figure 2 – MSL Project Specifications

Prior to the manufacture of the parachute, computational models of the parachute were developed to research the aerodynamic loading around the parachute from deployment,

inflation, to performance. Simultaneously, preliminary designs of the parachute evolved from the feedback of computational results. As far as the parachute's material, a higher packed parachute in lesser volume has developed with newer material and packing technology. New material such as Kevlar and Teflon has improved volumetric efficiency to greater amounts. In addition, non-dimensional comparative variables such as weighted average solid density have been the benchmark to compare newer materials with old traditional designs [ZS13]. Finding increased main pack densities, thus, has improved clearance with size limits and allowed weight growth without additional volume. This has been favorable to the requirements of heavier parachutes for a bigger payload.

Characteristics, such as shape, geometry, and overall profile are thus important for computational modeling. Having established a geometry and material, the main subject can focus in the physics around the parachute. The performance of the parachute is defined and dependent of Mach number, shape, size of the capsule, distance between the capsule and the parachute, the shape and the size of the canopy, the material properties of the canopy and the cables, and the angle of attack of the capsule [PSH96]. The facts from Peterson and Strickland produce a difficult problem to solve in fluid dynamics. It's a complex situation where the solutions are coupled within the parachute flow field and the body flexible structural dynamics. Karagiozis, Kamakoti, Cirak, and Pantano have identified this as a complex fluid structure interaction problem. It involves bluff and porous body aerodynamics, nonlinear structural dynamics and fully coupled interaction between the compressible fluid flow, with shocks, and the membrane structure undergoing large deformation [KKCP11].

The computational modeling and simulation efforts have been influential in system qualification and design. Since the Viking program all, if not most, missions consisted in supersonic decelerator systems that maintained the same aerodynamics and structural characteristics. Since 1:1 scale experiments are expensive and long-term to execute, computational studies of smaller scale provide and initiate phase qualifications. This, for instance, was the case for the MSL mission in 2010. In system qualification, Dr. Gaham Candler

from the University of Minnesota provided a CFD code that let Sengupta et al. to model the flow field around a full scale rigid representation of the MSL parachute with capsule [SSWRC07], see **Figure 3.**

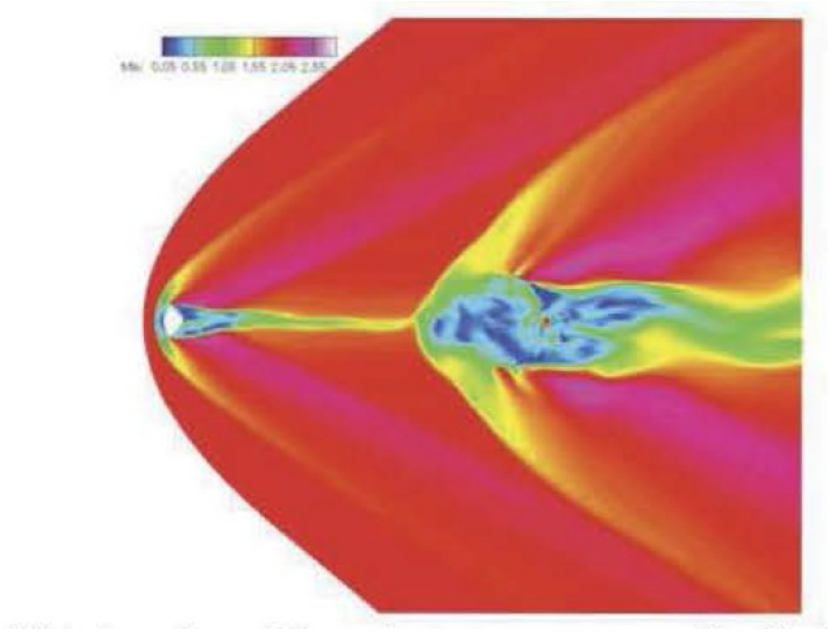


Figure 3 – Rigid Parachute CFD simulation by Sengupta et al.

Now speaking, in full terrain of computational fluid dynamics and structural dynamics, various authors have contributed relevant results to this study. Karagiozies et al. made significant contributions in describing the flow field. A detached shock (or bow shock) was seen ahead of the capsule and canopy. These shocks induced the supersonic behavior upstream of the flow field. A considerably stable turbulent wake developed behind the capsule, while an unstable turbulent wake developed behind the canopy [KKCP11]. Ross and Nebiker also pointed out that dynamic instabilities were caused by the violent clash between the turbulent wake behind the capsule and the bow shock ahead of the canopy [RN66]. Earlier in supersonic parachute development it was also found that the shape of the canopy plays an important role where the axisymmetric surface carries uniform pressure with only meridional stresses and zero hoop stresses, therefore fluid pressure is mainly carried by the suspension lines [T63].

Moreover, Heinrich pointed out that such suspension lines play a crucial role on causing system dynamics instability and flow unsteadiness [H66].

With the findings of Taylor and Heinrich, a lot of attention was put to the interaction between the capsule wake and the canopy bow shock. As stated earlier, their collapse is the driving source for the overall flow unsteadiness and system instability. In consequence, Stevens performed linear stability studies and found that all vibration frequencies were inversely proportional to the linear dimensions and their ratios can be defined in terms of the canopy mass over the suspension line-mass ratio and the elastic properties of the material [S72]. Karagozis et al. showed instants corresponding to the phases the canopy experiences, starting with the point of maximum inflation, average inflation, and minimum inflation. The average inflation showed mostly a collapse on the band, suggesting that the disk starts to collapse only after the band is experiencing strong nonlinear instabilities. Once deflated and midpoint back to inflation, the band and disk remain partially collapsed [KKCP11]. Nerem and Pake could develop parachute models and investigated the numerous effects, including damping on the dynamic pressure, the ratio of the maximum vehicle load over maximum canopy drag force, the dynamic load at supersonic conditions, inflation rates, and stability [NP73].

The convergence of computational solutions in supersonic decelerators require a high level of meshing technique due to the complex physical phenomena, and the coupling of a CFD flow field and an FEA flexible structure. Barnhardt et al. used the detached-eddy simulation approach to solve this problem, and could prove that the modeling of a flexible capsule vs. a rigid structure did not particularly make a difference in seeing the unstable bow shock upstream [BDNCG07]. This last research discovery gives this study the confidence to perform idealized rigid body models of the capsule-canopy system as first milestone.

1.3 Project Objective

This study will focus in creating a high-fidelity of the low-density supersonic decelerator phenomena. It will particularly begin with a geometric idealization of the model and perform proof of concept scenarios. The following listed points are the sequential project milestones:

- Establish the requirements to obtain a structured multi-block grid
- Proof of Concept about idealized shapes that are closely matched with the actual domain
- Construct a final structured multi-block grid around a capsule-parachute domain

1.4 Methodology

The focus of this project is to investigate how a high-fidelity compressible flow grid is generated using structured multi-blocks, as well as how this can induce to obtain accurate physics of the flow field that surround a parachute at supersonic speeds. In general, to achieve this, analytical and domain-specific studies must be performed using the fundamental equations of transport (e.g. Euler's or Navier-Stokes Equations). However, the complexity and effort required for such analytical studies often can be done supported by full performance of Computational Fluid Dynamics. Part of the reason is because exact solutions are difficult and time-consuming to achieve. At the same time, 1:1 scale experiments are expensive, and most experimental work is performed at NASA or highly-funded commercial/private entities.

With the successful development of CFD including hardware performance, open-source software development and large availability of resources that isn't anymore a significant financial burden, a good engineering problem-solving technique can adequately be designed. First, proof of concept models is to be created to understand (piece-wise) the different set up options of a structured multi-block grid. Examples of unique set up features in multi-block grids that are relevant for this research are O-grid type domains and the creation of pseudo (internal) surfaces. The proof of concepts is significant in validating the expected grid results against the

theoretical concepts and specifications of multi-block grids as documented by the GridPro software guides.

Second, it follows the complete construction of the entire domain. At this stage, it's crucial to have entered this stage of the project with a significant knowledge base. The software will provide sufficient object topology understanding. The flow field is intended to extend approximately 10 times the longest dimension of the capsule-parachute system. To conclude with the mesh stage and obtain a high-fidelity model, substantial time must be spent on understanding the object and flow field topology, GUI – to - engineer experience, and computer architecture capability. The topology will identify the main regions such as near the object's surface, the near outer boundaries, and the intermediate regions of the former two. It can be predicted that the mesh will be refined close to the object, and coarser away from the object. In other words, the mesh will be stretched, as the flow field appears closer to the outside boundaries. This is necessary as computational time and effort should be exploited only where it's necessary.

Chapter 2: Structured multi-block mesh generation

The high-fidelity connection to the knowledge base is made possible using a structured multi-block mesh generation software. This report, and this chapter will cover the techniques that were employed. GridPro is a widely popular, commercially available, software that incorporates multi-block automatic features. The output, a multi-block grid, contains a high-quality grid and optimized distribution across the entire domain. This organized structured mesh thus translates into a higher solution accuracy and faster convergence for the CFD solvers. Some of the specifications and measurements that define the high-quality of a multi-block structured grid is the orthogonal cell organization, smooth transitions, low-warpage, and continuous curvature clustering.

A near-orthogonal grid means that the grid lines that conform the cell intersections are near a 90-degree angle. Orthogonality in the grid has a strong influence in the equations of fluid dynamics as elliptic or hyperbolic methods of partial differential equation schemes can be formulated. Local orthogonal grids have great significant in the boundaries. In other words, cells will form perpendicular lines from the surface which would enable smooth mapping and clustering that would allow for a high-accuracy feature extraction in a CFD solver. Smoothness in a grid means that the multi-block decomposition will allow for smooth transitions from block-to-block as well as a desirable (I,j,k) specifications of number of cells in all three directions. One of the ways smoothness can directly be measured is through aspect ratio. This is the ratio of the longest to the shortest side in a cell. Warpage is the measure of the angle of a face which is defined as the angle of the normal vectors of the two planes generated by cutting through the diagonal corners of the face. Finally, a continuous curvature clustering is the ability to multiply the grid lines and therefore multiply the cells as it gets closer to the surface boundaries. At the same time, without affecting the overall quality of the grid regardless of the complexity of the surface.

2.1 Advantages of Hexahedrals (structured mesh)

The Finite Volume Method has a straight forward approach when dealing with hexahedral cells. This is the case as opposed to non-hexahedral cells such as tetrahedral or other known polyhedral structures. Hexahedral cells are six-sided cells and can easily be mapped into a three dimensional X-Y-Z coordinate system with three face pairs, see the figure below.

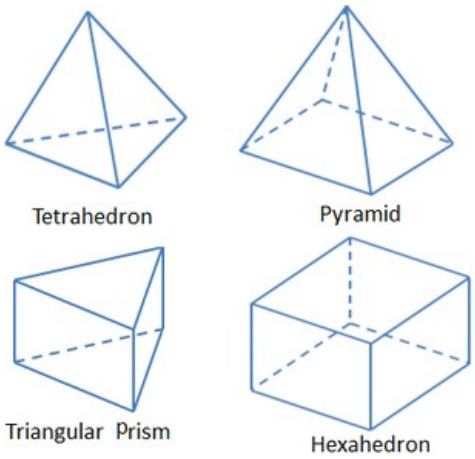


Figure 4 – Finite Volume Cell types

In a structured CFD grid, the mapping from the physical domain to the computational domain has more fluent transformations. This means that the way the information is addressed and contained within from the geometry and mesh files can explicitly be translated to the computer as this will store the data into matrices in the x,y, and z directions.

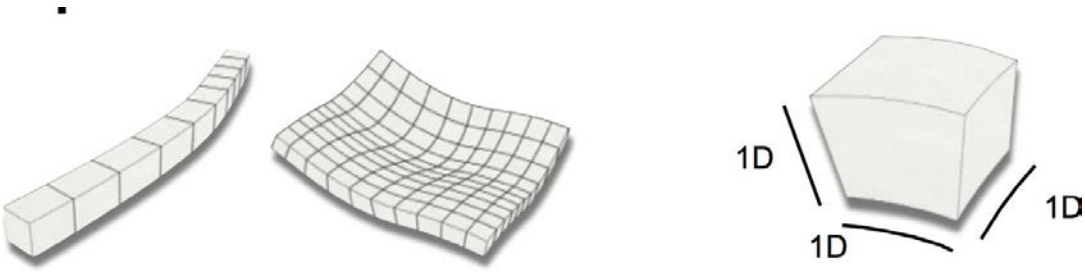


Figure 5 – Structured Mesh mapping

Having pairs of faces will induce a more flow alignment where, whichever numerical scheme is applied, the profile of solution will more naturally expand throughout the domain.

The flow flux which can be described by the flow streamlines will more easily connect between face pairs, that is between opposite faces –x to +x, -y to +y and –z to +z. Some of the challenges that may be encountered when not using hexahedral cells is the higher non-orthogonality which can introduce misalignment between the computed gradients.

Chapter 3: Proof of Concept – Orion capsule grid generation

A first study was done on exclusively learning how to use GridPro. The Orion spacecraft was chosen. A 1:1 geometry was drafted in SolidWorks and imported into the GridPro Workspace. The figure below shows the Orion spacecraft as it appears on the WS GUI. The main reason why the Orion capsule was chosen is due to its convex surface all-around. This makes it simple to understand the concepts that we are going to introduce next.

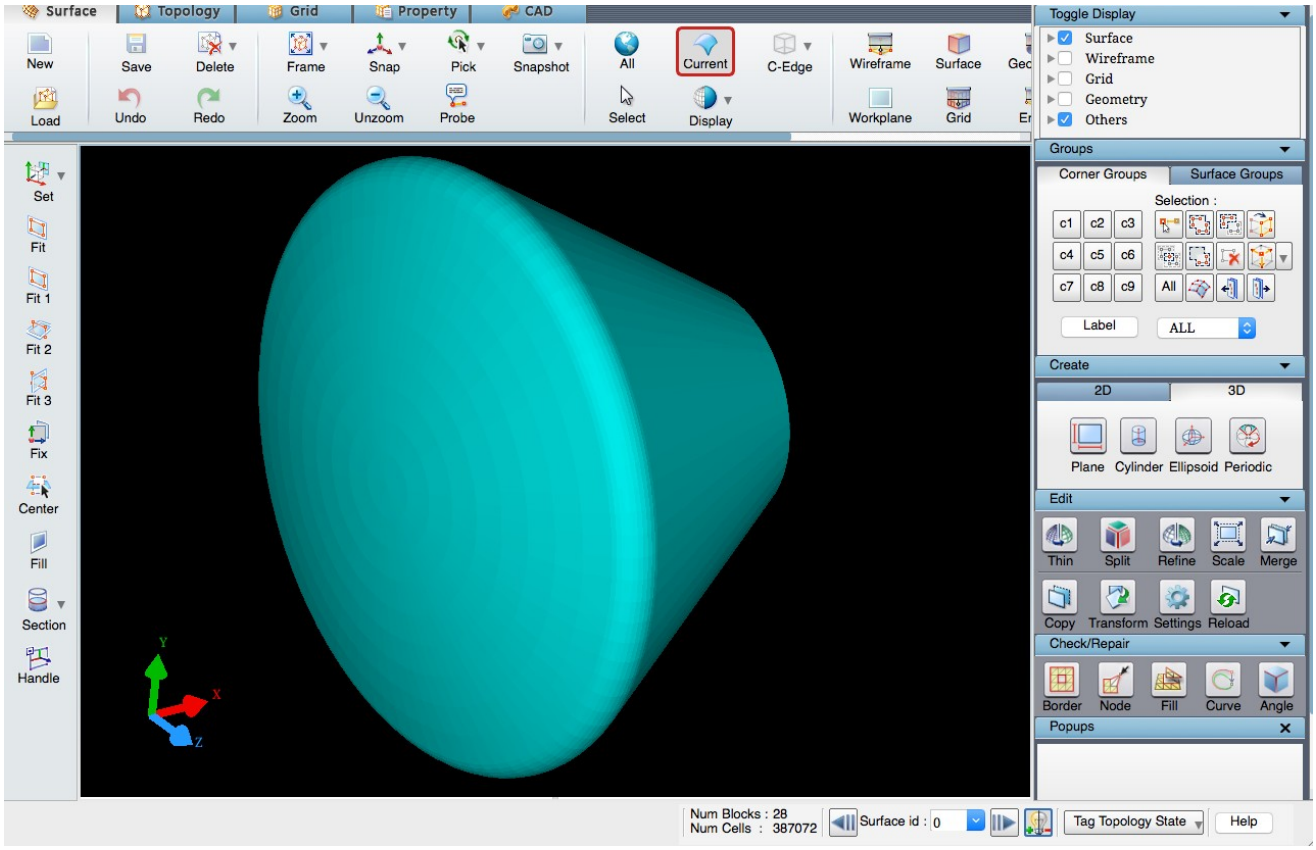


Figure 6 – GridPro WS GUI with Orion surface

3.1 O-type grids – Topology construction

O-type grids are highly advantageous type of grid for computational fluid dynamics. An O-type grid connects blocks in a full 360-degree around the shape forming an O-shape block continuum. As the grid wraps around the closed shape and it precludes from any skewness

rooted from the surface and boundary layer. This is part of the bottom-up approach to create a structured grid. As shown in the figure 7a, on the left, corner points were first created around the Orion surface. These corner points were connected in quadrilaterals in order map the block faces that lie on the Orion surface. Therefore, the blocks form almost-perpendicular to the surface. The Orion surface in the figure is made transparent to help visualize the topology. On the right figure 7b, the GridPro "wrap" tool is used to complete the formation of the blocks on the Orion surface. This is the starting point of the near region of the capsule. It clearly does not extrude to the far field. On the figure 7c below, it's a preview of what cell faces that touch the surface based on this O-grid topology.

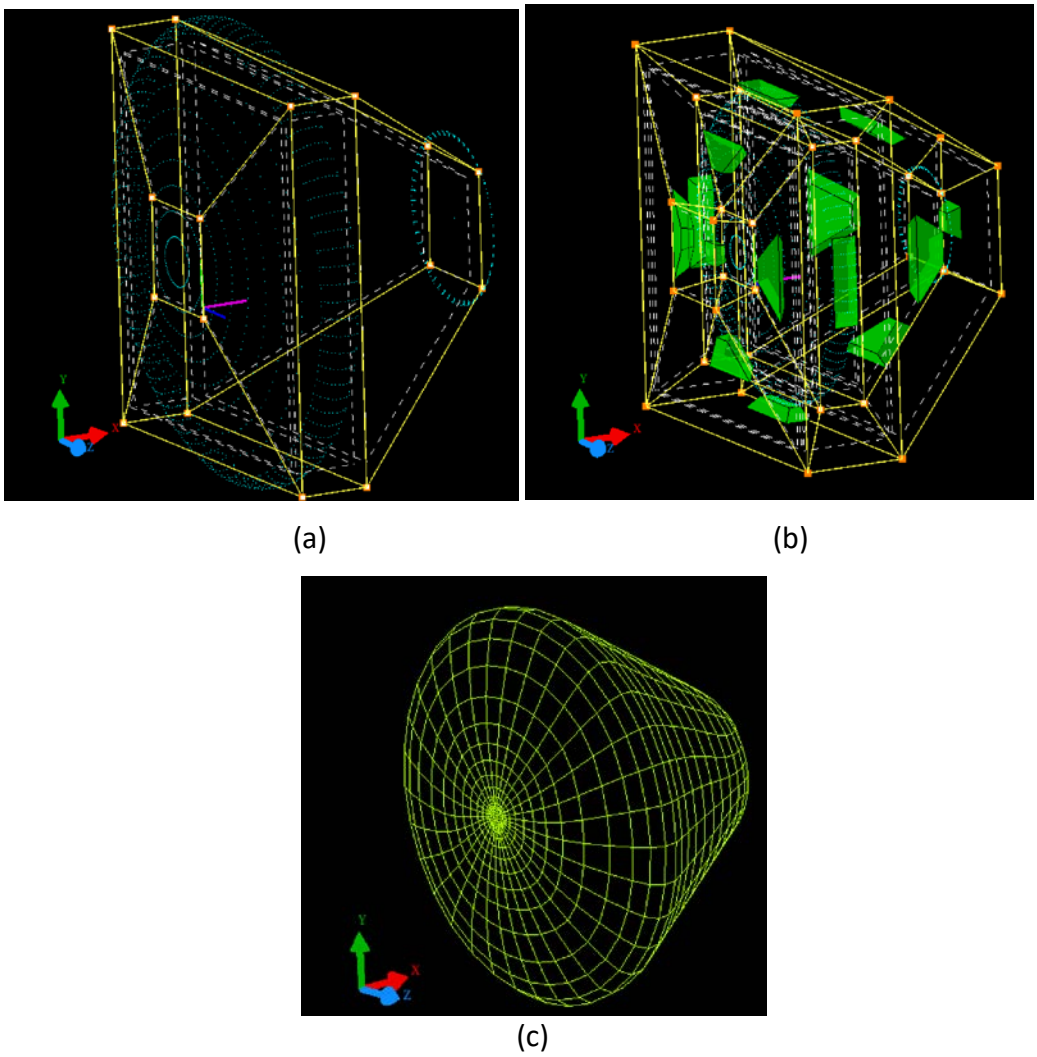
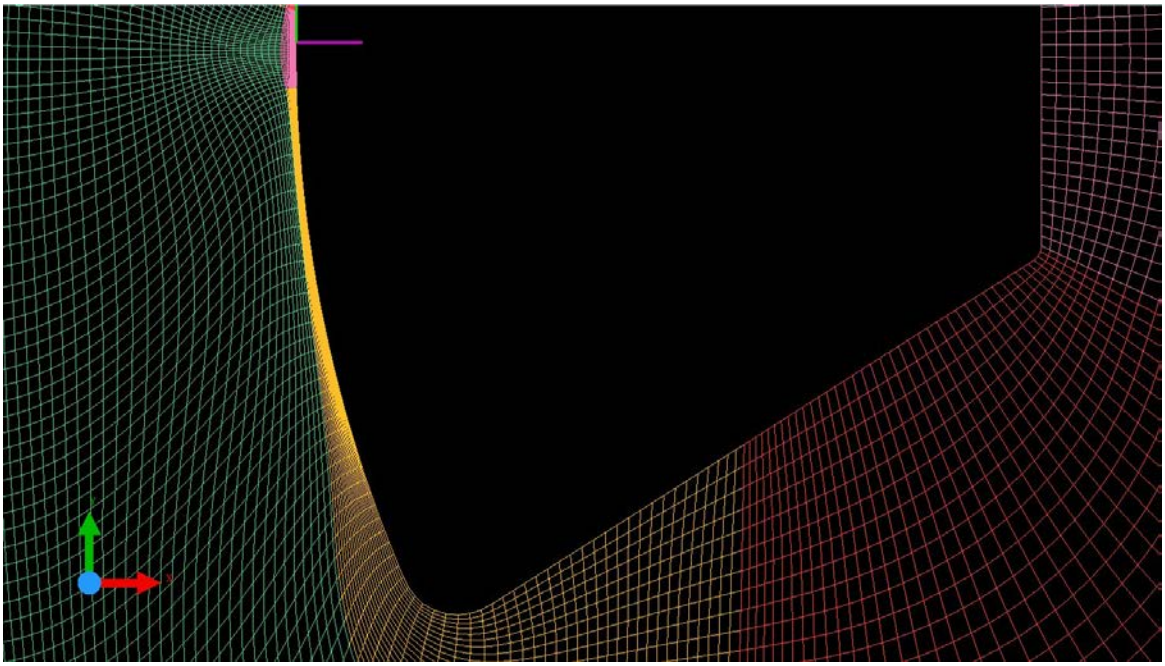
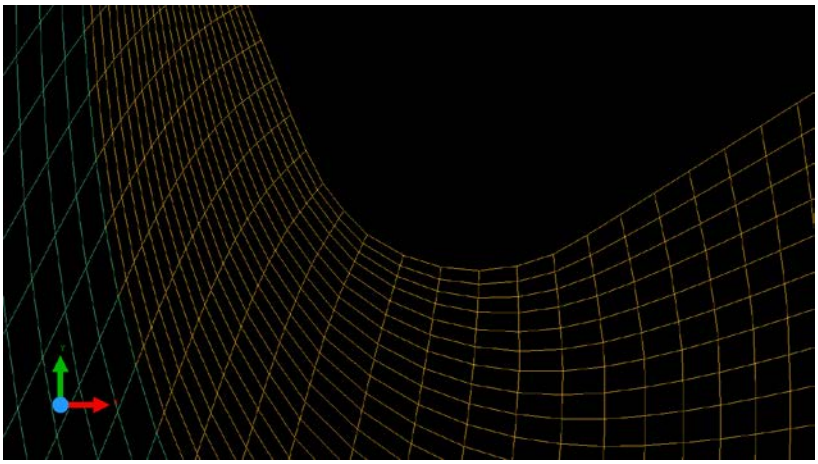


Figure 7 – GridPro Wrap tool for O-grids (a) topology at the Orion surface (b) Wrapped topology including block preview (c) Computed grid result

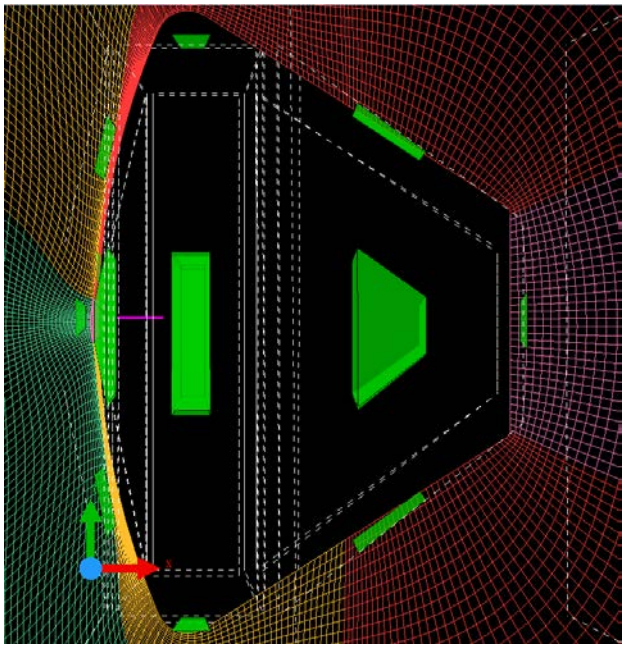
As part of this proof of concept, it can also be demonstrated that an O-type grid has smooth characteristics around a convex surface such as that of the Orion. One of the reasons here, is that orthogonal lines are stretched from the surface. To better visualize this, we increased the density of cells in the blocks, calculated the mesh again, and took a slice of the updated mesh that displays the orthogonal lines as shown in the figure below. The different colors correspond to the different blocks of the structured grid. Figure 8a shows the bottom section (symmetric) of the Orion. The transition in colors represent the defined blocks that enclose the Orion surface. Figure 8b is a much closer look to the high curvature surface. Finally, Figure 8c shows the representative blocks and topology that was constructed on the geometry to accomplish the structured grid.



(a)



(b)



(c)

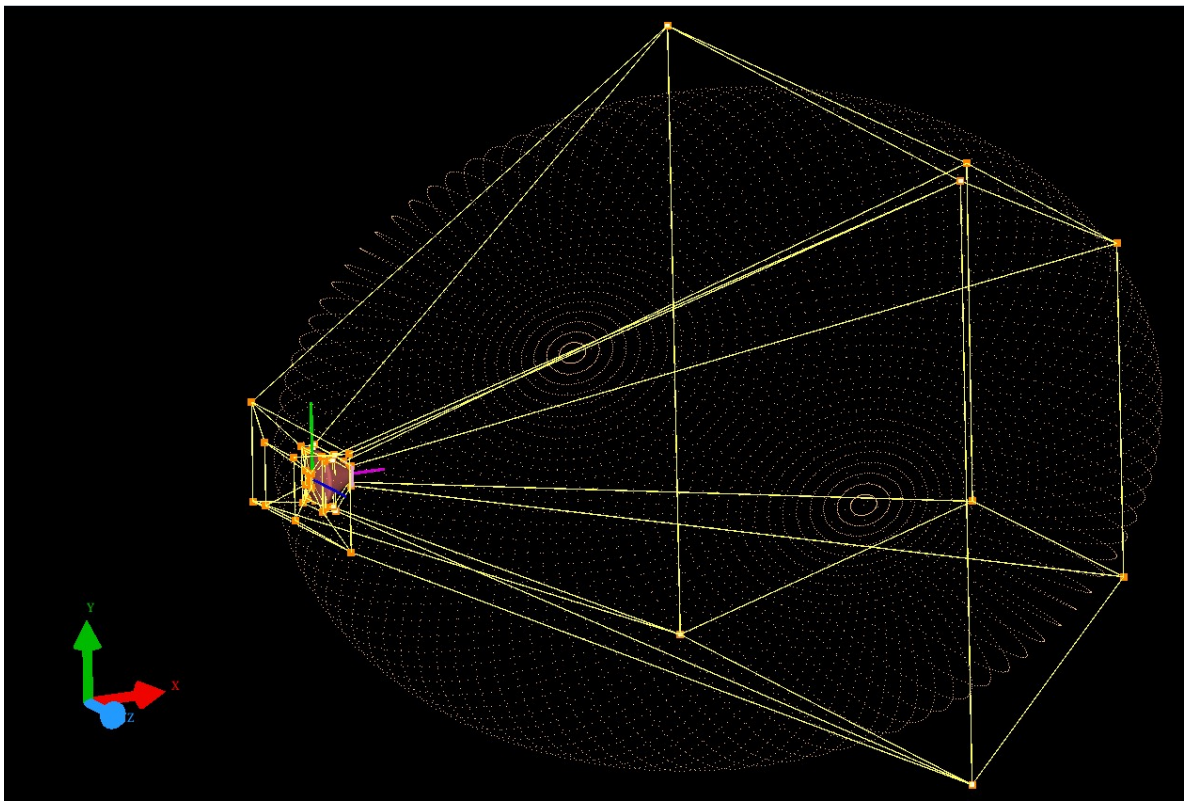
Figure 8 – Finer computed grid showing X-Y plane sheet across Orion (a) bottom Orion surface (b) close-up of bottom Orion surface (c) whole Orion showing blocks

Now that the near-field grid has been defined, the topology can be further constructed to cover the entire domain and far-field. Since we could capture the surface of the Orion spacecraft in a near-perfect O-grid, then another O-grid can be wrapped around it to extend as far as the far-field needs to be. The way this is done, in GridPro, is by directly manipulating the

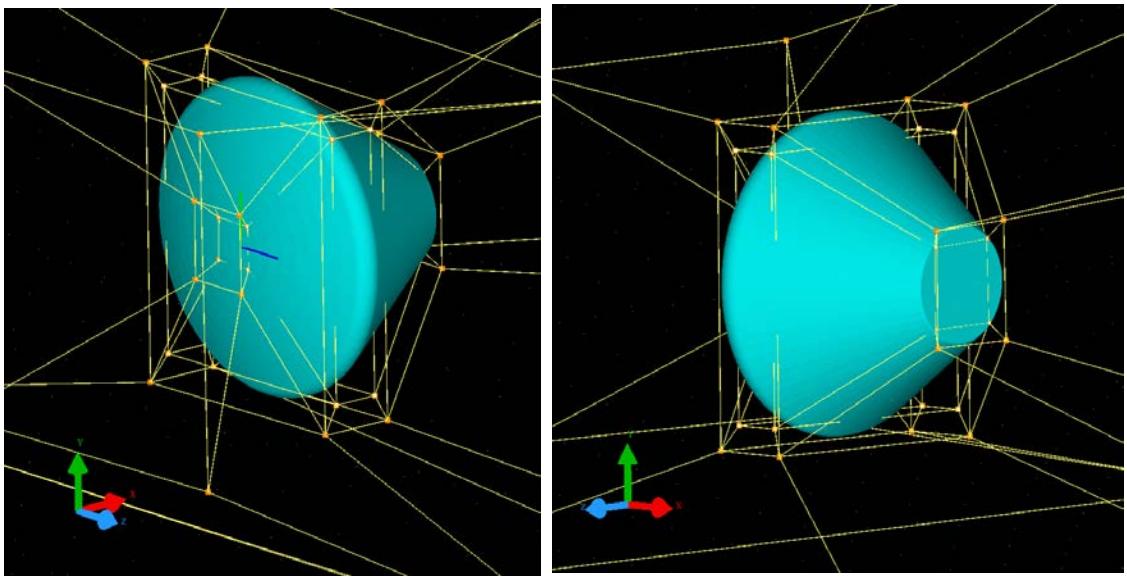
wireframe and enlarge the links behind the capsule. Moreover, the links ahead of the capsule do not need extend much forward, since the inlet boundary conditions are applied here.

Further here, the links that constitute the topology, have a defined density of cells from one link corner to the other same link corner. It can simply be deduced, that if any links are stretched further back behind the capsule, then the cell spacing is much larger behind the capsule, towards the outlet, than on the front. The topology now in principle is consisted of links with density of 8 cells (GridPro default). Furthermore, since a wrapped collection blocks were created, then a successful group of blocks can also be wrapped in either direction as many times as needed. This is because the validity of the topology was defined and there are no singularities of consideration. Next, we can define the entire three-dimensional domain. This is essentially constrained by what would be the inlet, wall, and exit boundary conditions. The shape enclosed by the domain limits is an ellipsoid which is a realistic and typical compressible flow domain shape as opposed to cuboid shaped domains.

As shown in the following Figure 9, the topology was stretched to represent the entire domain. Here, one can clearly see the two-level O-type grid from the Orion surface outwards where the outer spherical grid stretches out to the domain boundaries as previously defined.



(a)



(b) (c)

Figure 9 – Complete topology-wireframe around entire domain (a) near-isometric front view (b) front capsule close-up (c) rear capsule close-up (c)

3.2 O-type grids – Grid Computation

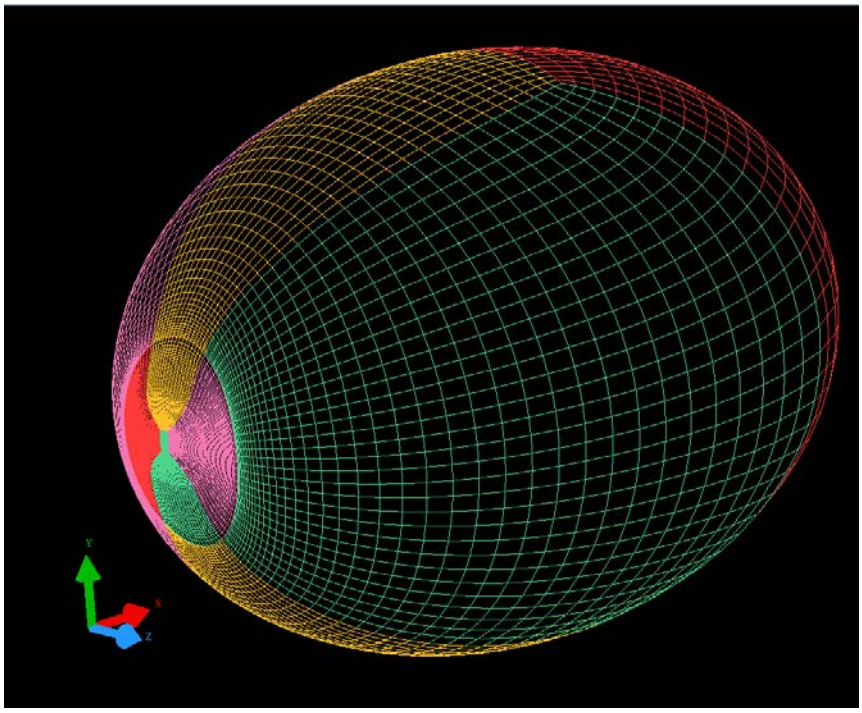
Once the topology is completed, the next step is to run the GridPro software (Ggrid) to compute the grid. Ggrid first validates the topology by ensuring that all blocks are defined and that the 3-dimensional cell domain can be passed into a computational domain. Once this step is completed, a set of Ggrid options are available for the user to choose. Of those, the ones of main concern are the solver type and scheduler. Firstly, for the solver type, there are two options, Navier-Stokes equations which are the full equations of transport or the Euler (inviscid) equations which are one-side of the Navier-Stokes equations. As part of this POC, and since we are not exactly interested in solving the resulting grid from this model, the Euler option is chosen.

The second solver option is the scheduler. In summary, the scheduler selections were to save the grid file every 100 sweeps. Next, the output file format was chosen to be ASCII which, as opposed to the other choice (binary), is readily available should changes be needed along the way. There were no other specific scheduler options chosen, however, GridPro contains many custom scheduling options especially used for grids with complex clustering and convergence criteria.

The software was then launched and for this idealized model, it took less than 1000 sweeps to quickly converge and obtain a multi-block structured grid. At that point, the computation can be aborted and the last successful saved file, specified by the scheduler, is the final product. The gridding process isn't explicitly or visually seen in the GridPro GUI. This is rather a computation in the background. A console with the status of the computation is printed on the computer so that a user can tail the process. What's seen in the console window contains real-time information about the quality and convergence of the gridding computation. Firstly, it prints the sweep count. The sweep is the number of times the software has run around the entire topology to optimize the grid. The sweeping process is feedback controlled, therefore every subsequent sweep improves the quality should all checks pass. Next, the residual is the rate of change of the grid from the last computed sweep against the current

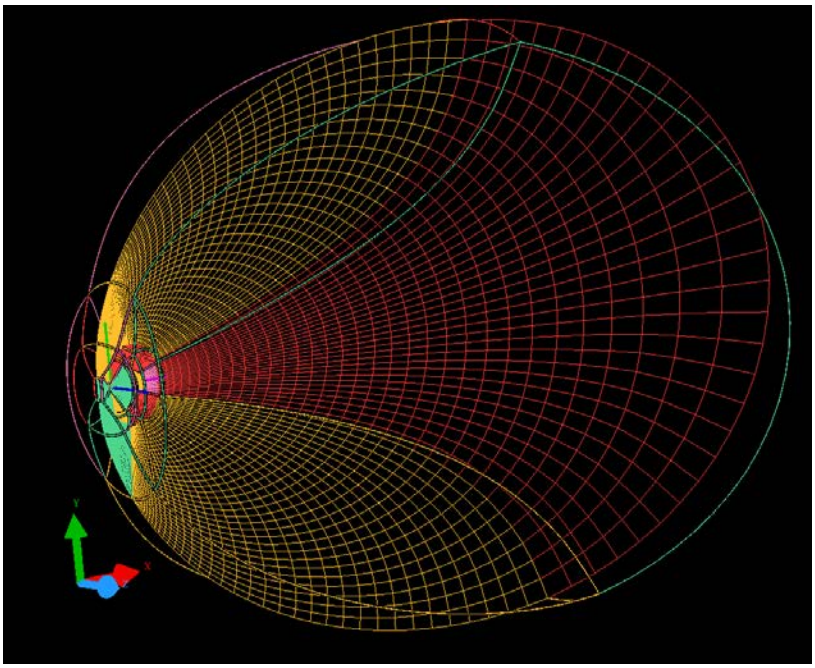
sweep. The smaller the value of rate of change becomes, the smaller the residual becomes, and this number approaching zero is indicative of a convergence behavior which is when a user can choose to kill the gridding process. The third and last parameter see on the console is the fold count. The number of folds is the total number of cell discontinuities in the domain. This is denoted as the number of sheet folds that usually is concentrated in a local section of the domain rather than entirely. This itself good as it's clear to the user, that the areas in the domain that need further refinement to improve the quality characteristics of the grid. A sample of the Ggrid log for the first 100 sweeps of a compute can be found in **Appendix A.**

When Ggrid is killed and a grid file is successfully created and saved, it can now be loaded into the GUI. For this Orion POC, the following images in **Figure 10** describe the resulting grid.

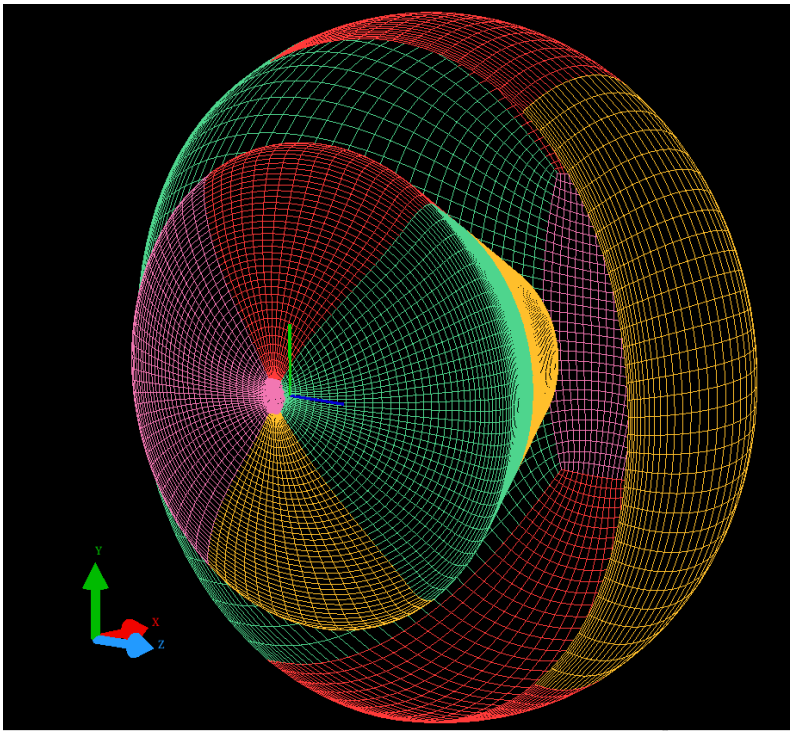


(a)

Figure 10 – Computed grid results (a) near-isometric view of entire domain

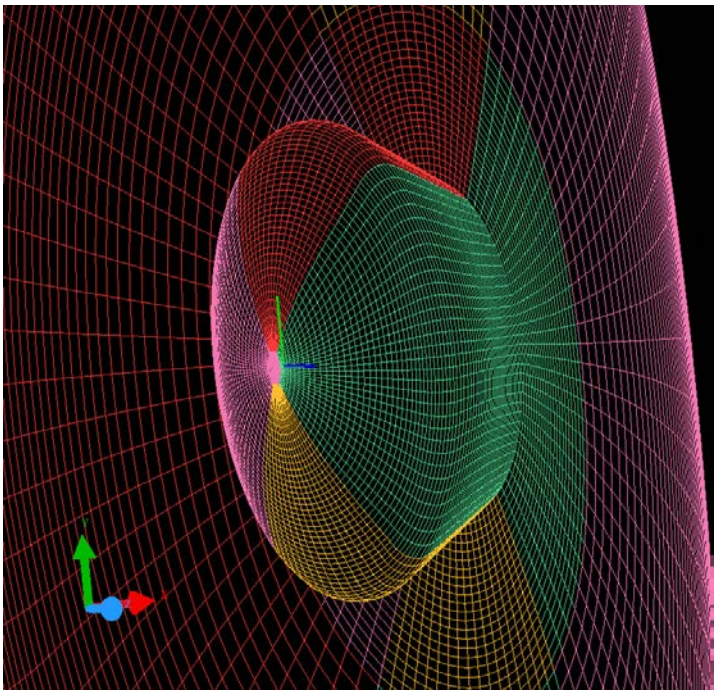


(b)

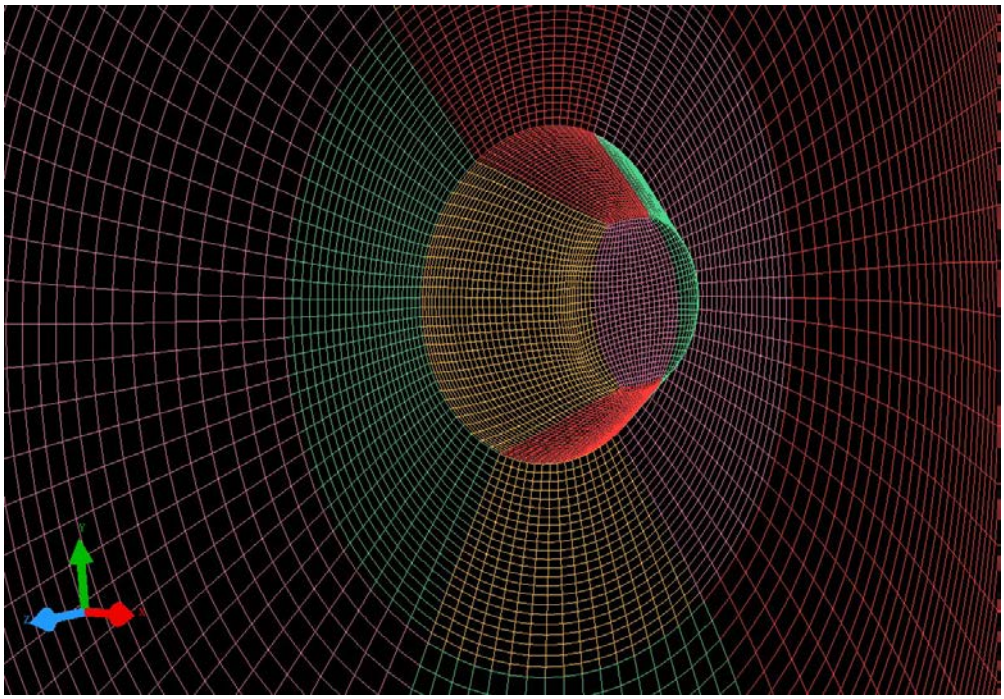


(c)

Figure 10 cont'd– Computed grid results (b) slice cut on a near X-Y plane (c) sliced view around Orion showing O-grid structure

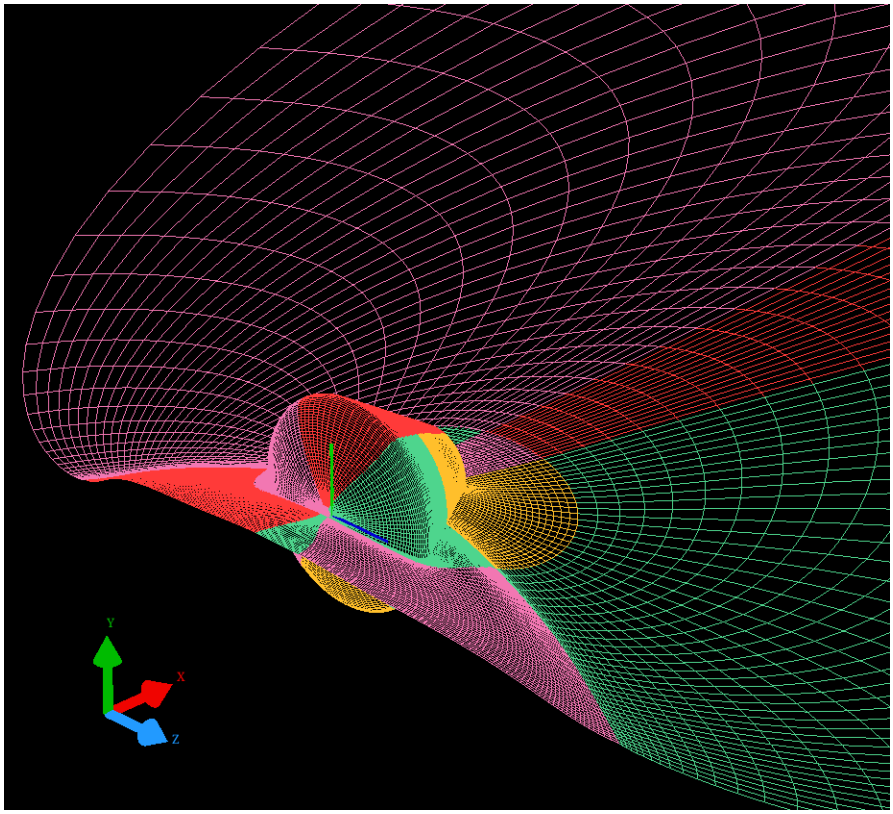


(d)



(e)

Figure 10 cont'd– Computed grid results (d) near Y-Z plane sliced view across Orion front (e) near Y-Z plane sliced view across Orion back



(f)

Figure 10 cont'd – Computed grid results (f) near X-Z plane sliced view across Orion

Chapter 4: Proof of Concept – Pseudo surfaces

In this chapter, a special tool in multi-block grids is learned. Internal or pseudo surfaces are the focus of this chapter. Pseudo surfaces are two-sided surfaces that are used to enhance the grid lines across a complex geometry, particularly, low smoothness and/or discontinuous surfaces, such as sharp edges. These pseudo surfaces are supported by GridPro. A capsule-parachute system would have several sharp areas so knowing how to use this tool is highly advantageous. To create these pseudo surfaces, they must exactly pass through the slope discontinuity of the geometry. Also, the pseudo surface should also cut the geometry orthogonally. Conveniently in GridPro, the topology members such as corners in corner groups and surfaces in surface groups can be selected to guide the pseudo surface creation.

4.1 Pseudo surface – set up

The procedure to for a pseudo surface is to first create a set of corners on the surface slope discontinuity using the triangulation nodes of the real surfaces. In GridPro, this task is called Feature Edges. Alternatively, if the surfaces are two or more and the discontinuous boundary happens to be between two neighboring surfaces, then GridPro can achieve the same purpose using the “Intersection of Surfaces” tool. The figure below shows the sample model that was used to learn to use pseudo surfaces. It’s essentially a T joint where the junction of the two tubes is bounded by a concave-like discontinuity.

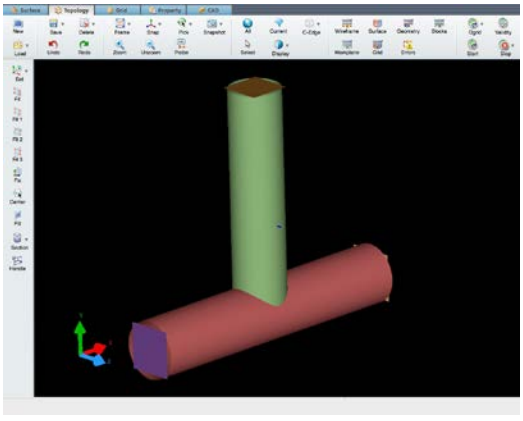


Figure 11 – T joint model for pseudo surfaces

A wrapped topology is first built as it normal. The exception here is that there is no smoothness between the two pipes hence the need for an internal surface creation. Again, the intersection of both surfaces is first defined by a special GridPro direct tool. It is also worth mentioning that there are two types of internal surfaces. For two neighboring surfaces of which the average angle between the two is less than 180 degrees, then the surface needed is a cone shaped internal surface. For two surfaces of which the average angle between the two is more than 180 degrees, then the surface needed is a cup shared internal surface. The case of the t-join requires the use of a cone shaped internal surface. The theory just mentioned is illustrated in Figure 12, as well as, the GridPro internal surface input window in Figure 13.

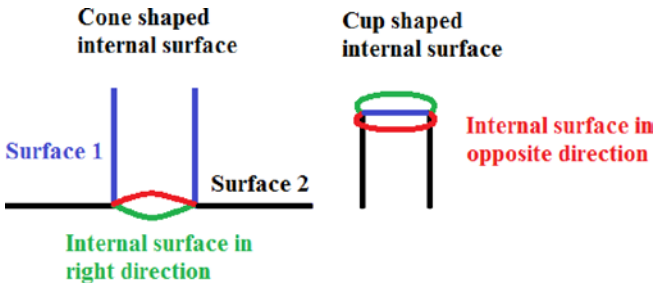


Figure 13 – Possibilities of internal surface creation

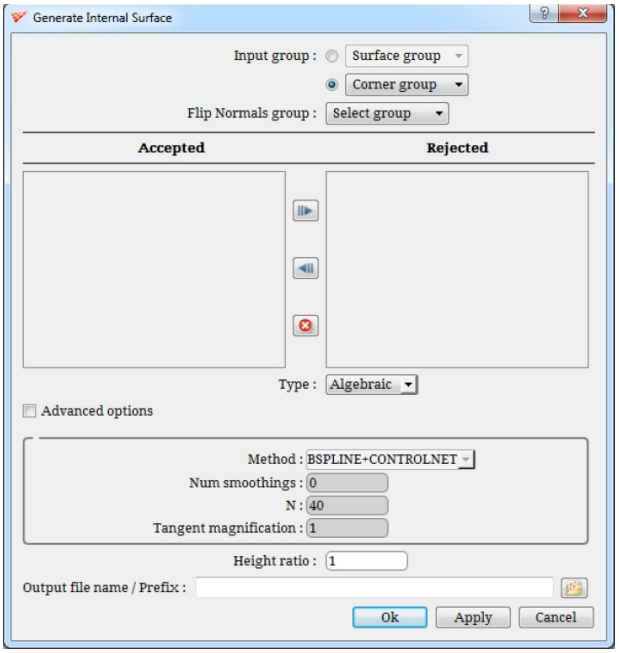


Figure 14– GridPro GUI internal surface input window

The wrapping tool does not need to be explained in this chapter. The pseudo surfaces must also be assembled to the topology by the corner assignments, the idea is the same as any non-pseudo surface topology assignment. However, an additional rule must apply here and that is that the corner/edges that are only assigned to an internal surface should share two blocks on either side. An additional sheet is inserted near where the pseudo surface lies. Then, the assignment is simply carried to the new inserted corners. The following two images next show the complete topology. As seen below in figure 15, the green blocks are adequately placed in the domain. The pseudo surface is assembled to the topology therefore seeing a valid topology is indicative that complete blocks of cells without folds are expected.

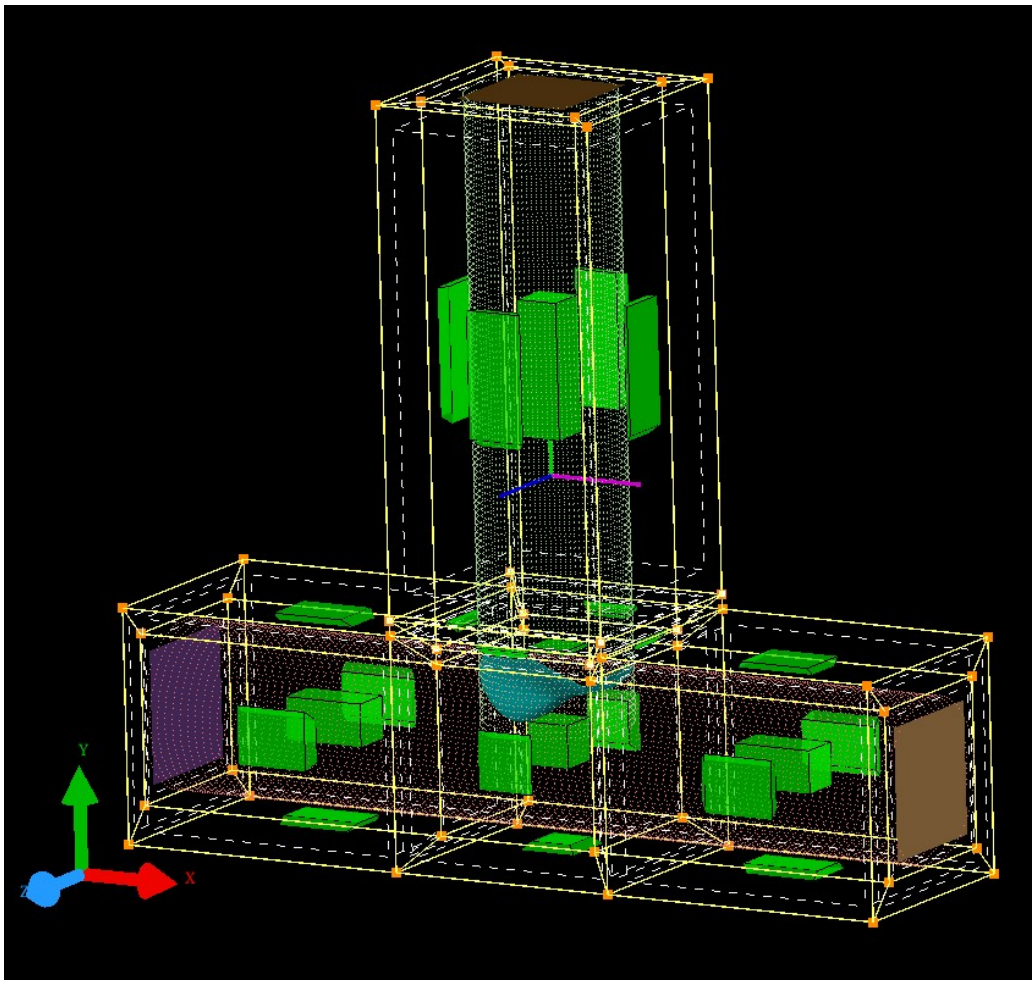


Figure 15 – T joint completed topology (with pseudo surface)

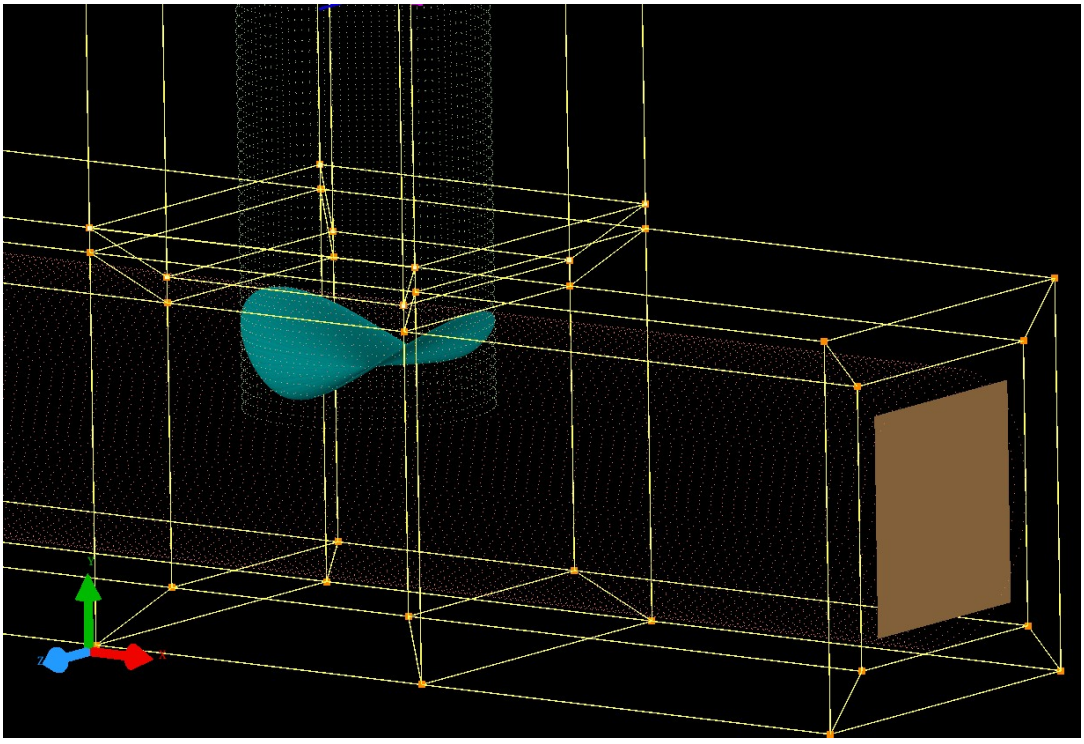
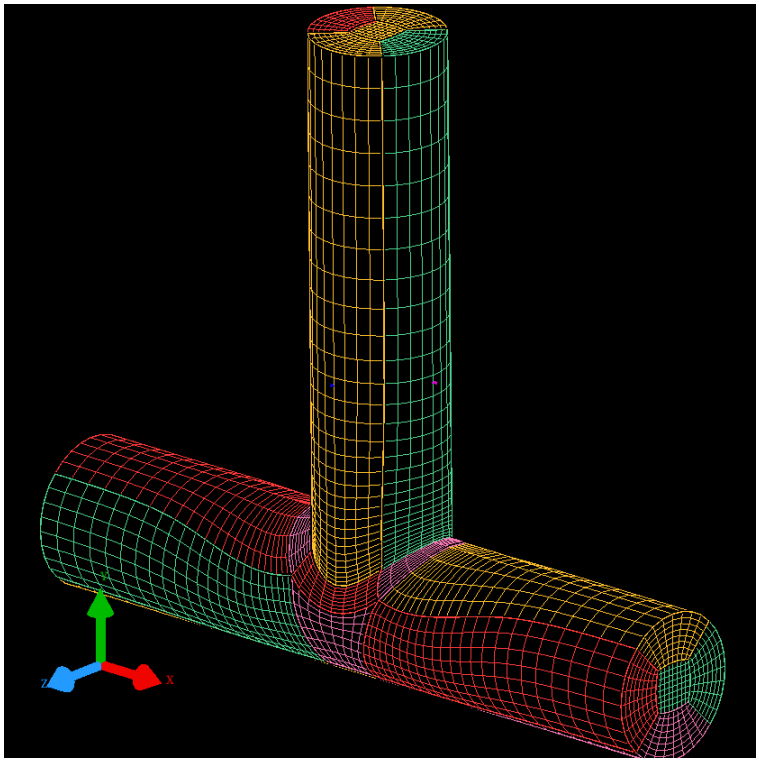


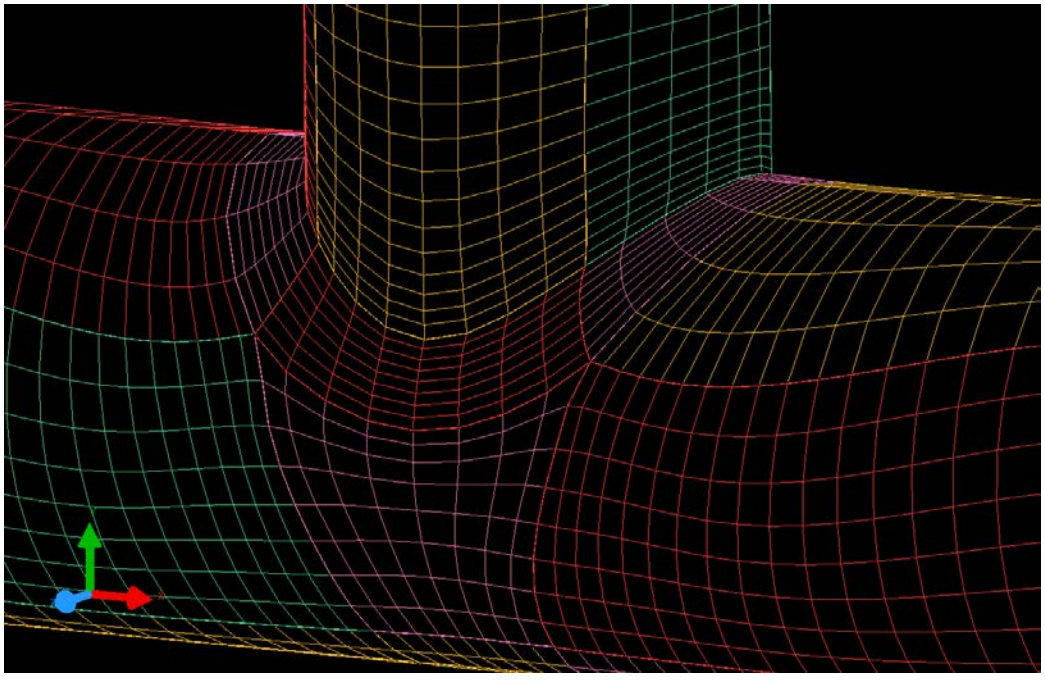
Figure 16 – T joint close-up to pseudo surface and corner assignment

4.2 Pseudo surface – Grid computation

The constructed topology is a valid topology. Ggrid can be run a similar manner as the previous chapter. Ggrid is set to use the Euler equations and use the default scheduler of saving a grid file every 100 sweeps. The results were favorable as there were no folds found and a low residual was achieved. The following figures display the computed structured multi-block grid of this T-joint example.



(a)



(b)

Figure 17 – T joint computed grid (a) near-isometric view (b) close-up to discontinuous joint section

Chapter 5: Capsule-Parachute Grid Generation

Having established successful proof of concept scenarios, a consistent knowledge base has been built to construct the intended capsule-parachute assembly in a compressible fluid flow domain. The selected assembly consists of a 18-panel disk parachute model that is 20 m in diameter. It's not an exact copy to any of the parachutes that have been used for Mars Exploration. However, it has the similar overall shape constraints. Some of the constraints are minimal thickness (1-4 microns) and the hemisphere-like shape. Lastly, although the Orion was replaced with the capsule geometry from the Viking program, the gridding technique is identical which is to use an O-type grid to wrap around it.

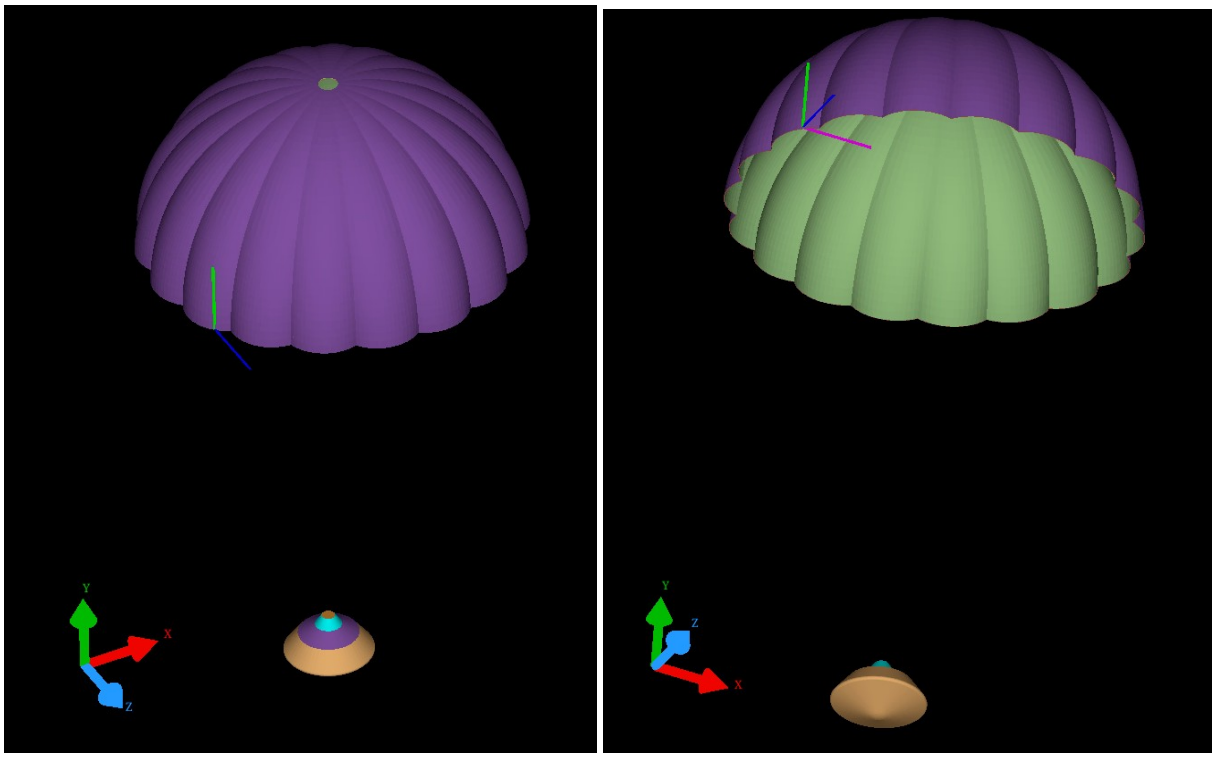


Figure 18 – Final geometry – (a) near-top view (b) near-bottom view

By combining the knowledge base from pseudo surfaces and GridPro automation tools, the entire domain can now be constructed. Firstly, the new capsule is again enclosed in a O-

type grid. It is also necessary and advantageous to divide the capsule surface into smaller surfaces bounded by the location of maximum slope. This way, the corner topology assignment to the shape is more stable in the gridding process. The following image below show the techniques learned in our POC and being applied to the capsule parachute system.

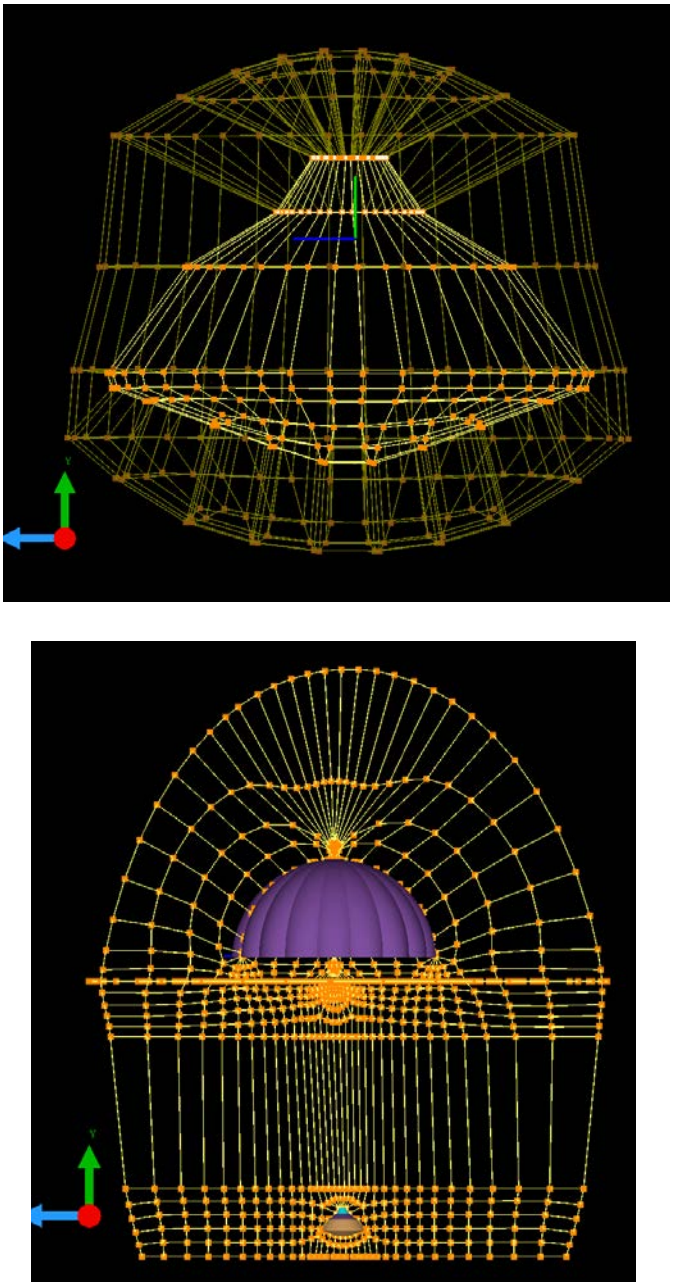


Figure 18 – Final topology (a) capsule displaying O-grid (b) topology face pattern on parachute

Pseudo surfaces were also constructed in 4 different locations of the parachute. The first 2 are on the base ring and the other 2 on the top hole. The images below are the final topology before Ggrid execution.

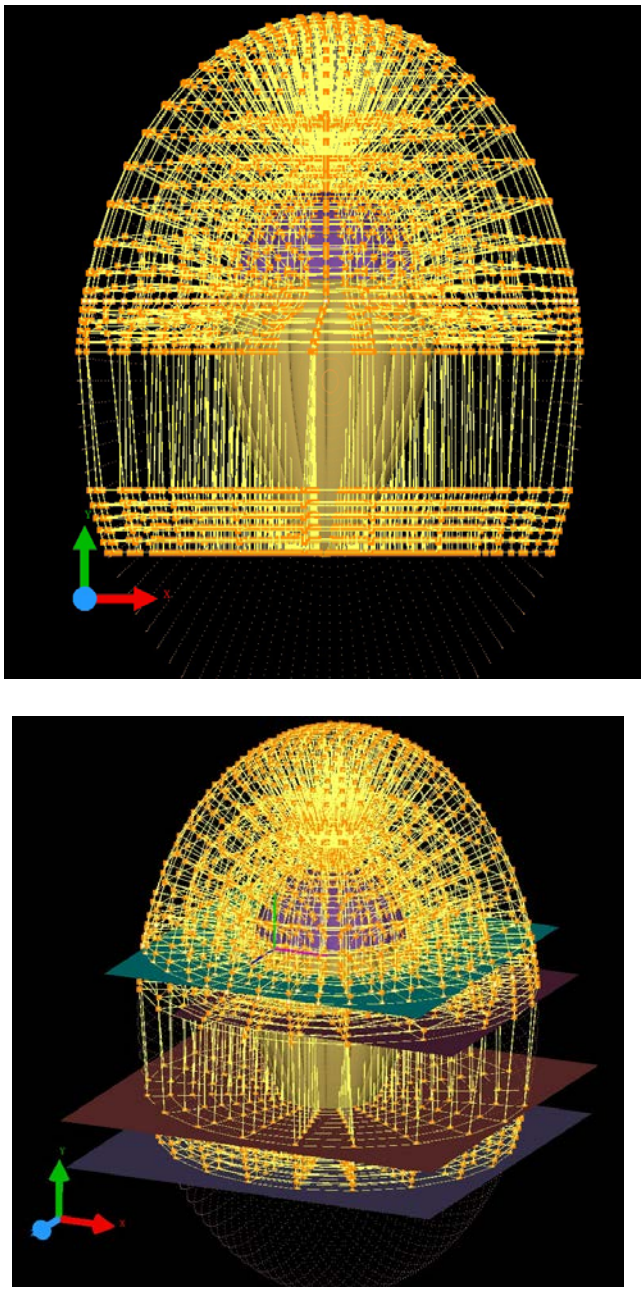


Figure 19 – Final topology (a) front view of the overall (b) symmetric view of overall domain

4.2 final capsule-parachute assembly – Grid computation

Ggrid is executed using Euler's equations (no clustering) and writing a grid file every 100 sweeps. The results were favorable. 0 folds were encountered; however, the residual was not optimal. This may be due to issues with aspect ratios constantly moving in a specific localized block or set of blocks. The final grid is shown in the figure below. Notice the different colors that correspond to different blocks.

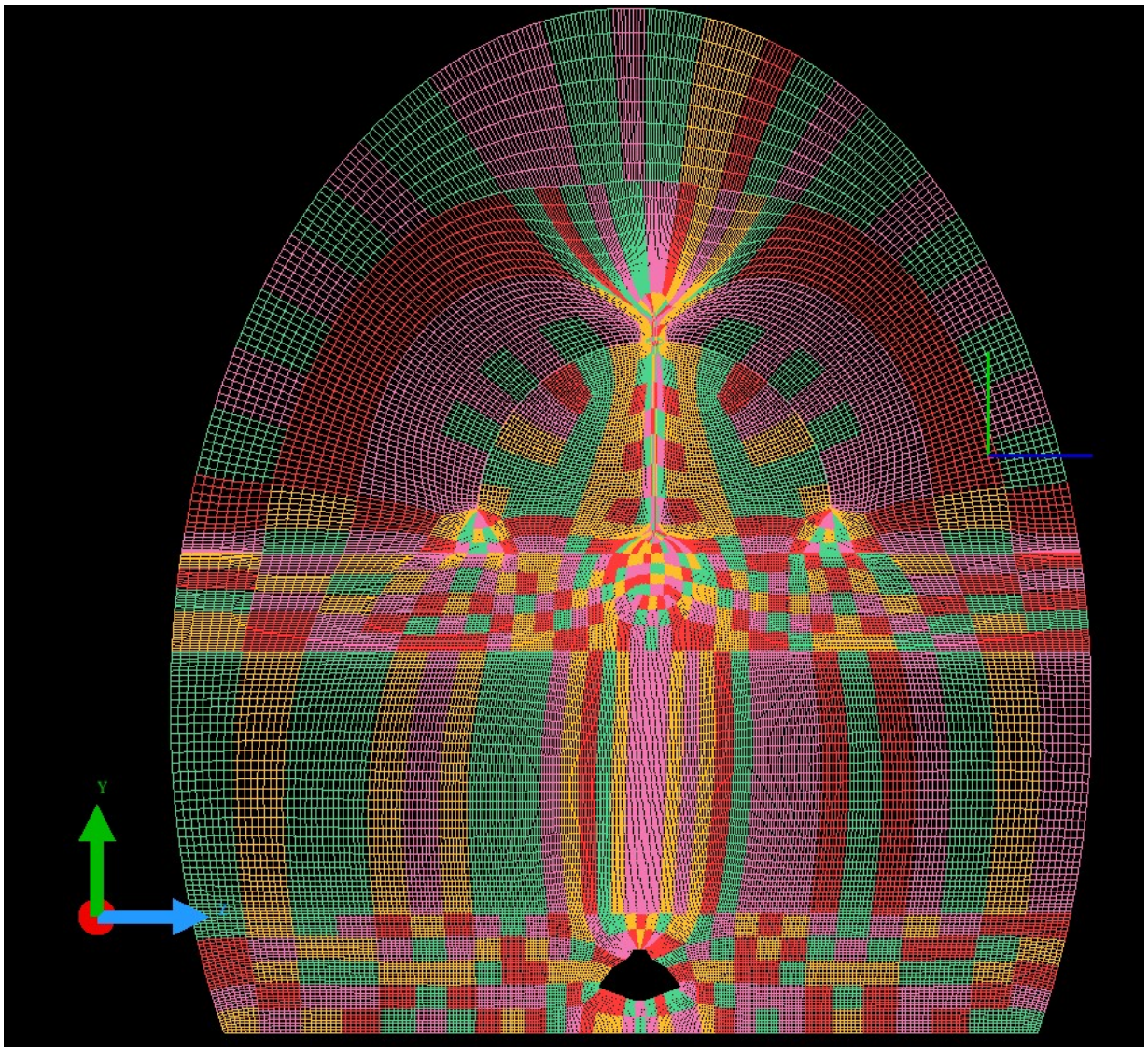


Figure 20 – Sliced front view of capsule-parachute system grid

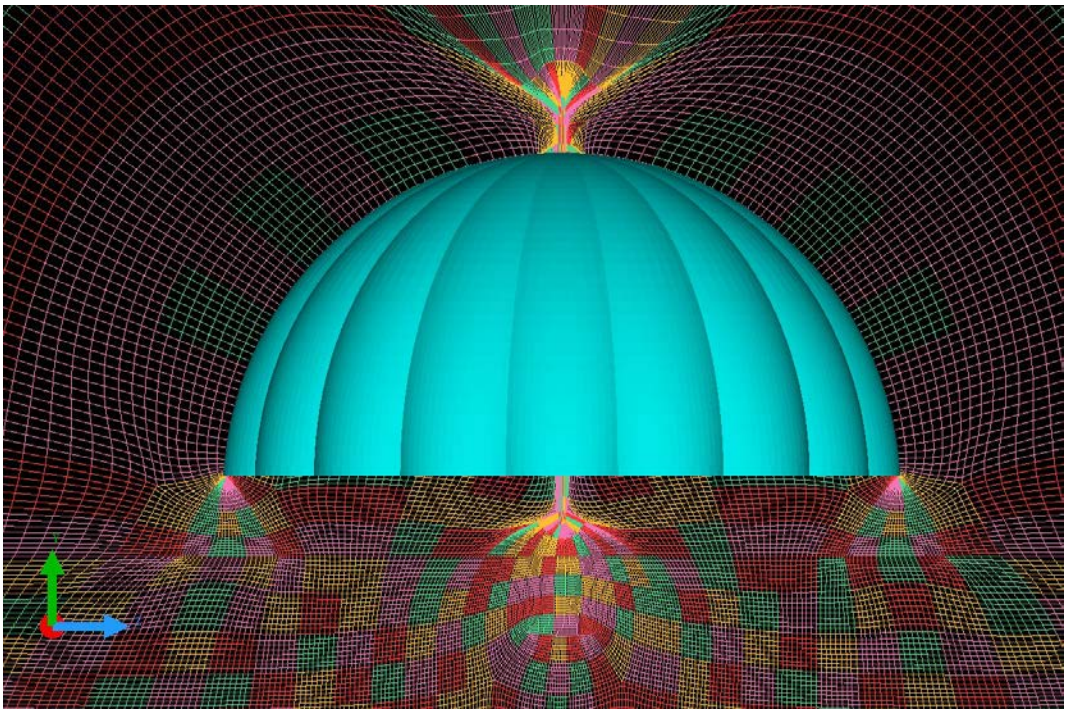


Figure 21 – Close-up of sliced front view of parachute

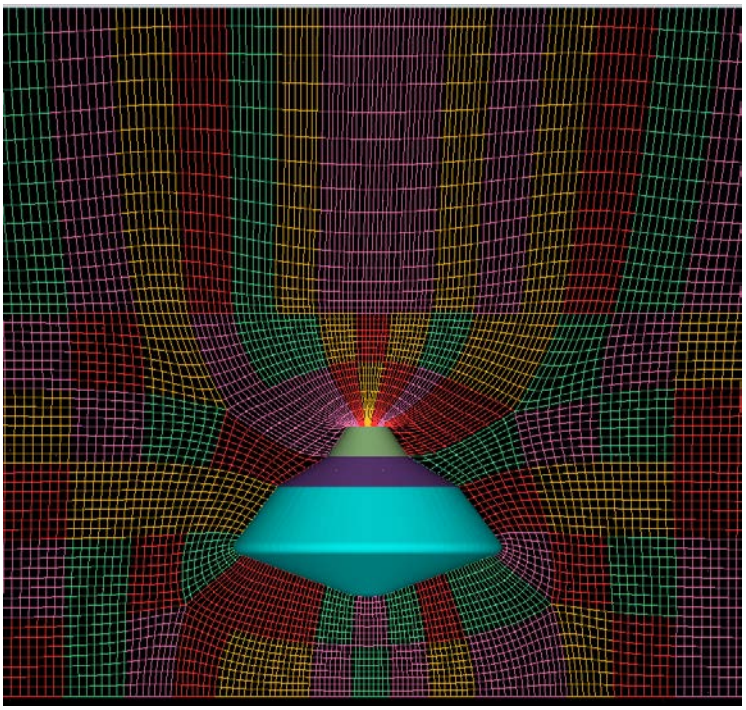


Figure 22 – Close-up of sliced front view of capsule

Chapter 6: Conclusion and Future Work

6.1 Conclusion

The investigations provided us with a framework for creating high-fidelity models for compressible flows. Within the subject of mesh-required Navier-Stokes problems, solutions that required great accuracy rely heavily on how the mesh was built. There are essentially two types of grids: unstructured and structured. Structured grids have proven to reach solutions faster and these don't demand a lot of computational expense as opposed to unstructured meshes. However, there is a great notion among engineers that structured grids are difficult to construct. This is true. Unstructured grids don't require as many special rules as those for structure grids. Thus, a significant amount of automated meshing goes on with unstructured grids. The amount of user intervention and manipulation is thus much less with unstructured grids. For all these reasons, the use of structured grids is not so demanded anymore.

This project's goal was to demonstrate good practices of engineering reasoning and methodical thinking in developing a plan to create a structured grid for a supersonic decelerator. The proof of concept tasks was vital to lead to the development of high-fidelity multi-block grid for complex assembly of a deployed parachute over a Martian capsule which is an on-going area of research. This papers proves to provide technical recommendations to achieving a multi-block grid in this domain-specific problem.

6.2 Future Work

Following the grid generation process, the mesh is to be imported to a commercial CFD solver. Here, the boundary conditions such as no slip condition on the wall, the free stream inlet far away from the body, and the outflow properties will be established. It can be predicted that a turbulent model will be chosen (Reynolds Number calculations first being verified), and finally a decision must be made to determine if the flow is viscous (Navier-Stokes) or inviscid (Euler).

At the end of the first rounds of simulations, the results will be tabulated and compared against experimental data (if available), and CFD published articles. Depending on the development of this project, the analysis can be linked to the design of these parachutes, and recommendations can be proposed; however, the immediate goal of this study is to fundamentally understand the flow field physics around the spacecraft as it decelerates from Mach 2.5 to Subsonic.

References

[BDNCG07] Barnhardt, M., Drayna, T., Nompelis, I., Candler, G., Garrard, W., 2007. Detached eddy siulations of the MSL parachute at supersonic conditions

[D69] Daveport, E., 1969. Static longitudinal aerodynamic characterisitcs of some supersonic decelerator models at Mach numbers 2.30 and 4.63

[H66] Heinrich, H., 1966. Aerodynamics of the supersonic guide surface parachute.

[KKCP11] Karagiozis, K., Kamakoti, R., Cirak, F., Pantano, C., 2011. A computational study of supersonic disk-gap-band parachutes using Large-Eddy Simulation coupled to a structural membrane

[LD05] Lingard, J., Darley, M., 2005. Simulation of parachute fluid structure interaction in supersonic low.

[LDU07] Lingard, J., Darley, M., Underwood, J., 2005. Simulatoin of Mars supersonic parachute performance and dynamics.

[NP73] Nerem, R., Pake, F., 1973. A model and calculation procedure for predicting parachute inflation.

[PSH96] Peterson, C., Strickland, J., Higuchi, H., 1996. The Fluid Dynamics of Parachute Inflation

[RN66] Ross. R., Nebiker, F., 1966. Survey of aerodynamic deceleration systems

[S72] Stevens, G., 1972. A theory of vibrations in parachutes.

[SBC97] Schofield, J., Barnes, J., Crisp, D., 1997. The Mars Pathfinder Atmospheric Structure Investigation/Meteorology (ASI/MET) Experiment

[SSWRC07] Sengupta A., Steltzner, A., Witkowski, A., Rowan, J., Cruz, J., 2007.

Overview of the Mars Science Laboratory Parachute Decelerator System

[T63] Taylor, G., 1963. On the shapes of Parachutes. In: The Scientific Papers of G.I. Taylor.

[ZS13] Zwicker, M., Sinclair, R., 2013. Pack Density Liminations of Hybrid Parachutes

Appendix

A – Sample Ggrid computation log (first 200 sweeps)

```
SCHEDULING GRID GENERATION:

>::::: Mon Dec 18 21:33:56 2017 :::::
>Scheduling at step 1(swp 0) with 5 actions..
+action: (-c all 1.0 1.2) set algebraic cluster parameters.
   3 collected, all reset: ratio=1 growth ratio=1.2
+action: (-C all 1.0 3) set surf cluster control parameters.
   3 collected, all reset: ratio=1 radius=3
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.0064 0.0066), r(8.3e+03 2.4e+04)],K[s(0.0096 0.0091), r(2.1e+03 7.2e+02)]
+action: (-r) readjust surf grid
 (0) are reap..ed.
+action: (-S 100) new sweep interval = 100
swp 0 0(3{2},1447) max(r69.2,c{f175(0.0):e27(0.0)},s9.93e+31(0.0023),a40.),act(B100,V100,F100),cri(23o,12s)
swp 1 0(3{2},124) max(r309.,c{f175(3.5):e19(3.3)},s3.12e+31(0.0043),a36.),act(B100,V100,F100),cri(20s)
  phase 0 1 -> 1.
swp 2 0(1{2},0) 1(2{1},370) max(r252.,c{f173(3.5):e17(4.2)},s18.8(0.0059),a33.),act(B100,V13,F20),cri(13s)
  phase 2 -> 1.
swp 3 1(3{1},300) max(r166.,c{f173(3.5):e19(4.4)},s16.2(0.0069),a35.),act(B85,V8,F18),cri(17s)
  phase 2 -> 2.
swp 4 1(2{1},282) max(r59.3,c{f174(3.5):e54(4.4)},s23.2(0.0080),a44.),act(B75,V7,F18),cri(8o,11s)
swp 5 1(2{1},217) max(r78.7,c{f177(3.5):e22(4.3)},s13.0(0.0090),a1.3e+02),act(B75,V7,F17),cri(4o,13s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.012 0.012), r(6.7e+03 6.7e+03)],K[s(0.017 0.016), r(4.4e+03 4.7e+03)]
swp 6 1(2{1},167) max(r47.7,c{f177(3.5):e15(4.1)},s8.24(0.0098),a80.),act(B57,V6,F16),cri(6o,10s)
swp 7 1(2{1},141) max(r78.0,c{f173(3.5):e14(4.0)},s7.12(0.011),a96.),act(B64,V22,F44),cri(58o,9s)
swp 8 1(2{1},154) max(r278.,c{f175(3.5):e14(3.8)},s8.82(0.011),a2.8e+02),act(B60,V26,F47),cri(91o,9s)
swp 9 1(2{1},91) max(r67.6,c{f170(3.5):e14(4.1)},s25.5(0.012),a95.),act(B60,V17,F34),cri(11o,12s)
swp 10 1(2{1},77) max(r86.0,c{f178(3.5):e13(3.8)},s15.8(0.013),a1.0e+02),act(B67,V15,F25),cri(5o,10s)
  phase 0 -> 2.
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.016 0.016), r(2.3e+03 1.3e+03)],K[s(0.028 0.022), r(7e+03 1.3e+04)]
swp 11 1(1{1},68) max(r85.6,c{f172(3.5):e13(4.1)},s14.3(0.014),a1.1e+02),act(B75,V16,F27),cri(6o,12s)
swp 12 1(1{1},63) max(r186.,c{f171(3.5):e14(4.4)},s14.8(0.014),a1.8e+02),act(B71,V16,F27),cri(17o,8s)
swp 13 1(1{1},79) max(r1.11e+03,c{f171(3.5):e14(4.5)},s19.3(0.015),a1.4e+02),act(B78,V28,F30),cri(16o,11s)
swp 14 1(1{1},78) max(r192.,c{f174(3.5):e14(4.4)},s46.0(0.015),a1.5e+02),act(B85,V34,F30),cri(10o,10s)
swp 15 1(1{1},59) max(r41.5,c{f169(3.5):e14(4.5)},s23.6(0.016),a1.3e+02),act(B75,V26,F25),cri(17o,10s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.022 0.019), r(2.6e+03 1.4e+03)],K[s(0.04 0.031), r(5.6e+03 1.2e+04)]
swp 16 1(1{1},21) max(r245.,c{f175(3.4):e14(3.5)},s16.1(0.016),a1.4e+02),act(B75,V18,F20),cri(27o,7s)
  phase 1 -> 2.
swp 17 max(r68.4,c{f176(3.5):e14(3.6)},s14.8(0.017),a2.4e+02),act(B64,V14,F23),cri(20o,9s)
swp 18 max(r281.,c{f173(3.4):e14(3.7)},s10.0(0.017),a2.9e+02),act(B53,V11,F23),cri(23o,10s)
swp 19 max(r169.,c{f176(3.5):e14(3.7)},s8.02(0.018),a3.1e+02),act(B67,V14,F24),cri(19o,13s)
```

```
  s 0(1:0) '..Hypersonics Project/test/Orion 3D CAD.STL.tmp' fold or confused on
   F0 c2 F9 f5 F10 f12 F12 f2
surf 1(1:1) '-ellip' fold on
 F58 f64 F62 f41 F65 f36 F68 f53 F69 f29 F74 f8 F77  f9  F80  f12
 F81 f9 F86 f8 F89 f6 F92 f6 F94 f8
surf 2(1:2) '-ellip' fold on
 F34 f20 F37 f15 F41 f19 F42 f21 F75 f56 F78 f56 F82 f56 F83 f56
  surf conn: 0 adjusted, 2 confused and 607 folded.
swp 20 max(r155.,c{f177(3.4):e14(3.8)},s16.2(0.018),a5.2e+02),act(B64,V22,F27),cri(17o,14s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.027 0.023), r(1.7e+03 1.6e+03)],K[s(0.051 0.048), r(4.3e+03 1.2e+04)]
swp 21 max(r85.3,c{f63(3.4):e14(3.7)},s19.3(0.018),a2.0e+02),act(B117,V39,F30),cri(10o)
swp 22 max(r29.4,c{f61(3.2):e14(3.8)},s16.7(0.018),a1.4e+02),act(B117,V25,F21),cri(15o)
swp 23 max(r127.,c{f177(3.2):e14(3.1)},s8.50(0.018),a1.1e+02),act(B110,V30,F21),cri(32o,12s)
swp 24 max(r136.,c{f56(3.4):e14(3.3)},s10.0(0.019),a1.2e+02),act(B107,V27,F22),cri(28o)
swp 25 max(r42.2,c{f177(3.3):e14(3.4)},s14.3(0.020),a82.),act(B100,V24,F22),cri(23o,16s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.032 0.026), r(2.6e+03 3.6e+03)],K[s(0.061 0.066), r(2.6e+03 1.2e+04)]
swp 26 max(r3.70e+04,c{f53(3.4):e14(3.4)},s12.8(0.021),a85.),act(B142,V34,F19),cri(25o)
swp 27 max(r270.,c{f168(3.2):e14(3.2)},s20.5(0.021),a1.0e+02),act(B103,V29,F26),cri(24o,15s)
swp 28 max(r32.2,c{f167(3.4):e14(3.1)},s14.6(0.022),a97.),act(B125,V27,F25),cri(25o,14s)
swp 29 max(r1.03e+03,c{f52(3.4):e14(3.1)},s6.69(0.023),a1.0e+02),act(B107,V26,F23),cri(21o)
swp 30 max(r74.1,c{f163(3.2):e14(3.0)},s13.9(0.021),a96.),act(B100,V26,F23),cri(10o,7s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.036 0.031), r(2.9e+03 6.5e+03)],K[s(0.071 0.084), r(1.8e+03 1.1e+04)]
swp 31 max(r547.,c{f53(3.4):e14(3.0)},s15.4(0.022),a85.),act(B114,V27,F19),cri(4o)
swp 32 max(r40.5,c{f173(3.3):e14(3.1)},s9.75(0.023),a73.),act(B96,V20,F20),cri(9o,15s)
swp 33 max(r90.4,c{f53(3.4):e14(3.1)},s9.58(0.023),a70.),act(B107,V27,F20),cri(16o)
swp 34 max(r330.,c{f169(3.2):e14(2.7)},s12.7(0.023),a89.),act(B92,V25,F24),cri(16o,8s)
swp 35 max(r60.7,c{f53(3.3):e15(2.6)},s35.3(0.021),a62.),act(B125,V24,F19),cri(9o)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.04 0.035), r(3.5e+03 9.2e+03)],K[s(0.081 0.1), r(1.4e+03 1.1e+04)]
swp 36 max(r303.,c{f171(3.3):e15(2.6)},s11.2(0.024),a81.),act(B96,V19,F18),cri(8o,10s)
swp 37 max(r35.3,c{f164(3.4):e15(2.6)},s29.0(0.024),a71.),act(B103,V20,F16),cri(4o,10s)
swp 38 max(r4.17e+03,c{f171(3.4):e15(2.6)},s8.69(0.023),a70.),act(B85,V17,F18),cri(9o,8s)
  s 0(1:0) '..Hypersonics Project/test/Orion 3D CAD.STL.tmp' fold or confused on
   F0 c2
surf 1(1:1) '-ellip' fold on
 F58 f21 F62 f4 F65 f2 F68 f9 F69  f1  F74  f20 F77  f29 F80  f32
 F81 f25 F86 f6 F89 f3 F92 f2 F94 f3
surf 2(1:2) '-ellip' fold on
 F75 f56 F78 f54 F82 f56 F83 f54
  surf conn: 0 adjusted, 2 confused and 377 folded.
swp 39 max(r785.,c{f45(3.4):e15(2.7)},s30.2(0.023),a75.),act(B89,V18,F14),cri(11o)
swp 40 max(r392.,c{f44(2.9):e15(2.7)},s28.3(0.024),a61.),act(B85,V16,F17),cri(8o)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.045 0.039), r(6.6e+03 1.6e+04)],K[s(0.09 0.12), r(1.1e+03 1.1e+04)]
swp 41 max(r282.,c{f176(2.9):e15(2.7)},s9.21(0.024),a80.),act(B96,V19,F21),cri(10o,11s)
```

swp 42 max(r2.36e+03,c{f48(3.3):e15(2.5)},s10.2(0.024),a68.),act(B117,V24,F17),cri(4o)
swp 43 max(r83.7,c{f53(3.2):e15(2.5)},s9.70(0.024),a75.),act(B92,V18,F15),cri(5o)
swp 44 max(r18.6,c{f172(3.3):e15(2.5)},s14.0(0.024),a50.),act(B78,V19,F20),cri(3o,9s)
swp 45 max(r50.1,c{f45(3.3):e15(2.5)},s10.1(0.024),a57.),act(B125,V22,F13),cri(4o)
set static ctrl funcs... (0.4) (0)

s2 :IJ[s(0.048 0.044), r(9.6e+02 4.4e+03)],K[s(0.099 0.13), r(8.3e+02 1.1e+04)]
swp 46 max(r45.6,c{f32(3.0):e15(2.5)},s11.4(0.024),a53.),act(B82,V14,F10),cri(3o)
swp 47 max(r53.1,c{f175(3.2):e15(2.5)},s10.8(0.024),a44.),act(B64,V11,F16),cri(4o,16s)
swp 48 max(r7.39,c{f34(3.3):e15(2.5)},s10.9(0.024),a70.),act(B110,V21,F13),cri(5o)
swp 49 max(r5.48,c{f43(3.0):e15(2.5)},s12.1(0.024),a44.),act(B85,V15,F11),cri(1o)
swp 50 max(r5.80,c{f175(3.1):e15(2.5)},s12.6(0.024),a65.),act(B67,V11,F16),cri(3o,9s)
set static ctrl funcs... (0.4) (0)

s2 :IJ[s(0.052 0.049), r(6.3e+02 2.8e+03)],K[s(0.11 0.15), r(7e+02 1.1e+04)]
swp 51 max(r4.07,c{f169(3.3):e15(2.5)},s12.9(0.024),a45.),act(B100,V18,F13),cri(4o,12s)
swp 52 max(r1.04,c{f178(3.3):e15(2.5)},s13.7(0.024),a41.),act(B92,V16,F14),cri(15s)
swp 53 max(r17.6,c{f175(3.3):e15(2.5)},s13.9(0.024),a42.),act(B85,V14,F12),cri(1o,11s)
swp 54 max(r0.730,c{f36(3.3):e15(2.5)},s14.0(0.024),a41.),act(B85,V12,F12),cri(1o)
swp 55 max(r2.89,c{f52(2.9):e15(2.5)},s14.8(0.024),a38.),act(B96,V15,F14),cri(3o)
set static ctrl funcs... (0.4) (0)

s2 :IJ[s(0.056 0.054), r(5.4e+02 2.9e+03)],K[s(0.12 0.16), r(5.8e+02 1e+04)]
swp 56 max(r0.830,c{f173(3.1):e15(2.5)},s15.2(0.024),a38.),act(B92,V20,F15),cri(12s)
swp 57 max(r6.06,c{f52(3.3):e15(2.4)},s15.4(0.024),a36.),act(B92,V15,F15),cri(1o)
s 0(1:0) '..Hypersonics Project/test/Orion 3D CAD.STL.tmp' fold or confused on
F0 c1
surf 1(1:1) '-ellip' fold on
F74 f15 F77 f26 F80 f29 F81 f21 F86 f1
surf 2(1:2) '-ellip' fold on
F75 f54 F78 f46 F82 f56 F83 f48
surf conn: 0 adjusted, 1 confused and 296 folded.
swp 58 max(r0.728,c{f52(2.9):e15(2.4)},s15.5(0.024),a44.),act(B96,V16,F13)
swp 59 max(r27.8,c{f168(3.2):e15(2.4)},s15.7(0.024),a37.),act(B89,V15,F11),cri(3o,11s)
swp 60 max(r0.863,c{f52(3.3):e15(2.4)},s15.7(0.024),a37.),act(B67,V14,F15)
set static ctrl funcs... (0.4) (0)

s2 :IJ[s(0.059 0.06), r(4.2e+02 2.1e+03)],K[s(0.13 0.17), r(4.9e+02 1e+04)]
swp 61 max(r1.19,c{f176(3.0):e15(2.4)},s15.6(0.024),a37.),act(B96,V15,F11),cri(19s)
swp 62 max(r0.758,c{f50(3.3):e15(2.4)},s15.9(0.024),a38.),act(B60,V10,F11)
swp 63 max(r0.768,c{f52(3.2):e15(2.4)},s15.5(0.024),a35.),act(B82,V19,F19)
swp 64 max(r2.81,c{f172(3.1):e15(2.4)},s15.0(0.024),a35.),act(B125,V19,F17),cri(2o,8s)
swp 65 max(r8.12,c{f176(3.3):e15(2.4)},s15.5(0.024),a36.),act(B78,V11,F11),cri(1o,16s)
set static ctrl funcs... (0.4) (0)

s2 :IJ[s(0.062 0.065), r(3.8e+02 2e+03)],K[s(0.14 0.19), r(4.3e+02 9.8e+03)]
swp 66 max(r0.780,c{f52(3.3):e15(2.4)},s15.4(0.024),a33.),act(B71,V13,F15)
swp 67 max(r45.1,c{f174(3.1):e15(2.4)},s14.6(0.024),a32.),act(B78,V12,F18),cri(17s)
swp 68 max(r0.809,c{f52(3.2):e15(2.3)},s14.5(0.024),a30.),act(B121,V22,F15)
swp 69 max(r0.658,c{f52(3.1):e15(2.3)},s13.8(0.024),a28.),act(B100,V19,F14)
swp 70 max(r0.833,c{f172(3.1):e15(2.3)},s12.9(0.024),a27.),act(B121,V22,F13),cri(14s)
set static ctrl funcs... (0.4) (0)

```
  s2 :IJ[s(0.066 0.071), r(3.6e+02 2.1e+03)],K[s(0.15 0.2), r(4e+02 9.3e+03)]
swp 71 max(r0.907,c{f50(3.3):e15(2.3)},s13.4(0.024),a27.),act(B78,V13,F14)
swp 72 max(r0.643,c{f52(3.1):e15(2.3)},s13.2(0.024),a26.),act(B78,V14,F12)
swp 73 max(r1.79,c{f174(3.0):e15(2.3)},s13.1(0.024),a25.),act(B96,V18,F11),cri(16s)
swp 74 max(r0.896,c{f52(3.3):e15(2.3)},s12.6(0.024),a26.),act(B85,V17,F14)
swp 75 max(r12.6,c{f173(3.1):e15(2.3)},s12.6(0.024),a25.),act(B107,V18,F17),cri(1o,11s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.069 0.076), r(3.4e+02 2.2e+03)],K[s(0.15 0.21), r(3.7e+02 9e+03)]
swp 76 max(r0.560,c{f47(3.2):e15(2.3)},s12.1(0.024),a24.),act(B125,V22,F14)
  s 0(1:0) '..Hypersonics Project/test/Orion 3D CAD.STL.tmp' fold or confused on
   F0 c1
surf 1(1:1) '-ellip' fold on
 F74 f4 F77 f15 F80 f14 F81 f5
surf 2(1:2) '-ellip' fold on
 F75 f36 F78 f36 F82 f41 F83 f36
  surf conn: 0 adjusted, 1 confused and 187 folded.
swp 77 max(r1.24,c{f176(2.9):e15(2.3)},s12.0(0.024),a24.),act(B100,V21,F19),cri(1o,13s)
swp 78 max(r1.31,c{f176(3.3):e15(2.3)},s11.6(0.024),a24.),act(B121,V21,F17),cri(15s)
swp 79 max(r0.761,c{f52(3.3):e14(2.3)},s11.2(0.024),a23.),act(B96,V15,F15)
swp 80 max(r0.852,c{f171(3.1):e14(2.3)},s11.3(0.024),a23.),act(B132,V24,F18),cri(10s)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.072 0.082), r(3.2e+02 2.3e+03)],K[s(0.16 0.23), r(3.4e+02 8.4e+03)]
swp 81 max(r1.75,c{f170(3.3):e15(2.3)},s11.0(0.024),a22.),act(B142,V28,F15),cri(14s)
swp 82 max(r0.621,c{f47(3.2):e14(2.3)},s10.5(0.024),a22.),act(B121,V20,F15)
swp 83 max(r1.59,c{f174(3.3):e14(2.3)},s10.7(0.024),a21.),act(B107,V19,F18),cri(15s)
swp 84 max(r0.742,c{f40(3.3):e14(2.3)},s10.3(0.024),a21.),act(B128,V25,F17)
swp 85 max(r0.819,c{f52(3.3):e14(2.2)},s10.1(0.024),a21.),act(B125,V22,F15)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.075 0.088), r(3.1e+02 2.3e+03)],K[s(0.18 0.24), r(3.3e+02 7.7e+03)]
swp 86 max(r2.53,c{f176(3.1):e14(2.2)},s10.0(0.024),a21.),act(B128,V26,F16),cri(13s)
swp 87 max(r0.628,c{f52(3.2):e14(2.2)},s9.93(0.024),a21.),act(B125,V22,F14)
swp 88 max(r17.7,c{f176(2.9):e14(2.2)},s9.69(0.024),a20.),act(B117,V18,F16),cri(3o,16s)
swp 89 max(r0.752,c{f173(3.2):e14(2.2)},s9.44(0.024),a20.),act(B142,V26,F14),cri(15s)
swp 90 max(r0.676,c{f49(3.2):e14(2.2)},s9.32(0.024),a20.),act(B96,V17,F15)
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.078 0.093), r(3.1e+02 2.1e+03)],K[s(0.19 0.25), r(3.2e+02 7.2e+03)]
swp 91 max(r0.662,c{f42(2.9):e14(2.2)},s9.26(0.024),a20.),act(B107,V21,F18)
swp 92 max(r2.74,c{f176(3.0):e14(2.2)},s9.37(0.024),a20.),act(B128,V23,F21),cri(13s)
swp 93 max(r0.514,c{f51(3.2):e14(2.2)},s9.29(0.024),a20.),act(B135,V25,F16)
swp 94 max(r0.660,c{f52(3.3):e14(2.2)},s8.84(0.024),a19.),act(B96,V16,F17)
swp 95 max(r12.8,c{f176(3.3):e14(2.2)},s8.95(0.024),a19.),act(B100,V17,F18),cri(1o,13s)
  s 0(1:0) '..Hypersonics Project/test/Orion 3D CAD.STL.tmp' fold or confused on
   F0 c2
surf 2(1:2) '-ellip' fold on
 F75 f1 F78 f2 F82 f5 F83 f9
  surf conn: 0 adjusted, 2 confused and 17 folded.
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.081 0.099), r(3.1e+02 2e+03)],K[s(0.2 0.26), r(3.3e+02 6.7e+03)]
```

```
swp 96 max(r0.491,c{f42(3.2):e14(2.2)},s9.04(0.024),a19.),act(B121,V22,F13)
swp 97 max(r0.780,c{f173(3.3):e14(2.2)},s8.62(0.024),a19.),act(B89,V19,F17),cri(15s)
swp 98 max(r5.38,c{f176(3.2):e14(2.2)},s8.73(0.024),a19.),act(B107,V20,F22),cri(1o,16s)
swp 99 max(r0.613,c{f52(3.2):e14(2.2)},s8.86(0.024),a18.),act(B128,V23,F13)
Amount of time taken for 100 sweeps is 0 hours 0 minutes 1 seconds
+action: (-w) output blocks
st:dep=5/5,dir=1111/2000
st:found 0. too bad.
  all to 'blk-orion-capsule.tmp': block 1/28: snap 0/81...
  all to 'blk-orion-capsule.tmp':snap 151/2276    block    1/28...
  all to 'blk-orion-capsule.tmp':snap 151/2276   block   28/28...
  all to 'blk-orion-capsule.tmp':snap 151/2276

Write action took 0 hours 0 minutes 0 seconds
  all to 'dump.tmp': block 1/28...
  all to 'dump.tmp': block 28/28...
  all to 'dump.tmp':

Write action took 0 hours 0 minutes 0 seconds


>::::: Mon Dec 18 21:33:59 2017 :::::
>Scheduling at step 2(swp 100) with 5 actions..
+action: (-c all 1.0 1.2) set algebraic cluster parameters.
    3 collected, all reset: ratio=1 growth ratio=1.2
+action: (-C all 1.0 3) set surf cluster control parameters.
    3 collected, all reset: ratio=1 radius=3
set static ctrl funcs... (0.4) (0)

  s2 :IJ[s(0.084 0.1), r(3e+02 1.9e+03)],K[s(0.21 0.27), r(3.4e+02 6.3e+03)]
+action: (-r) readjust surf grid
 (0) are reap..ed.
+action: (-S 100) new sweep interval = 100
swp 100 max(r15.0,c{f176(3.3):e14(2.2)},s8.70(0.024),a18.),act(B57,V11,F17),cri(1o,16s)
set static ctrl funcs... (0.4) (0)
```