Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, HI, July 11-15, 2022

ICCFD11-3205

Rapid Hypersonic Simulations using *US3D* and *Pointwise*

C. Tang*

*Aerothermal Engineer, NASA Ames Research Center, Moffett Field, CA 94035

Abstract: For hypersonic simulations, unstructured flow solvers typically have problems predicting surface heat fluxes when strong shocks are present. To address these issues, this paper outlines a workflow that applies best practices developed for structured grids to unstructured meshes. In addition, unstructured grid generation can significantly reduce the time required to create quality grids for complex geometries. Several examples are computed using *DPLR*, a structured grid flow solver, and *US3D* flow, an unstructured mesh solver. Results from the two codes are compared, and they show excellent agreement. Overall, the unstructured grid workflow offers a viable and attractive alternative for hypersonic simulations.

Keywords: Unstructured Flow Solver, Anisotropic Hybrid Mesh, Shock-Alignment

1 Introduction

It is often desirable to run Computational Fluid Dynamics (CFD) simulations to predict the aerothermal environment of a spacecraft during atmospheric entry. Point-matched, structured grid flow solvers, such as *DPLR* [1] and *LAURA* [2], have proven to provide accurate heat flux estimates for many hypersonic applications [3-5]. However, for complex geometries, the grid generation/adaption process is often tedious and typically a bottleneck in the simulation workflow. Unstructured grid generation offers a potential alternative by simplifying the mesh generation/adaption process. While unstructured grid solvers have problems predicting surface heating when strong shocks are present [6], these numerical issues can be mitigated by simply applying best practices developed for structured grids to unstructured meshes. As demonstrated in [7], proper shock-alignment of unstructured grids can minimize numerical noise emanating from shocks; therefore, spurious oscillations in computed surface quantities (such as pressure, temperature, and heat flux) are greatly reduced.

To illustrate the workflows for structured vs. unstructured grids, several test cases are computed using two codes: *DPLR* (a structured grid, Navier-Stokes flow solver for reacting flows), and *US3D* [8] (an unstructured grid, Navier-Stokes flow solver for reacting flows). For grid generation, the commercial software package *Pointwise* [9] is used to create all surface and volume meshes. While the proposed framework is designed specifically using *Pointwise* as the mesh generator, other unstructured flow solvers, such as *Loci-CHEM* [10] or *FUN3D* [11], can be used in place of *US3D*.

2 Workflow for Structured Grids

As illustrated in Figure 1, a workflow for structured grids starts with the generation of a volume grid around the spacecraft. For a simple shape, like an idealized smooth Orion capsule, a structured surface mesh can be easily constructed on the capsule's surface. A volume grid can be generated by extruding the surface mesh outward using a hyperbolic extrusion algorithm available in *Pointwise*. Shown in Figure 2 is the surface grid, volume mesh, and initial solution computed using the *DPLR* flow solver.

The next step in the workflow is to align the grid with the bow shock and to adjust the wall spacing so there is sufficient grid resolution to properly resolve the boundary layer in a viscous simulation. The wall spacing is adjusted by specifying either a constant wall spacing or a cell Reynolds number (Re_c):

$Re_c = \frac{(a+V)\Delta\eta}{v}$, where a is the sound speed, V is the local velocity magnitude, $\Delta\eta$ is the wall spacing, and v is the kinematic viscosity. For accurate aerothermal simulations, it is recommended that $Re_c < 5$ everywhere at the wall surface.

The tedious tasks of shock detection, shock smoothing, grid alignment/wall clustering, and updating the volume mesh can all be handled automatically using the built-in grid adaption tool within *DPLR*. It usually takes approximately 3-5 CPU minutes to generate a new grid using the grid adaption subroutine. Illustrated in Figure 3 are plots of centerline Mach contours for the initial grid and the final grid after 3 adaptions. While the grid adaption algorithm in *DPLR* makes the workflow very streamlined, it only works for a specific grid topology (a **single** layer in the body-normal direction [k] that spans from the wall surface to the outer boundary). For complex geometry, this single-layer requirement is very constraining and makes grid generation quite tedious and challenging. It is certainly possible to create structured grids using other grid topologies. However, the steps for shock detection, smoothing, grid alignment/wall reclustering, and volume generation will need to be done outside of the flow solver. Since the volume grid needs to match an outer boundary (bow shock surface), elliptic grid generation, which can be quite time-consuming, is likely required.

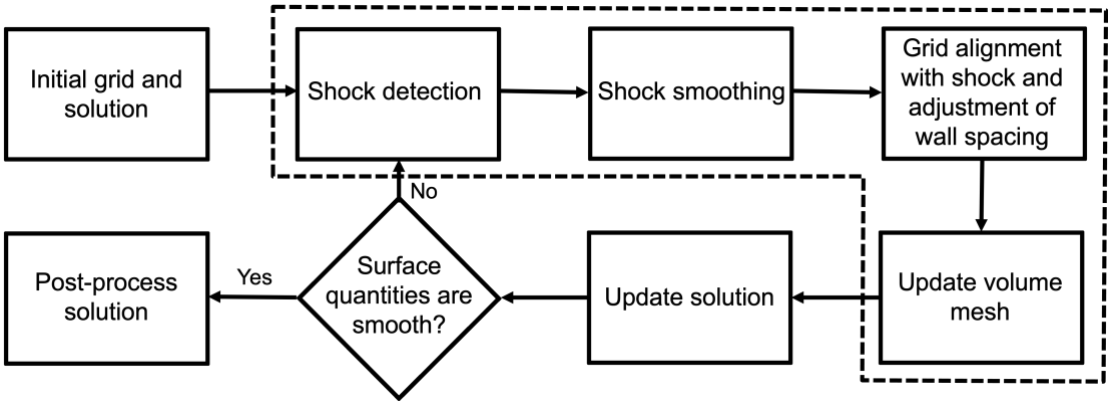

Figure 1: Workflow for structured grids.

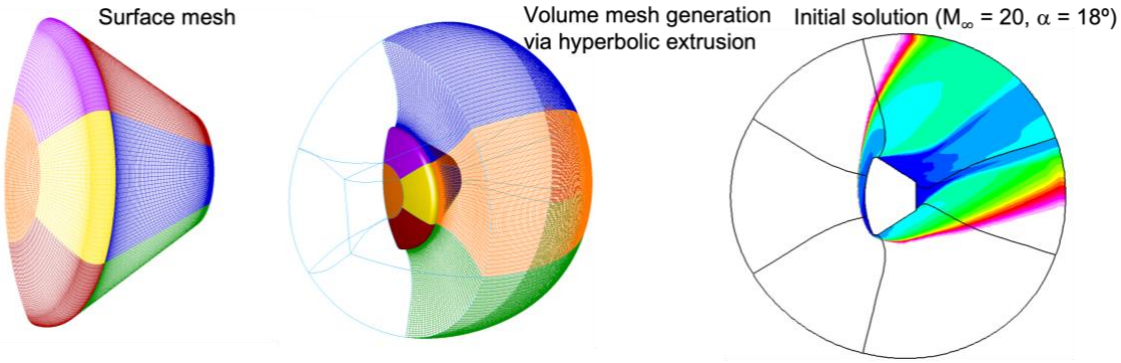

Figure 2: Surface grid, volume mesh, and centerline Mach contours on initial grid.

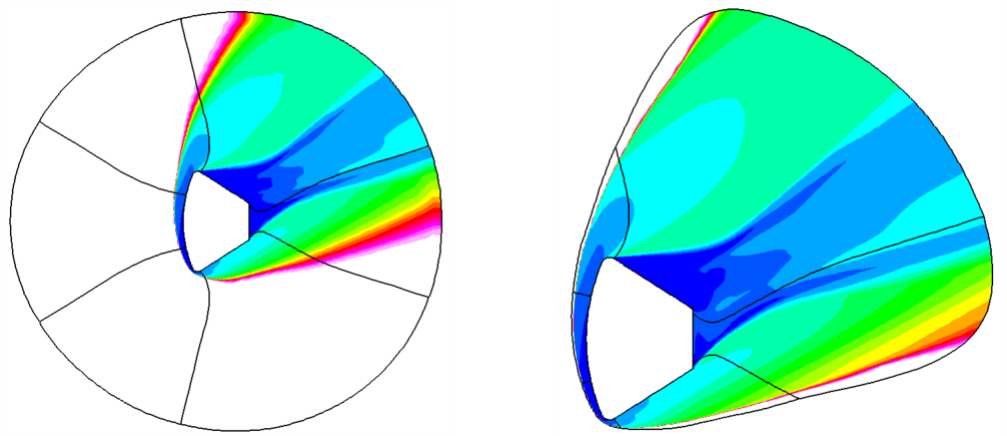

Figure 3: Centerline Mach contours on initial grid (left) and grid after 3 adaptions (right).

3 Workflow for Unstructured Grids

3.1 Initial grid

As an alternative to structured grids, a workflow using unstructured meshes can simplify the grid generation/adaption process and produce faster turnaround times for hypersonic simulations. Shown in Figure 4 is the workflow for unstructured grids. To start the simulation, an initial grid needs to be generated. One option in *Pointwise* is to utilize the T-Rex algorithm [12], a hybrid meshing method using 3D anisotropic tetrahedral extrusion. This method is similar to a hyperbolic extrusion algorithm where surface points are extruded outward to create a volume mesh. However, one key difference is that each surface point is checked to see if the extrusion would result in a collision with other grid points. If this collision test fails, the extrusion process is stopped locally for that point. The marching front continues for the remaining points until all collision tests fail or a specified maximum number of layers (usually ~ 50-80 layers) is reached. When the marching front stops, the remaining volume is filled by a Delaunay-based isotropic mesh algorithm. The T-Rex algorithm is quite robust, and it can generate high-quality boundary layer grids around complex geometries.

The setup for running T-Rex is straightforward. A user creates surfaces defining the spacecraft and outer boundaries; specifies boundary types, maximum number of prismatic layers, wall spacing, and stretching ratio. Once these parameters are set, the algorithm can be initialized to run without any user interactions. Shown in Figure 5 are plots of a hybrid mesh generated for an Orion capsule. The right picture shows a close-up of the prismatic layer generated at the surface of the capsule and isotropic cells connecting the prismatic grid with the outer boundary (a hemisphere with a planar surface at the exit).

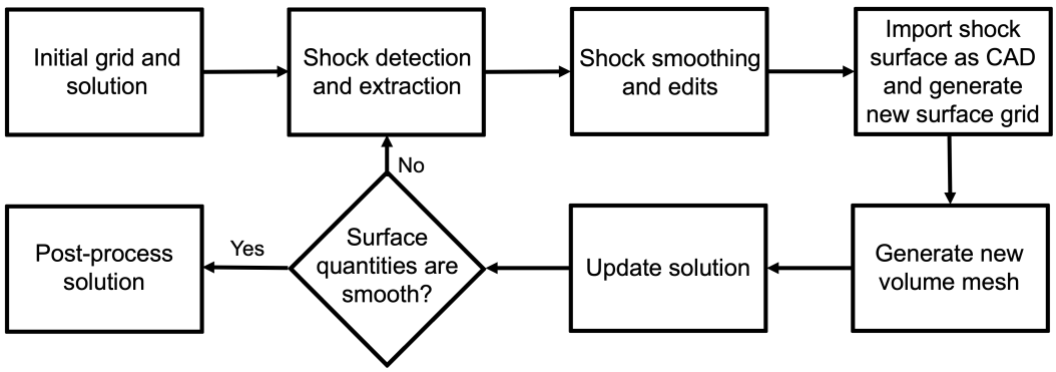

Figure 4: Workflow for unstructured grids.

3.2 Initial solution

After the unstructured grid is generated, *US3D* is used to run a Navier-Stokes simulation on the initial grid. The computational domain is initialized using freestream conditions and an artificial boundary layer at the surface (iconr = 0). A 2nd order implicit Euler-time integration and line relaxation method (impl = 21) is selected to solve the governing equations. For a hybrid grid, line relaxation is active in the prismatic layers at the wall surface. Outside of this region, the solver switches to a point implicit method. For numerical stability, it is often beneficial to run *US3D* with first order inviscid fluxes in space (iorder = 1) and a large artificial dissipation factor (epsj = 1.0–5.0). Once the simulation has converged sufficiently (formation of a bow shock and subsonic wake), the computation can be restarted using second order fluxes and a lower dissipation value.

3.3 Shock detection and extraction

The shock detection/extraction process starts by post-processing the *US3D* solution to output Mach number at each point in the volume grid. The location of the bow shock is selected using a percentage of the freestream Mach number (typically 90%-95% of M_∞). Using the *Tecplot* [13] visualization package, a Mach iso-surface is extracted from the flow solution and the surface is written out as a *Tecplot* data file. Since the data format is not compatible with *Pointwise*, the file is converted to a UCD format using the *TEC_TO_UCD* [14] code. Shown in Figure 6 is a plot of the iso-surface extracted from *Tecplot*. The Mach iso-surface is quite rough because the bow shock is coarsely resolved on the initial grid.

3.4 Shock smoothing and edits

The iso-surface in UCD format can be imported as a database file in *Pointwise* and exported out as a STL file. This format conversion is desirable because STL files can be imported into the *MeshLab* [15] software package for its numerous smoothing options. Shown in the right image of Figure 6 is the Mach iso-surface smoothed using a combination of HC Laplacian smoothing and Laplacian smoothing. For the initial iso-surface, many smoothing steps are needed (~ 30-50 steps using the Laplacian smooth filter) to obtain a relatively smooth surface. For subsequent adaptions, the extracted iso-surface is fairly smooth so only a few smoothing steps (~ 3-5) is needed. The smoothed iso-surface is imported into *Pointwise* as a database. If desired, the database can be trimmed to reduce the size of the outer boundary or remove any rough edges.

3.5 Generate new surface and volume grids

Although the new iso-surface is smooth, it may contain triangles with very small areas, which could result in the generation of poor-quality volume cells. This problem can be resolved by creating a new unstructured surface via the "on database entities" option in *Pointwise*. By selecting the "isotropic" option for unstructured domains, the new surface should be quite uniform in cell size. If desired, the user can change the cell size distribution by adding source terms. Shown in Figure 7 is a plot of a new grid representing the bow shock surface.

To generate a new volume grid, the new surface is used as the outer boundary and the boundary type is set as a "wall" with a spacing of 1 mm (recommended value for shocks). This setting enables T-Rex to generate a prismatic grid that marches from the bow shock toward the spacecraft's surface. The exit of the outer boundary is closed off by creating an unstructured surface mesh (using a 2D version of T-Rex with a matching wall spacing of 1 mm) and setting the boundary type as "Match". At the spacecraft's surface, the wall spacing should be set to satisfy a cell Reynolds criterion ($Re_c < 5$). A wall spacing of 1-10 microns and a stretching ratio of 1.15-1.20 are typically values used to setup the T-Rex algorithm. Shown in the middle picture of Figure 7 is a close-up view of the new volume grid. Since the "wall" spacing at the outer boundary is much larger than the spacing at the vehicle's surface, the shock front could grow quite far inward. To prevent the shock front from growing too close to the spacecraft's

surface (and limiting the growth of the boundary layer grid), a collision buffer setting of 2.0-4.0 is specified. Once all parameters are set, the user can initiate the T-Rex algorithm to generate a hybrid volume grid.

It should be noted that the bow shock location was selected as 90%-95% of the freestream Mach number. Therefore, it is possible that the outer boundary may be located inside the bow shock. To ensure we can correctly impose freestream conditions at the outer boundary, the outer surface is extruded outward (~ 10-30 layers with an initial spacing of 1 mm and stretching ratio of 1.15-1.20) using the extrusion algorithm in *Pointwise*.

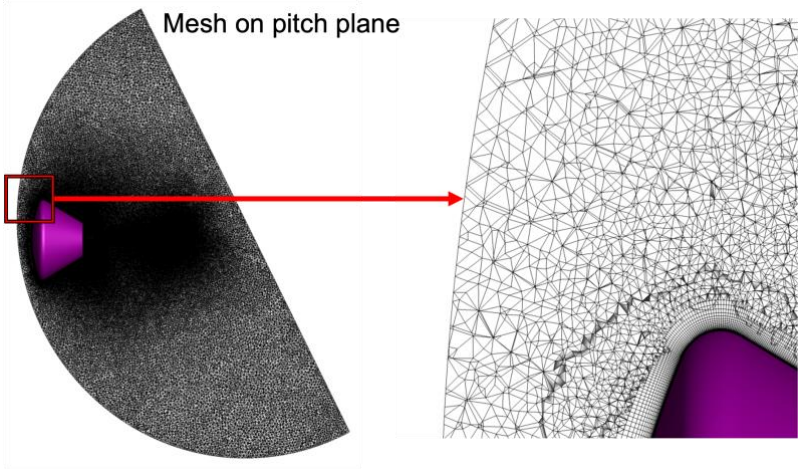

Figure 5: Unstructured grid generation using T-Rex

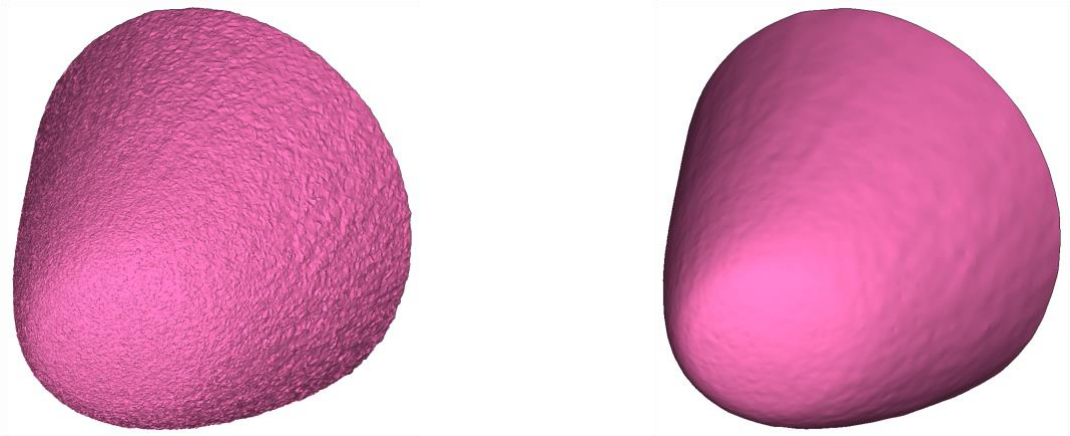

Figure 6: Mach iso-surface from Tecplot (left) and smoothed iso-surface using *MeshLab* (right).

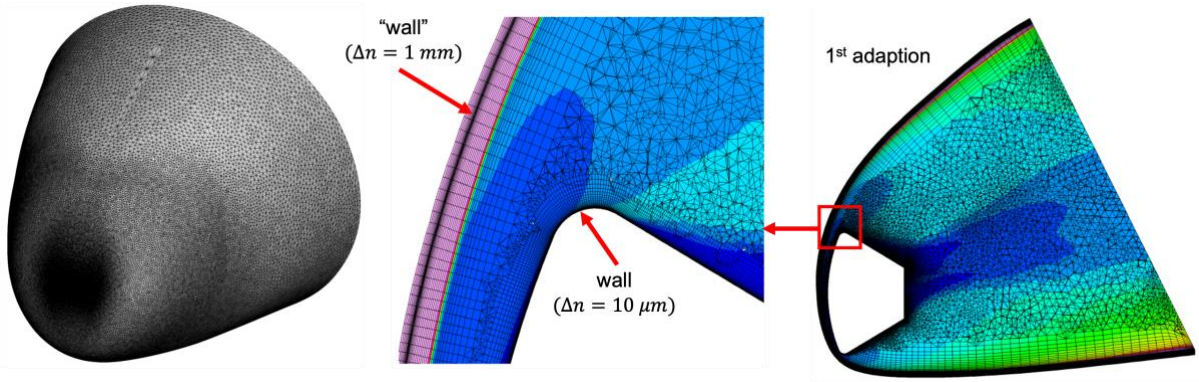

Figure 7: New surface mesh (left), volume mesh (right), and close-up view of volume mesh (middle).

3.6 Update solution

For faster convergence, a user should use the *us3d-interp* utility to interpolate the previous solution onto the new volume grid and use the interpolated solution to restart a *US3D* simulation. Once again, for numerical stability, it may be helpful to reduce the solver to first order accuracy and use a large artificial dissipation factor. After the simulation has converged, the computation can be restarted using second order fluxes and a lower dissipation value. The numerical dissipation near the wall surface can also be reduced by setting the kbl parameter to the average number of prismatic layers defining the boundary layer (~ 40-60).

4 Numerical Results

To illustrate the differences in workflow between structured vs. unstructured grids, two examples are presented: (1) a simple geometry using an idealized, smooth Orion capsule and (2) a complex geometry using the Space Shuttle. *DPLR v4.02.2* is used on structured grids while *US3D v1.1.7* is selected for unstructured grids. The following settings were selected to make both codes as similar as possible:

(1) Laminar Navier-Stokes (easier to compare results without the complication of turbulence modeling)
(2) A five-species (N2, O2, NO, N, O) five-reaction finite-rate air chemistry model [16]
(3) 2nd order modified Steger-Warming fluxes [17]
(4) Thermal nonequilibrium with a two-temperature model [18]
(5) Gupta collision integral-based transport properties [19]
(6) Self-Consistent Effective Binary Diffusion (SCEBD) [20]
(7) NASA Lewis database for species properties
(8) Fully-catalytic wall boundary condition with radiative equilibrium ($\varepsilon = 0.85$)

4.1 Orion Capsule

To compare *DPLR* and *US3D* results on a simple geometry, laminar Navier-Stokes simulations are computed on a smooth Orion capsule with freestream conditions of $M_\infty = 20$ and $\alpha = 18^o$. For the structured grid, a single-layer grid containing 16 million points was generated using a hyperbolic extrusion algorithm in *Pointwise*. The built-in grid adaption subroutine in *DPLR* was used to adapt the volume mesh 4 times. For the unstructured grid, the initial grid was constructed using the surface mesh from the structured grid and a rotated hemisphere to define the outer boundary (see Figure 5). The initial grid contained 10.5 million grid points, and the grid generation with T-Rex took approximately 10 CPU minutes on a MacBook Pro laptop (2.6 GHz 6-Core Intel i7). The unstructured grid was adapted twice, and the final mesh contained 15 million grid points. It should be noted that for unstructured grids, a large numerical dissipation value (~ 1.0-2.0) may be needed to damp out oscillations at the stagnation point. Shown in Figure 8 are Mach contours on the initial *US3D* grid, final *US3D* grid after 2 adaptions, and final *DPLR* grid after 4 adaptions. A comparison of the surface heat flux (normalized by the peak heating rate from the *DPLR* solution) is shown in Figure 9. The right image shows the heat flux contours on the Orion heatshield. It is clearly evident that the initial unstructured grid (without shock alignment) produced significant oscillations in the surface heating. After two adaption cycles, the *US3D* solution agrees quite well with the final *DPLR* solution. A plot of the normalized heat flux on the capsule's centerline is shown in the left image. A closer examination shows *US3D* predicted a peak heat flux that is 4% higher than the *DPLR* estimate. For simple geometries, the unstructured workflow does not offer a speed advantage over a structured workflow since the built-in grid adaption routine in *DPLR* is very fast and well automated. Overall, the results from both codes are in excellent agreement when the workflows and best practices (as detailed in the previous section) are followed.

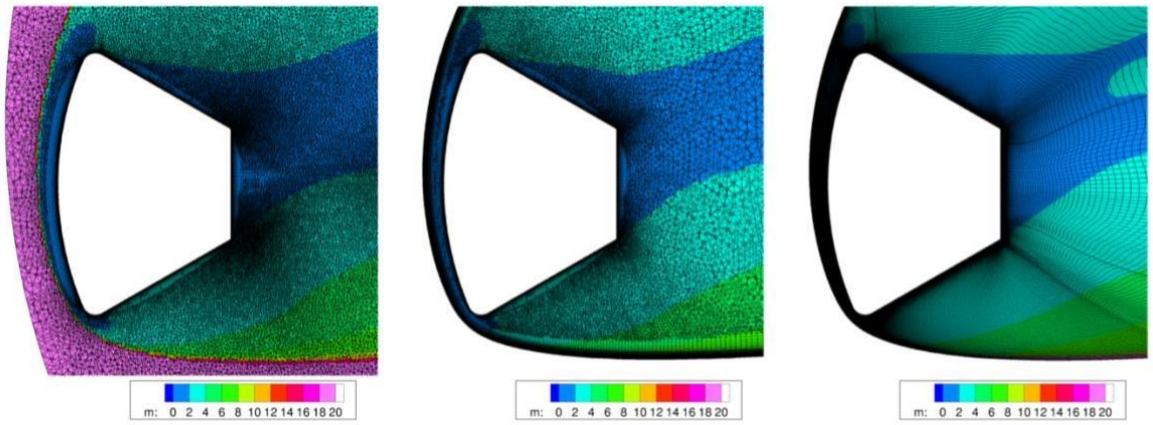

Figure 8: Mach contours on initial *US3D* grid (left), final adapted *US3D* grid (middle), and final adapted *DPLR* grid (right).

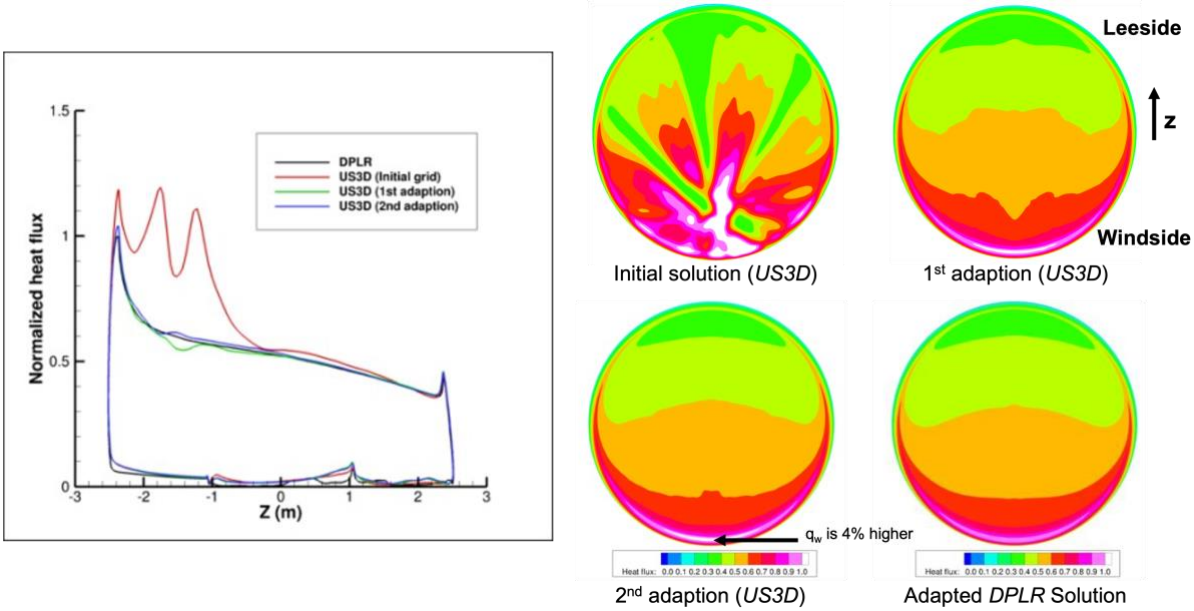

Figure 9: Comparisons of normalized heat flux between *US3D* and *DPLR*.

4.2 Space Shuttle

While it is straightforward to generate a single-layer, structured grid for a smooth capsule, grid generation is often quite challenging for a more complex configuration. As an example, the Space Shuttle CAD geometry (shown in Figure 10) contains numerous gaps and concave surfaces. As detailed in [21], certain geometric features were simplified (no gaps in the elevons) or ignored (body flap and aft end of the vehicle) in order to make the grid generation possible. Even with these simplifications, it took skilled users several weeks to create a structured volume grid. Shown in the right image of Figure 10 is the structured grid after 3 grid adaptions with the *DPLR* flow solver ($M_\infty = 17.9$, $\alpha = 40^o$).

To test the unstructured workflow, the structured surface grids of the Shuttle were used to create a hybrid volume mesh. One modification was to form a watertight surface by closing off the trailing edges of the wing and vertical tail, and the aft end of the vehicle (see Figure 11). This change allows us to model the entire Orbiter and to impose a supersonic outflow boundary condition at the exit. Once again, a hemisphere with a planar surface exit defines the outer boundaries of the initial mesh. One of the best features of the T-Rex algorithm is that the grid generation process remains the same regardless of the

geometry complexity. Shown in Figure 11 is the centerline *US3D* grid after 2 adaptions. The final mesh contained 18 million grid points, and it took around 18 CPU minutes to generate the hybrid mesh using T-Rex.

A comparison of the heat flux (normalized by the stagnation heating from the *DPLR* solution) shows excellent agreement between the two codes. Shown in Figure 12 is the normalized heat flux contours on the windside (left) and leeside (middle), and a plot of the centerline heat flux down the length of the vehicle (right). The peak centerline heat flux predicted by *US3D* is about 2% lower than the *DPLR* estimate. As indicated by these results, an unstructured flow solver can produce heating estimates comparable to a structured flow solver if best practices are applied.

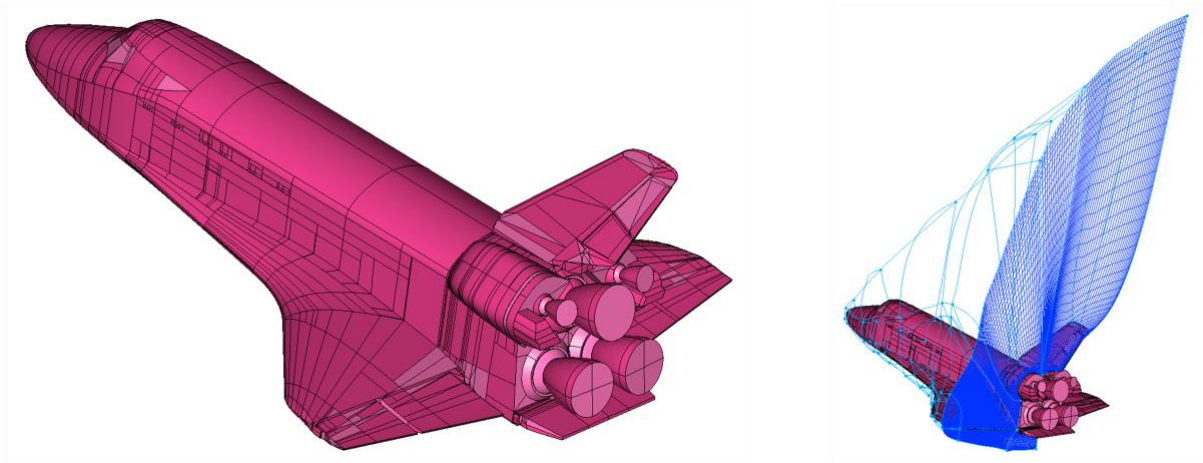

Figure 10: Space Shuttle CAD geometry (left) and adapted structured grid (right).

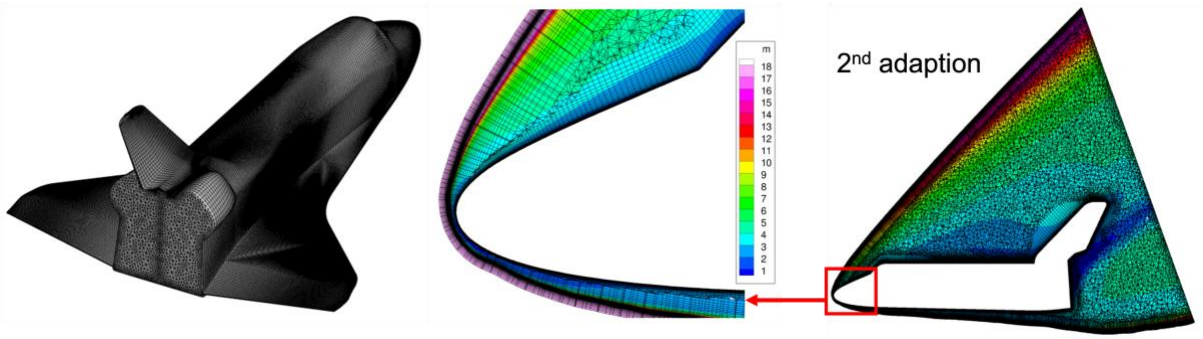

Figure 11: Watertight surface grid (left), final adapted *US3D* grid (right), and close-up view of adapted grid (middle).

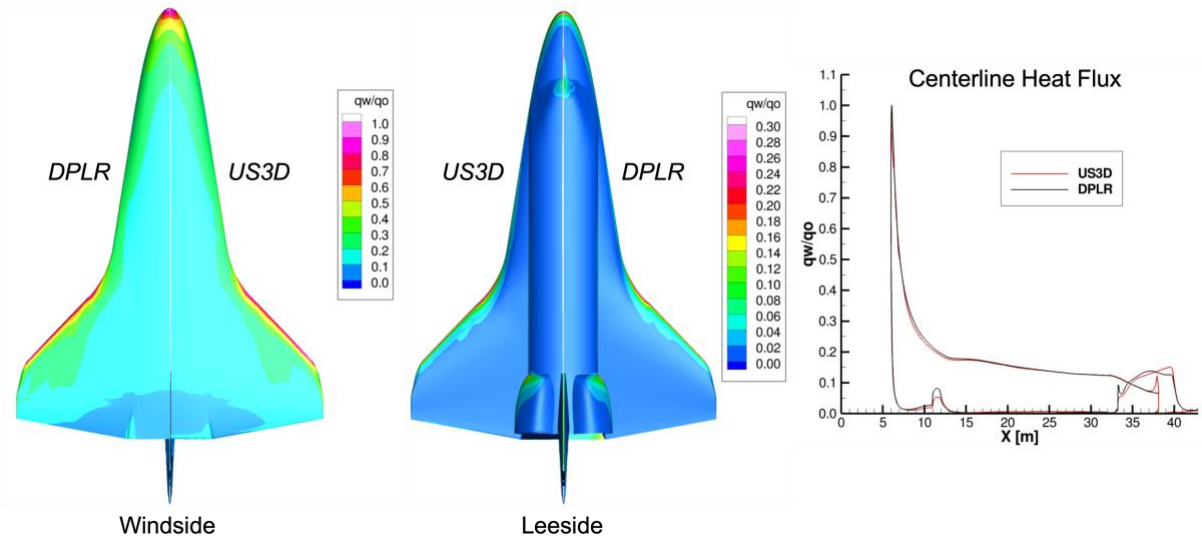

Figure 12: Normalized heat flux contours on windside (left), leeside (middle), and centerline (right).

5 Advantages of Unstructured Grid Generation in *Pointwise*

Pointwise also offers many useful features when generating unstructured grids. In particular, the automatic surface mesh generation tool can quickly create high-quality surface grids from a database. To illustrate this feature, a truncated CAD of the Shuttle's body flap and aft end is imported and assembled as a model in *Pointwise* (see left image in Figure 13). By clicking on the automatic surface mesh button, unstructured grids are generated on the database in approximately 90 CPU seconds. Shown in the right image is the surface meshes produced using the auto-mesh feature. It should be noted that the trimmed surfaces in the CAD file were not combined to form larger "quilts" during the model assembly process. As a result, small mesh cells are produced from small trimmed surfaces. This issue can be mitigated by combining similar trimmed surfaces (based on an angle tolerance defined by normal vectors of two adjacent surfaces) to form larger quilts. This process reduces the number of database entities and minimizes the generation of small surfaces. In addition, surface grids can be manually adjusted to eliminate small features (e.g., protuberances and gaps) that are not important in a simulation.

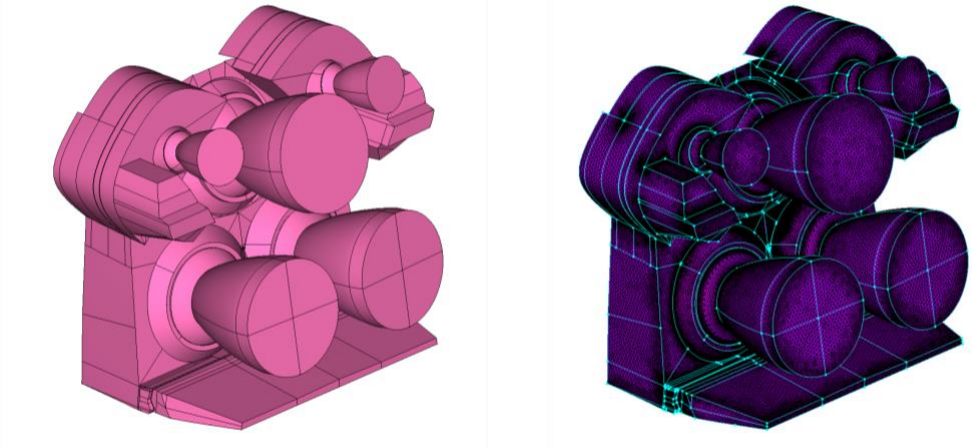

Figure 13: Truncated CAD geometry (left) and automatic mesh generation (right).

To test out the robustness of the T-Rex algorithm, unstructured grids defining the body flap and aft end were combined with the original structured Shuttle grids to form a new surface mesh (see Figure 14). The T-Rex algorithm took approximately 21 CPU minutes to generate a hybrid mesh (containing ~17.9 million grid points). Shown in the right image of Figure 14 is temperature contours (normalized by the

stagnation temperature) on the Shuttle's surface. Clearly, the T-Rex algorithm is quite robust in handling complex geometries, and it can quickly generate a hybrid mesh.

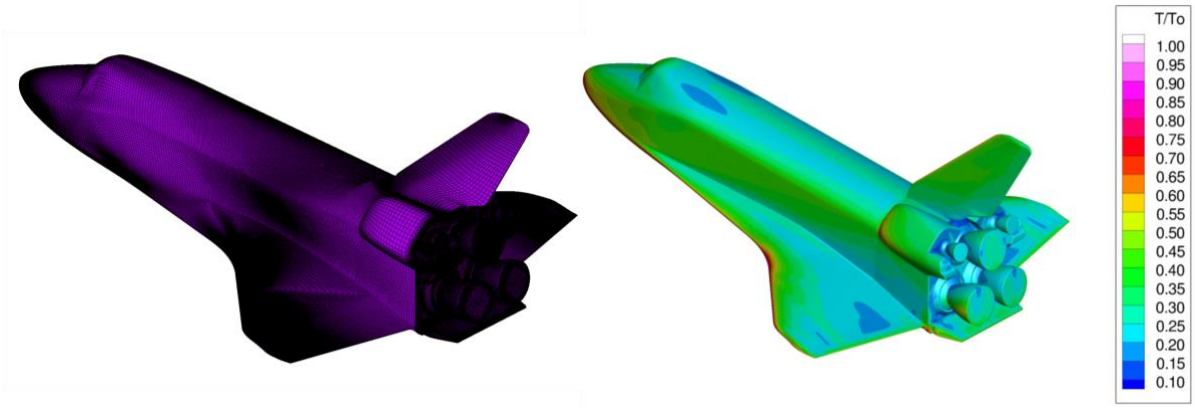

Figure 14: New Shuttle surface (left) and temperature contours (right)

Another useful feature in *Pointwise* is the ability to add sources to refine grids in a specific region of the computation domain. Shown in Figure 15 is an example of a box source (created in *Pointwise*) added to T-Rex for refining the wake region of the Shuttle simulation. Since the "adaption" option for the exit boundary was turned on, the exit surface was also refined by the box source. Similarly, surfaces on the vehicle can be selected to adapt based on nearby sources. For wall surfaces defined by a CAD database, this option assures that refined surface grids are automatically projected onto the database. Thus, incorporating automatic mesh refinement to the current workflow is quite feasible.

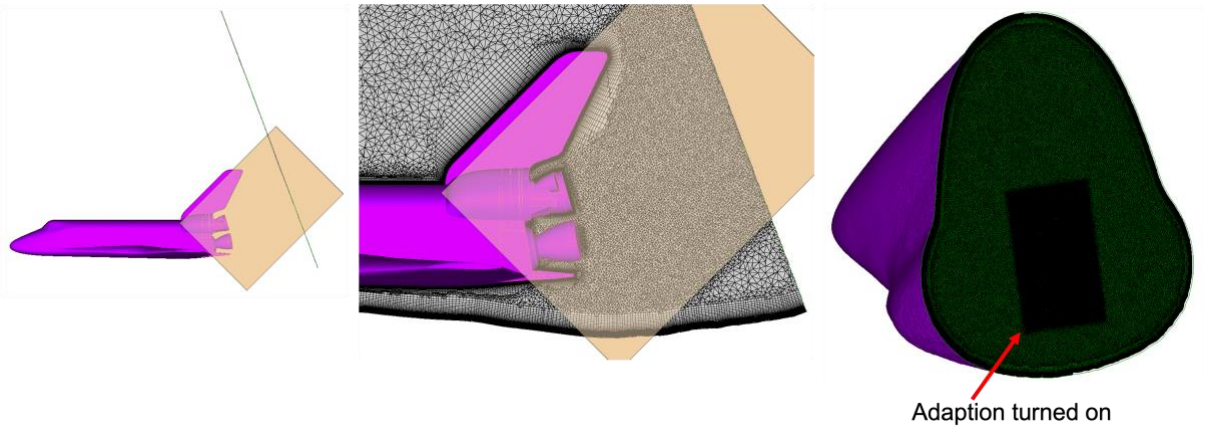

Figure 15: Box source (left), refined volume mesh (middle), and adapted exit surface (right)

6 Concluding Remarks

A workflow using unstructured grid/solver has been developed and tested. As demonstrated by the Space Shuttle example, the new process can quickly produce accurate heating estimates for a complex geometry, and it is a viable and promising alternative to the traditional structured grid methods. In particular, the T-Rex meshing algorithm is shown to be fast, robust, and easy to implement. When combined with the automatic surface mesh option in *Pointwise*, the unstructured grid generation process is extremely fast and efficient. While an unstructured workflow does not offer a speed advantage over a structured workflow for simple geometries, the difference is significant for complex geometries. For example, it would likely take a skilled user many weeks to generate a point-matched, structured grid for the full Shuttle geometry (as illustrated in Figure 10). A corresponding unstructured mesh would probably take about one day. Unstructured mesh generation also offers great flexibility in modification of existing surface and volume grids because grid points can be easily added or deleted without the need

to adhere to a particular grid topology. Overall, the current unstructured workflow offers a fast and straightforward process to model hypersonic flows for complex geometries.

References

[1] 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.

[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.

[3] Wright, M. J., Prabhu, D. K., and Martinez, E. R., "Analysis of Apollo Command Module Afterbody Heating Part I: AS-202," *Journal of Thermophysics and Heat Transfer*, Vol. 20, No. 1, 2006, pp. 16-30.

[4] Tang, C., "Simulations of Afterbody Heating Rates on the Apollo Command Module for AS-202 Using Hyperbolic Grids," 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, CA, June 2006.

[5] Wright, M., Loomis, M., and Papadopoulos, P., "Aerothermal Analysis of the Project Fire II Afterbody Flow," *Journal of Thermophysics and Heat Transfer*, Vol. 17, No. 2, 2003, pp. 240-249.

[6] Candler. G., Barnhardt, M., Drayna, T., Nompelis, I., Peterson, D., and Subbareddy, P., "Unstructured Grid Approaches for Accurate Aeroheating Simulations," 18th AIAA Computational Fluid Dynamics Conference, AIAA 2007-3959, Miami, FL, June 2007.

[7] McCloud. P., "Best Practices for Unstructured Grid Shock-Fitting," 55th AIAA Aerospace Sciences Meeting, Grapevine, TX, January 2017.

[8] Candler, G., Johnson, H., Nompelis, I., Subbareddy, P., Drayna, T., Gidzak, V., and Barnhardt, M., "Development of the US3D Code for Advanced Compressible and Reacting Flow Simulations," 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2015.

[9] Pointwise, http://www.pointwise.com/pw

[10] Luke, E., On Robust and Accurate Arbitrary Polytope CFD Solvers (Invited), 18th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, AIAA-2007-3956, 2007.

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

[12] T-Rex meshing, http://www.pointwise.com/articles/t-rex-hybrid-meshing-in-pointwise

[13] Tecplot, https://www.tecplot.com/

[14] TEC_TO_TEC, https://people.math.sc.edu/Burkardt/f_src/tec_to_ucd/tec_to_ucd.html

[15] MeshLab, https://www.meshlab.net/

[16] Park, C., "Review of Chemical-Kinetic Problems of Future NASA Missions, I: Earth Entries," *Journal of Thermophysics and Heat Transfer*, Vol. 7, No. 3, 1993, pp. 385-398.

[17] MacCormack, R., and Candler, G., "The Solution of the Navier–Stokes Equations Using Gauss-Seidel Line Relaxation," *Computers and Fluids*, Vol. 17, No. 1, 1989, pp. 135–150.

[18] Park, C., Nonequilibrium Hypersonic Aerothermodynamics, Wiley, New York, 1990, pp. 344–348.

[19] Gupta, R., Yos, J., Thompson, R., and Lee, K., "A Review of Reaction Rates and Thermodynamic and Transport Properties for an 11-species Air Model for Chemical and Thermal nonequilibrium Calculations to 30,000 K," NASA RP-1232, Aug. 1990.

[20] Ramshaw, J. D., "Self-Consistent Effective Binary Diffusion in Multi-component Gas Mixtures," *Journal of Non-Equilibrium Thermodynamics*, Vol. 15, No. 3, 1990, pp. 295–300.

[21] Alter, S. J., McDaniel, R. D., and Reuther J. J., "Development of a Flexible Framework of Common Hypersonic Navier-Stokes Meshes for the Space Shuttle Orbiter," JANNAF 27th Airbreathing Propulsion Subcommittee Meeting, Colorado Springs, CO, Dec. 1-5, 2003.