Best Practices for Unstructured Grid Shock Fitting -ROUGH DRAFT-

Peter L. McCloud*

ERC-Inc, Houston, TX, 77571

Unstructured grid solvers have well-known issues predicting surface heat fluxes when strong shocks are present. Various efforts have been made to address the underlying numerical issues that cause the erroneous predictions. The present work addresses some of the shortcomings of unstructured grid solvers, not by addressing the numerics, but by applying structured grid best practices to unstructured grids. A methodology for robust shock detection and shock fitting is outlined and applied to production relevant cases. Results achieved by using the Loci-CHEM Computational Fluid Dynamics solver are provided.

I. Introduction

Traditionally, Computational Fluid Dynamics (CFD) for hypersonic conditions have been solved using codes that rely on structured grids consisting of point-matched blocks. Codes such as LAURA[1,2] and DPLR[3–5] have long been considered the standard for hypersonic applications. The requirement to have point-matched structured grids can make grid generation tedious or nearly impossible when the geometry becomes complex. To address grid generation complexity, efforts are being made to create hypersonic CFD codes that utilize either structured overset grids[6,7] or unstructured grids. Unstructured grids offer the promise of easier grid generation processes, but solving hypersonic problems on unstructured grids has inherent numerical complications. Satisfactory inviscid flux formulations are currently an on-going area of research.

Two unstructured grids solvers currently being developed to be the next generation of hypersonic solvers are US3D[8] and FUN3D.[9] While US3D has shown great promise for unstructured grids, it should be noted that the preferred grid topology for US3D is to use purely hexahedrals to take advantage of the higher order methods. US3D also requires that a cell connectivity map be created.

The present work sidesteps the underlying numerical issues and addresses the issues of solving hypersonic problems on unstructured grids by adopting structured grid solver best practices. This requires creating a more complex grid topology, but the flexibility of unstructured grid generation can easily incorporate the increased complexity. A methodology of performing unstructured shock fitting on production relevant cases was developed, tested and verified against test data.

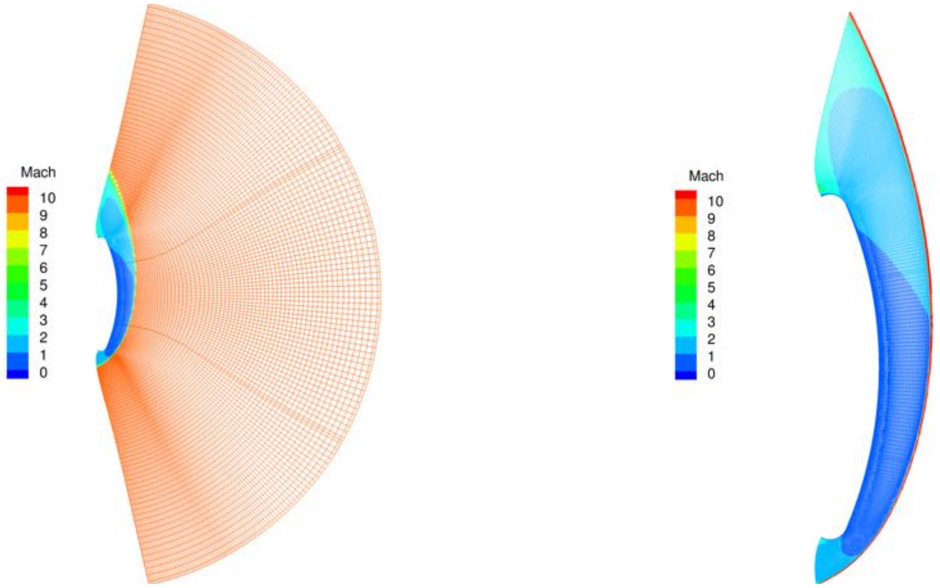
While the framework built for performing the unstructured shock adaption was built to be solver agnostic, and can even work with structured grid solvers, the results shown where computed using the Loci/CHEM[10,11] CFD solver. Loci/CHEM is an unstructured, finite-rate chemistry solver developed at Mississippi State University (MSU) by Dr. Ed Luke.

II. Structured Shock Fitting Methodology

Structured grids for hypersonic problems are typically point-matched and the coordinates are labeled i, j, and k, with the k direction being the off-body direction. Once a solution is obtained on the initial grid, the shock fitting process is started. The user typically specifies a percentage of the free stream Mach number. For DPLR, this is typically 95%. The adaptation algorithm analyzes the k-lines and determines where the Mach number reaches the specified value. This location then represents the shock location. It should be noted that the identified points on all of the k-lines essentially represent a Mach iso-surface, a property that is

*Aerothermodynamics Engineer, JETS-EG3 NASA JSC, Lifetime Member utilized in the unstructured process. With the shock surface identified, the locations are typically smoothed relative to each other. Finally, the k-lines are then modified such that the k_{max} point lies just outside of the shock with a minimal margin for maximum computational efficiency.

Example results of the structured shock fitting methodology are provided in figure 1. The example shows results from an axisymmetric heatshield forebody at Mach 10, with the flow impinging the heatshield at a specified angle of attack. The first image shows the center-line Mach iso-contour for the initial unadapted grid. The second image shows the center-line Mach iso-contour for the final shock-adapted grid. These two images together show how the k-lines are modified such that the end points lie just beyond the bow shock.



(a) Center-line Mach iso-contour for the initial unadapted grid.

(b) Center-line Mach iso-contour for the final shock-adapted grid.

Figure 1. Before and after shock fitting results on an axisymmetric heatshield forebody at Mach 10 using structured grids.

The drawback to this methodology is that it does not allow for the existence of internal shocks or the presence of plumes in the flow field. As the algorithm follows the k-lines and encounters a location where the Mach number meets the criteria, even if not the bow shock, it will treat that feature as the bow shock and the process will fail.

III. Unstructured Shock Fitting Methodology

III.A. Process

The overall goal of the shock fitting process is to create a grid with element faces that are aligned to the shock, decreasing the numerical noise. The present work concentrated on developing a shock fitting methodology that leveraged structured grid best practices and made the process solver agnostic wherever possible in order for it to be applicable to a wide range of tools. The shock fitting process developed for the present work is outlined in figure 2. The process is composed of an iterative loop composed of several sub-steps with the first loop starting from the initial solution.

The first step of the loop is applying a shock detection algorithm to build a surface that represents the shock. Next, the surface is smoothed to better align the cell faces to the shock. To maximize the cell quality at the shock, the shock surface is then re-meshed. With the shock surface finalized, a prismatic layer is then extruded on both sides of the surface. The volume grid is then re-meshed using the prismatic layer as a new outer boundary. The final step of the loop is to solve for the updated solution. The loop is repeated until the solution achieves sufficient accuracy.

Implementation of the unstructured shock fitting methodology is part of the *mesh_tools* tool suite developed at the NASA Johnson Space Center (JSC) Aerosciences Branch. The tool suite is a collection of Python scripts and modules that utilize the Visualization Tool Kit (VTK).[12] The VTK libraries provide efficient data structures and algorithms that maximize the performance of the Python scripts and utilize a pipeline process where filters are applied to data sets to obtain the desired results.

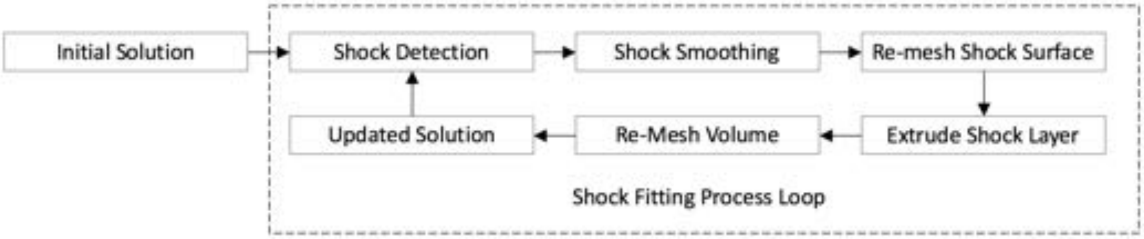


Figure 2. Shock fitting process.

III.A.1. Initial Solution

As with structured grids, the first step in the process is to obtain an initial grid and solution. The present work utilized Altair's Hypermesh[19] to build a quad-dominant surface mesh and AFLR3[13, 14] to build the volume mesh. The initial volume mesh consists of a prismatic layer grown from the viscous walls and the remaining volume is filled with tetrahedra. Obtaining the initial solution can sometimes be problematic due to carbuncles forming at the bow shock location. Typically, the best results are obtained by averaging the solution over of a number of iterations to account for the shock movement due to carbuncles.

III.A.2. Shock Detection

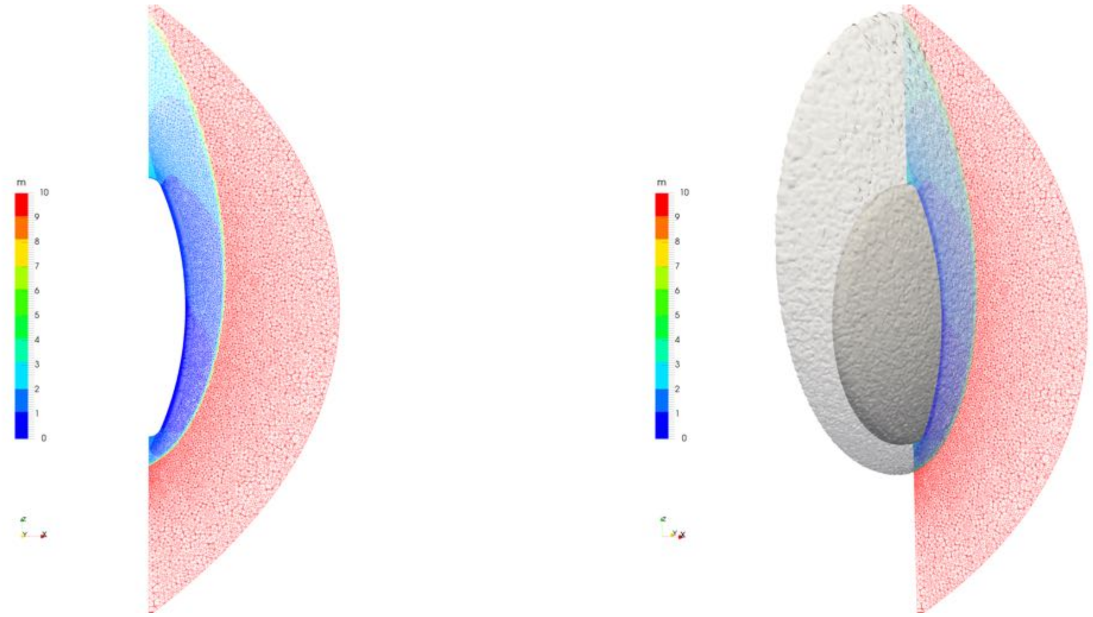
One of the main challenges with the unstructured shock fitting is identifing the shock location efficiently without the connectivity information inherent in structured grids. As mentioned previously, the structured shock fitting process is essentially identifying a Mach iso-contour. To start the shock fitting process, the solution is read in and the Mach iso-contour is then found simply by applying the *vtkContourFilter*. For a typical smooth body hypersonic case, the resulting Mach iso-contour represents the bow shock. An example of the shock detection results are shown in figure 3. The example shown is for the same axisymmetric heatshield shown in figure 1, but run on an unstructured grid instead. The first image shows the initial center-line Mach iso-contour. The second image shows the result of the shock detection process using a Mach iso-contour set to 95% of the freestream Mach number.

This simplified approach to identifying a hypersonic bow shock breaks down when internal shocks or plumes are present in the flow field. However, additional processing of the iso-contour can provide the isolated surface that represents the bow shock. The connectivity of iso-contour is first analyzed, using the *vtkPolyDataConnectivityFilter*. This allows the iso-contour to be broken into separate surfaces, each associated with the various flow phenomena (bow shock, internal shock, plume). Identification of the bow shock is achieved by examining the bounding box for each of the surfaces that make up the iso-contour. For a hypersonic problem, the bow shock will encompass all of the other flow features, so the bounds of the bow shock will match the bounds of the iso-contour as a whole. This approach is particularly efficient since the bounding box information is automatically computed when the VTK data structures are created.

An example of a complex case with internal flow features is shown in figure 4. The case shown is for the Orion Exploration Flight Test (EFT)-1 launch configuration. The Launch Abort Vehicle (LAV) that surrounds the Orion capsule during ascent has large cavities and protrusions that create numerous internal shocks. This case was chosen because it's currently not possible to perform shock adaption on this geometry using either DPLR or LAURA. The first image shows the center-line Mach iso-contour results for the initial solution. The second image shows the initial step of the shock detection process and the internal flow features present. The last image shows the final results of the shock detection process where the internal flow features have been removed using the outlined process.

III.A.3. Shock Smoothing

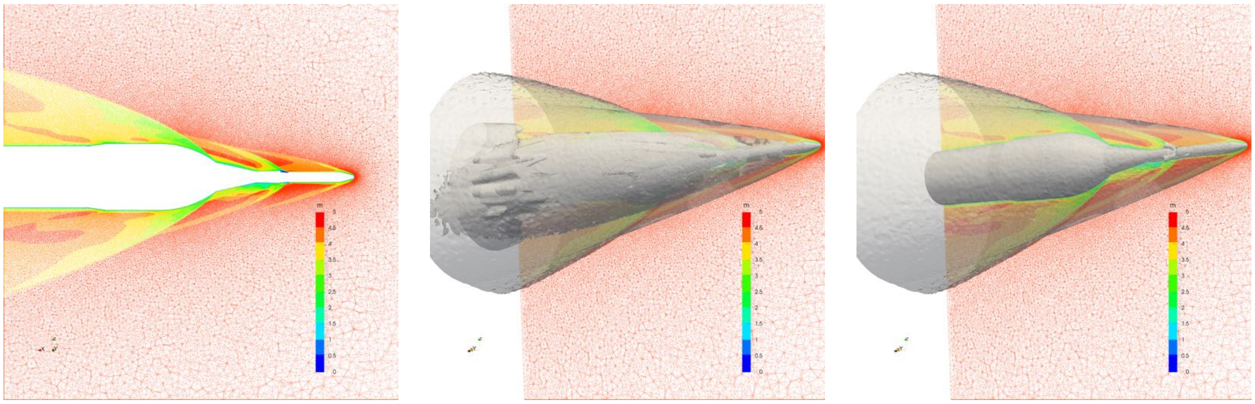
The overall goal of the shock fitting process is to create a grid with element faces that are aligned to the shock, decreasing the numerical noise. The raw shock surface identified in the prior step can be quite noisy,



(a) Center-line Mach iso-contour for the initial unadapted unstructured grid

(b) Center-line Mach iso-contour with a Mach iso-surface set to 95% of the freestream Mach number

Figure 3. Shock detection results for a axisymmetric heatshield forebody at Mach 10 using unstructured grids.



(a) Center-line Mach iso-contour.

(b) Center-line Mach iso-contour with a Mach iso-surface set to 95% of the freestream Mach number, with internal features present.

(c) Center-line Mach iso-contour with a Mach iso-surface set to 95% of the freesteam Mach number, with internal features removed.

Figure 4. Iso-contour results for a complex case, the Orion EFT-1 launch configuration.

particularly when the shock lies within a region of tetrahedrons, as seen in figure 3. This noise is eliminated by smoothing the raw shock surface so that it is aligned well to the physical shock.

The smoothing algorithms used for structured shock fitting depend on the structured nature of the grids and aren't applicable to unstructured surfaces. Therefore new algorithms had to be identified for the present work. Testing was performed to idenfity a robust smoothing process that would work for a variety of cases. Algorithms evaluated for the smoothing process were:

- Laplacian
- Gaussian
- Taubin
- Mean Curvature
- Two-Step

Testing on a variety of cases found that a combination of algorithms applied in passes was the most robust approach. No single algorithm was able to provide a sufficiently smoothed surface. Gaussian smoothing has the unwanted side effect of shrinking the mesh, but is the best at removing particularly noisy regions. Taubin smoothing is perhaps the most conservative algorithm as it does not shrink the surface, but its effects tend to plateau as successive iterations are applied. The mean curvature algorithm provides the smoothest surface with the fewest number of iterations, without any shrinkage, but can become unstable easily. The Two-Step algorithm is the superior method for cleaning particularly noisy surfaces, but had the tendency to introduce noise in smoother surfaces.

Based on the above testing, a criterion was also developed for judging the amount of noise present in the surface so that the best smoothing process could be applied. Mean curvature was found to be a good metric of the quality of the shock surface. After the shock detection process, the filter *vtkCurvatures* is applied to compute the mean curvature. Based on the amount of noise present, one of two possible routines will be used. For noisy surfaces, such as when the shock lies in a field of tetrahedrons, the best combination was found to be a series of four passes, with each pass consisting of 20 initial Taubin iterations, followed by three iterations of the Two-Step algorithm and five Gaussian smoothing iterations. For smoother surfaces, the best combination was found to be successive passes with 25 Taubin iterations followed by 80 mean curvature smoothing iterations and 10 iterations of Laplacian smoothing. Both the Taubin and Laplacian stabilize the mean curvature iterations.

After each pass, the mean curvature is re-computed and the scripts evaluate if the most recent pass sufficiently improved the result. If no significant improvements in the surface quality are identified, the resulting surface is passed onto the next step in the process. If the quality is still improving significantly, the process is continued up to a maximum of 30 passes.

For the noisy surfaces, Meshlab[15] is externally called by the *mesh_tools* suite to do the surface smoothing. This is performed to utilize the Two-Step algorithm, which is as yet to be implemented internally. For smooth surfaces, all of the required algorithms have been the implemented internally to the package.

III.A.4. Re-Meshing the Shock Surface

The smoothing process creates an unstructured surface that has minimal noise in the mean curvature, but doesn't take into account the element sizing distribution. The smoothing process tends to create face sizes that lead to poor volume element quality. To get the best results from the unstructured shock fitting process, the smoothed shock surface needs to be re-meshed to have the highest volume element quality possible. To do this, the smoothed shock surface is passed externally to Gmsh,[16] an open source grid generation software. Gmsh then re-meshes the shock surface using a user defined spacing distribution. After the re-meshing is complete, the final shock surface grid is read back in and a check is made to ensure that the surface normals are orientated correctly.

III.A.5. Extruding a Prismatic Layer around the Shock Surface

Creating elements at the shock surface that have faces aligned with the shock has been demonstrated by Bonfiglioli, et al.[17] to improve the accuracy of the solution. For the present work, it was found that going a step further and creating layers of prismatic elements aligned with the shock provided the best results. The prismatic layer is formed by marching the re-meshed shock surface both upstream and downstream for several layers. The upstream layers are necessary to provide margin for the shock as it settles into the final location thoughout successive shock fitting loops. The number of layers required can depend on the problem, but for the present work 10 to 30 layers were typically used.

The prismatic layers were grown using the Pointwise[18] grid generation package. The *mesh_tools* scripts write out the appropriate Pointwise glyph script and run Pointwise externally. After Pointwise completes the extrusion process, it writes out the resulting volume grid and the downstream surface of the prismatic shock layer. This downstream surface will be referred to as the shock interface surface for the rest of the paper.

III.A.6. Rebuilding the Volume Grid

The final step of the unstructured shock fitting process is to rebuild the volume grid. A new outer boundary is created using the shock interface surface. The shock interface will be open on the downstream side and an exit surface is created to complete the water-tight outer boundary. Depending on the type of problem, the exit plane will either simply fill the hole in the shock interface surface, or bridge the space between the viscous walls and the shock interface surface. The completed outer boundary is then merged with the original wall surfaces to complete the boundary surfaces that define the volume. A new volume grid is then grown in a similar manner to the initial volume grid.

IV. Results

The unstructured shock fitting methodology outlined was applied to a variety of problems in a production capacity. While the *mesh_tools* scripts are not integral to any solver, they can be scripted to run alongside a solver and complete the process in a primarily automated manner. A sub-sample of results are presented below.

Most of the test cases were run with the Loci-CHEM solver. One of the test cases used to check out the unstructured shock fitting process is the smooth body axisymmetric heatshield with just the forebody. This is a class of problem that is solved more efficiently using a structured grid solver, but serves as an excellent test of the shock fitting capability.

The first case was run using laminar conditions, with perfect air and an iso-thermal wall boundary. After obtaining the initial solution, the shock fitting process was repeated for five loops. Figure 5 shows the center-line Mach iso-contour results for the initial solution and the final shock fit solution. For the initial solution, the shock lies in a region of tetrahedrons. The final results show that the shock adaption process has created a prismatic layer of cells, aligned well with the shock.

The axisymmetric heatshield forebody heat flux results are shown for each shock fit iteration in figure 6. The results show the large noise in the heat flux results on the initial grid. The first shock adaption step shows a dramatic improvement with the creation of the prismatic layer around the shock. Subsequent steps show further improvement in the heat flux results as the cell alignment is improved.

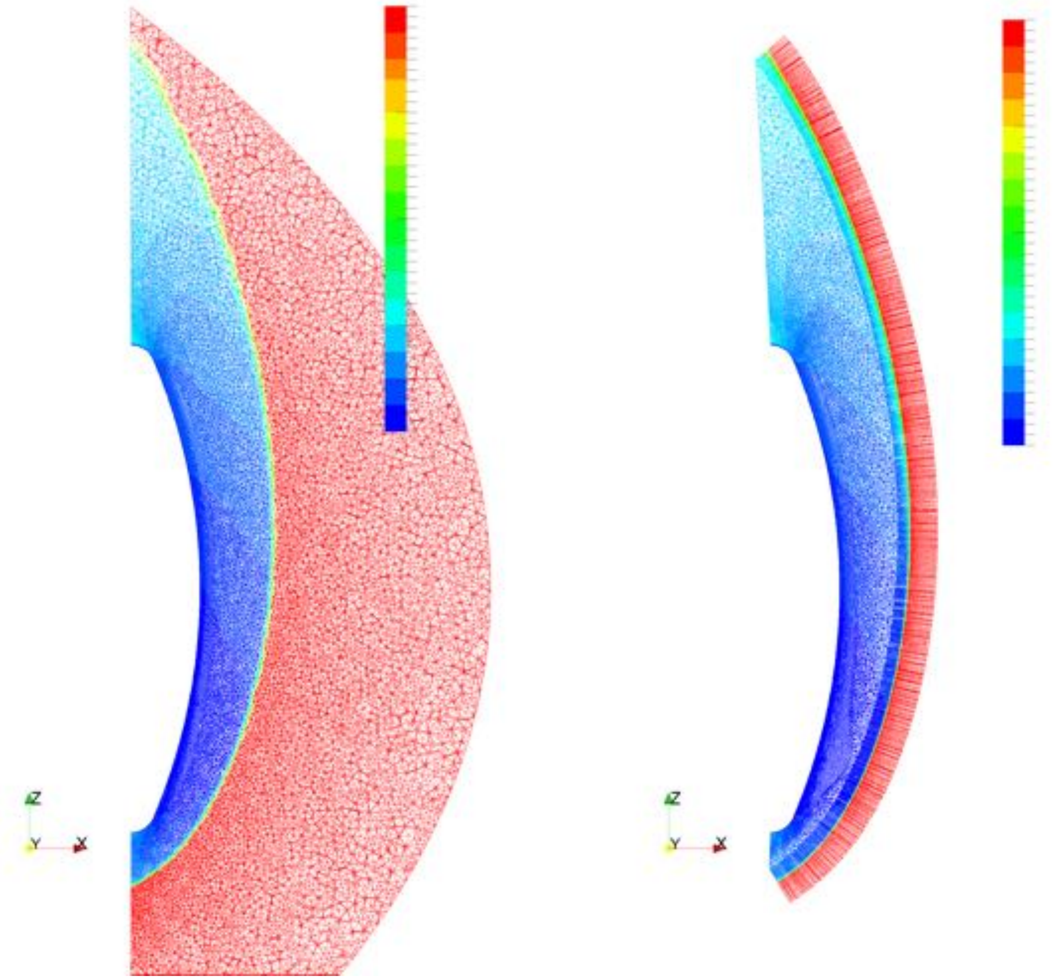
V. Conclusion

A process for unstructured shock fitting to improve unstructured hypersonic CFD predictions has been developed and tested. This process is robust and can handle both internal shocks and plumes that occur with complex geometries. It has been implemented inside of the *mesh_tools* suite and tested alongside the Loci-CHEM CFD solver. The process creates an efficient grid topology with the prismatic layers at the shock and viscous walls and tetrahedrons in the remaining regions. Results show a dramatic improvement in the heat flux predictions for unstructured hypersonic problems.

References

[1] Mazaheri, A., Gnoffo, P., Johnston, C., and Kleb, B., *LAURA Users Manual*, Tech. Rep. NASA TM 2010-216836, 2010.

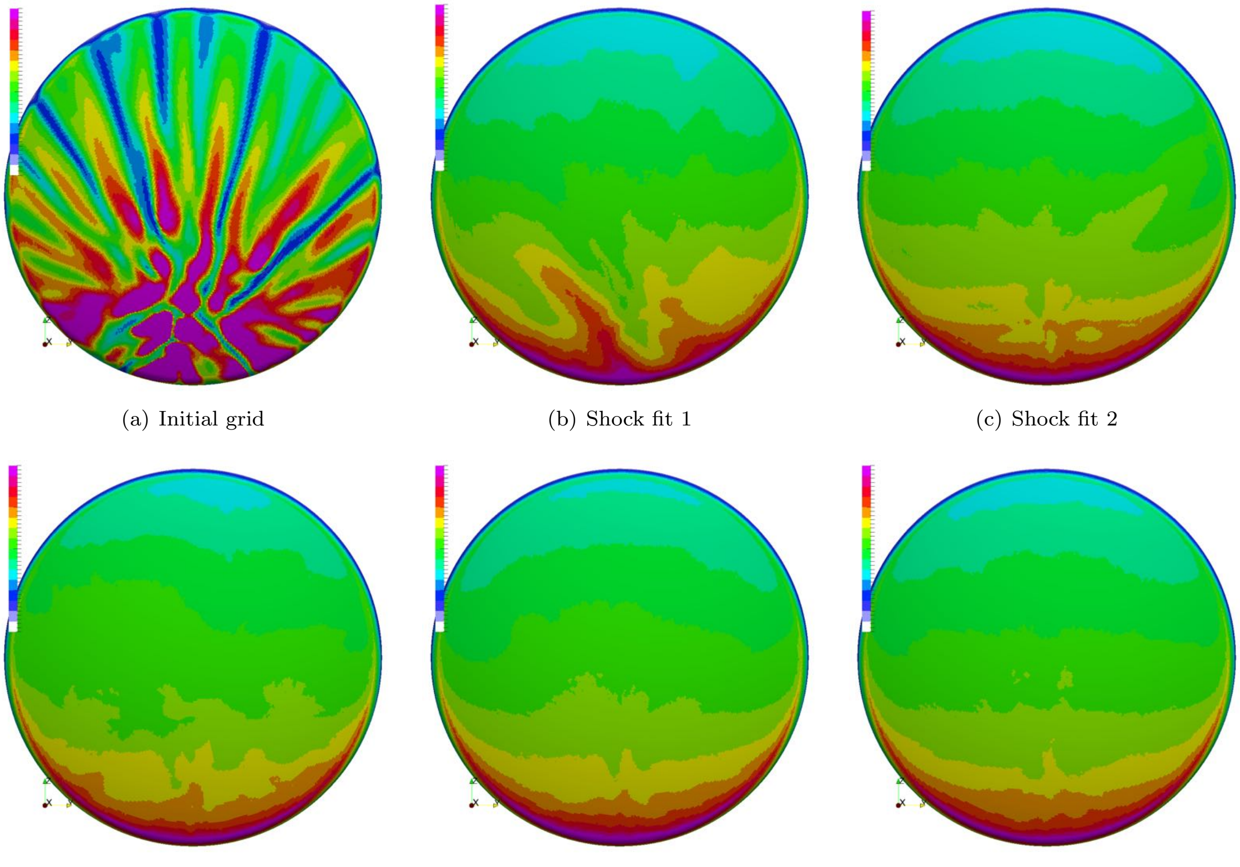
[2] Gnoffo, P., Gupta, R., and Shinn, J., *Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Nonequilibrium*, Tech. Rep. NASA TP-2867, 1989.



(a) Center-line Mach iso-contour for the initial unstructured grid.

(b) Center-line Mach iso-contour for the final shock-adapted unstructured grid.

Figure 5. Before and after shock fitting results on an axisymmetric heatshield forebody at Mach 10 using unstructured grids.



(a) Initial grid

(b) Shock fit 1

(c) Shock fit 2

(d) Shock fit 3

(e) Shock fit 4

(f) Shock fit 5

Figure 6. Surface heat flux results for the axisymmetric heatshield forebody at Mach 10 using unstructured grids.

[3] Wright, M., *A Family of Data-Parallel Relaxation Methods for the Navier-Stokes Equations*, Ph.D. thesis, University of Minnesota, 1997.

[4] Wright, M., Candler, G., and Bose, D., *Data-Parallel Line Relaxation Method for the Navier-Stokes Equations*,AIAA Journal, Vol. 36, No. 9, 1998, pp. 1603-1609, 1998.

[5] Wright, M., White, T., and Mangini, N., *Data-Parallel Line Relaxation (DPLR) Code User Manual Acadia-Version 4.01.1*, NASA/TM-2009-215388, NASA Ames Research Center, October 2009.

[6] Boger, D, Noack, R., Amar, A., Kirk, B., Lillard, R., Olsen, M., and Dries, K. *Overset Grid Applications in Hypersonic Flow Using the DPLR Flow Solver*, AIAA-2008-921, 46th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, 2008.

[7] Hyatt, A., Barnhardt, M., Prabhu, D., Mansour, N., and Boger, D. *Verification and Validation of Structured and Overset Grid Capabilities on Hypersonic Multibody Configurations*, AIAA-2010-4454 40th Fluid Dynamics Conference and Exhibit, Fluid Dynamics and Co-located Conferences, 2010.

[8] Candler, G., Subbareddy, P., and Brock, J. *Advances in Computational Fluid Dynamics Methods for Hypersonic Flows*, Journal of Spacecraft and Rockets 2015 52:1, 17-28, 2015.

[9] Biedron, R., et al. *FUN3D Manual: 12.9*, Tech. Rep. NASA TM 2016-219012 2016.

[10] Luke, E. and George, T. *Loci: A Rule-Based Framework for Parallel Multidisciplinary Simulation Synthesis*, J. Functional Program., Vol.15, No.3,2005, pp. 477-502, 2005.

[11] Luke, E. *On Robust and Accurate Arbitrary Polytope CFD Solvers (Invited)*, AIAA Paper 2007-3956, 2007.

[12] Schroeder, W., Martin, K., and Lorensen, B. *The Visualization Toolkit (4th ed.)*, Kitware, ISBN 978-1-930934-19-1, 2006.

[13] Marcum, D. *Unstructured grid generation using automatic point insertion and local reconnection*, In The Handbook of Grid Generation, Edited by J.F. Thompson, B. Soni, and N.P. Weatherill, chapter 18, pages 131. CRC Press, 1998.

[14] Marcum, D. and Weatherill, N. *Unstructured grid generation using iterative point insertion and local reconnection*, AIAA Journal, 33(9):16191625, 1995.

[15] Callieri, M., Ranzuglia, G., Dellepiane, M., Cignoni, P., and Scopigno, R. *Meshlab as a Complete Open Tool for the Integration of Photos and Colour with High-Resolution 3D Geometry Data*, CAA 2012 Conference Proceedings, pp. 406-416, 2013.

[16] Geuzaine, C. and Remacle, J. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009.

[17] Bonfiglioli, A., Grottadaurea, M., Paciorri, R., Sabetta, F., Onofri, M., and Bianchi, D. *Numerical Simulation of Hypersonic Flows Past Three-Dimensional Blunt Bodies Through an Unstructured Shock-Fitting Solver*, 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, International Space Planes and Hypersonic Systems and Technologies Conferences, 2011.

[18] *Pointwise*, http://www.pointwise.com/pw/, Accessed: 2016-05-13

[19] *HyperMesh*, http://www.altairhyperworks.com/product/HyperMesh, Accessed: 2016-05-13