AIAA AVIATION Forum
August 2-6, 2021, VIRTUAL EVENT
AIAA AVIATION 2021 FORUM

10.2514/6.2021-2701

Towards an Automated Unstructured Grid Adaptation Workflow with VULCAN

Matthew O'Connell*, Jeffery White†, Kyle B. Thompson‡,
NASA Langley Research Center,
Hampton, VA 23681, USA

Early work is presented for an unstructured grid adaptation workflow with VULCAN and *refine*. Anisotropic simplex grids are iteratively adapted to match a Riemannian metric tensor field describing desired mesh spacing. The Riemannian metric tensor field is obtained from Hessians of CFD solution output scalar sensor fields; both Mach number and static temperature sensor fields are explored. In addition, we describe a Newton-method-based solver recently implemented in VULCAN utilizing Jacobian-Free-Newton-Krylov that can be used to increase flow solver automation on early grids in the adadptation process. Hypersonic flow solutions are presented on a high Reynolds number flat plate and wall heat flux is compared against a highly resolved structured solution. Additionally, complex shock boundary-layer interaction is explored in a high Mach number compression corner and complex 3D flow phenomena are evaluated on the Boundary Layer Transition (BOLT) vehicle.

I. Nomenclature

V	=	cell volume
Q	=	solution vector of partial differential equations
R	=	nonlinear residual of discretized partial differential equations
ε	=	finite-difference step size
^{n}	=	iteration level
λ	=	line search step size

II. Introduction

Computational Fluid Dynamics (CFD) codes are heavily used for both the analysis and design of supersonic and hypersonic vehicles. CFD techniques utilizing structured grids have long been preferred over unstructured techniques for high speed applications due to their higher accuracy, particularly in computing wall heat flux or skin friction. Calculations with fixed unstructured meshes [1] and off-body adaptation based on solution gradients [2] show unsmooth surface heating profiles. However, generating structured grids can be time consuming, particularly in regions where the geometry is complex, such as the flow path of a scramjet. And though structured-grid adaptation has been successfully used with local and mostly one-dimensional grid morphing schemes, it can also be challenging to generate structured grids to adequately match local flow characteristics. In cases with highly complex geometry or flow physics, a tetrahedral-based unstructured discretization technique combined with robust grid adaptation is a desirable tool — so long as accuracy is sufficient. Recent work with VULCAN, a finite-volume CFD solver focused on hypersonic applications, has been focused on improving accuracy and robustness of the unstructured discretization to enable unstructured adaptation [3–5]. This improved tetrahedral unstructured discretization is combined with dramatic progress made in the last decade for anisotropic solution-adaptive methods to resolve simulations with shocks and boundary layers [6], where a community effort has verified these adaptation methods for subsonic and transonic flows. [7–9]. In particular, smooth skin friction is shown in Ref. [9] which is relevant to other boundary normal derivatives like heating.

It has previously been demonstrated in Ref. [10] that deep nonlinear convergence may also be required to achieve accurate wall heat flux in hypersonic problems. The defect-correction scheme in VULCAN can often reach deep levels

*Research Scientist, Computational Aerosciences Branch, AIAA Member.
†Senior Research Scientist, Computational Aerosciences Branch, AIAA Associate Fellow.
‡Research Scientist, Aerothermodynamics Branch, AIAA Member.

of convergence on bespoke unstructured grids. However, when utilizing an automated adaptation-based workflow, the VULCAN solver may be required to solve a sequence of tens of automatically generated grids to resolve the highly nonlinear flow solution. These adapted grids have tight anisotropic spacing to achieve higher accuracy, but fine spacing lowers dissipation and the anisotropy can increase stiffness to create a challenging nonlinear problem. Therefore, a robust and automated nonlinear solver strategy, such as Pseudotransient Continuation (PTC) based on Newton's method, is an attractive option.

The goal of this paper is twofold. First is to demonstrate recent development efforts in adding a nonlinear solution procedure to VULCAN, based on Newton's-method, that increases robustness and automation. The increased robustness and automation enables a semiautomatic adaptation workflow with VULCAN and *refine* [11] to obtain solutions on unstructured simplex grids. The second goal is to explore the use of Mach number or static temperature sensor fields to drive mesh adaptation for hypersonic problems. The multiscale metric controls estimated interpolation error in a field. [12–14]. We demonstrate two sensor fields to see how they are related to heating prediction. The choice of sensor field impacts the accuracy of the flow solutions, which is demonstrated by a comparison of wall-heating rates and grid spacing.

III. Unstructured Mesh-Adaptation Methodology

The approach taken to utilize anisotropic mesh adaptation with VULCAN follows the overall strategy described in Ref. [15]. The problem domain geometry is defined with the Engineering Sketch Pad (ESP) [16] that includes OpenCSM [17] as a constructive solid modeler. Attributes [17], composed of name and value pairs, are carried throughout the geometry build process to the various faces of geometry and used to specify the VULCAN boundary conditions. Once the geometry of the problem domain is defined in OpenCSM, an initial surface triangulation is created by the Electronic Geometry Aircraft Design System (EGADS) [18], where parallel execution is facilitated by EGADSlite [19]. At this point, the problem domain is bounded by a water-tight surface mesh where each surface triangle is associated with underlying geometric entities. Maintaining this association between the discrete mesh and the underlying geometric representation is critical because during mesh adaptation the association is used by *refine* [11] to ensure the adapted meshes represent the true geometric intent as the grid is adapted. The initial EGADS surface mesh is adapted based on a geometric curvature and feature size [20] with an interpolated geometry constraint metric.

Once the surface triangulation is complete, the volumetric fluid domain is then filled with tetrahedra using TetGen [21]. This initial Delaunay-based TetGen volume mesh does not satisfy the surface geometry constraint metric, so *refine* adaptation is used to satisfy the geometry constraint metric on the surface and volume simultaneously. The mesh is now as coarse as possible while satisfying the geometry constraint metric. An optional step can be performed to refine the initial mesh based on an implied metric to a user-specified complexity (number of elements). At this point, the mesh is ready to obtain an initial solution using VULCAN.

With the generation of the initial grid complete, the adaptation loop begins. The OpenCSM boundary attributes are mapped onto the discrete mesh as well as to numerical boundary conditions implemented in VULCAN. The JFNK solver within VULCAN is provided by Enigma [10], which is used to obtain a solution on the discrete mesh. A scalar sensor field is extracted from the VULCAN solution (static temperature or Mach number). A Hessian of the sensor scalar field is calculated and is used to construct a Riemannian metric tensor field. The Riemannian metric field specifies desired anisotropic spacing to control the L_2 norm of the estimated interpolation error of the scalar field [12–14]. This is a key development over an earlier, Hessian-based element that controls estimated interpolation error in L_∞ norm because using the Riemannian metric field provides control over smooth and unsmooth contributions to the error simultaneously. The Riemannian metric field is scaled globally to specify a new mesh of a targeted size (mathematically defined as the complexity of the metric [13]). The mesh adaptation package *refine* is used to generate a new mesh that closely matches the spacing requested in the scaled Riemannian tensor field. By controlling for interpolation error, the output Riemannian tensor field will often describe a spacing field calling for highly anisotropic elements. This is particularly true inside boundary layers and at shocks. The newly-generated mesh can be adapted both in the interior of the mesh, and along the surfaces with the surface meshes being adapted onto an underlying EGADS geometry representation and thereby producing a mesh that conforms to the anisotropic nature of the tensor field.

Once a newly-adapted mesh is complete, the cell-centered solution state of VULCAN is averaged to the nodes of the original mesh, and the nodal solution is linearly interpolated from the original mesh onto the nodes of the adapted mesh. The interpolated solution field defined at nodes of the new mesh is averaged to obtain an initial solution at the cell centers of the new mesh. This interpolated solution is used as the starting state for VULCAN when solving on the new grid. Figure 1 shows the workflow that continues until accuracy requirements or resource limits (e.g., execution time or mesh size) are reached.

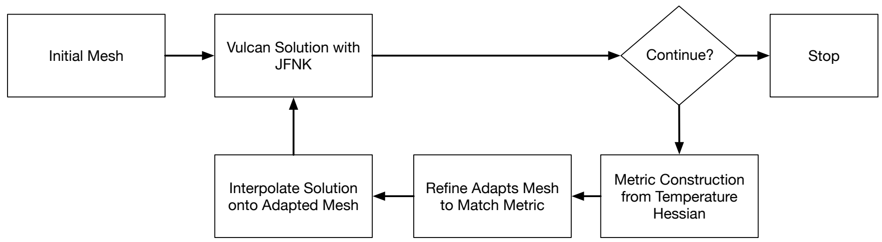


Fig. 1 Adaptation workflow: An initial mesh is iteratively adapted to control interpolation error in the temperature field.

For the results in this paper, the grid sizing in terms of complexity follows a fixed schedule. Each case begins with a defined initial target complexity. Complexity as an integral measure of the metric and a sharp estimate of the cells in the adapted mesh, see [**?**]. The complexity is doubled from the first to the second adaptation cycle and every fifth adaptation cycle. Each series of five grids are adapted to the same target complexity, although the number of elements may vary between grids in the same series by a few percent. This paper will report mesh sizes in terms of the number of control volumes (triangles in 2D, or tetrahedra in 3D) as opposed to the number of nodes, or the grid complexity, even though it is the grid complexity that is specified directly and not the number of control volumes.

IV. Jacobian-Free Newton-Krylov in VULCAN

A strong nonlinear solver is highly desirable when using the anisotropic mesh-adaptation workflow described in section III. The following section describes the addition of a Jacobian-Free Newton-Krylov (JFNK) based nonlinear solving technique to VULCAN. Deep nonlinear convergence for each grid is a long-term goal, and JFNK provides a tool toward achieving that goal. For the work presented in this paper, the JFNK technique is primarily used during coarse and medium-sized adapted grids where Newton convergence can be obtained. However, the two main reasons for using JFNK over the typical defect-correction-based nonlinear solver used in VULCAN are automation and robustness. It has been the author's experience that the simplex anisotropic meshes are more challenging to achieve deep nonlinear convergence than with traditional mixed-element meshes obtained via commercial grid generators. The nondivergence behavior of Krylov schemes combined with an automated CFL controller significantly reduce the human-in-the-loop effort for adapted grids. When using the Sketch-to-Solution workflow each case requires solving tens of adapted grids per case as whereas an engineer may only use three (coarse, medium, fine) or fewer grids in a fixed-grid workflow. Automation is key to successfully leveraging adapted grids.

The strong nonlinear solver used within VULCAN for this work employs the Enigma plugin framework [10] utilizing a PETSc [22] backend. A C++ wrapper to VULCAN was developed to implement the `Enigma::EquationSet` abstraction. Enigma was utilized to solve the minimization process:

$$Q \ni R(Q) = 0 \tag{1}$$

R is the nonlinear residual of the discretized PDEs within VULCAN and Q is conserved quantities of mass, momentum, and energy. The solution is updated iteratively as

$$\frac{\partial R}{\partial Q}\Delta Q = -R(Q^n) \tag{2}$$

$$\Delta Q = Q^{n+1} - Q^n \tag{3}$$

Where the Jacobian of the PDE, $\frac{\partial R}{\partial Q}$, is updated via the same solution as that used to evaluate R. Because Eq. 2 requires the solution of a (potentially stiff) linear system at each update, a preconditioned Krylov subspace method is used at each nonlinear step. Krylov subspace methods only require that the matrix-vector product $\frac{\partial R}{\partial Q}\Delta Q$ be formed for each search direction in the Krylov subspace during the linear solve, which can be approximated via a Frechet derivative [**?**]

$$\frac{\partial R}{\partial Q}\Delta Q = \frac{R(Q + \varepsilon\Delta Q) - R(Q)}{\varepsilon} \tag{4}$$

This approximation is referred to as the Jacobian-Free Newton-Krylov (JFNK) method. At each nonlinear step, VULCAN calculates the nonlinear residual R and computes first approximations of the Jacobian $\frac{\partial R}{\partial Q}$. The nonlinear residual is used directly by Enigma, and the approximate Jacobian is used by the block successive overrelaxation Gauss-Seidel preconditioner within PETSc. It should be noted that JFNK does require at least one more residual evaluation per iteration than a method that explicitly forms the Jacobian, and is not necessarily more cost effective than explicitly forming the Jacobian. The full second-order VULCAN stencil may include up to one hundred off-diagonal entries, and a complex chemically-reacting simulation may have Jacobian blocks as large as 50×50. The major benefit of JFNK is that the full and exact Jacobian does not need to be formed explicitly. This combination means that storing the full Jacobian matrix may require a prohibitively large amount of memory. Furthermore, it is not practical to form exact Jacobians for complex problems. The existing and proven Jacobians within VULCAN are leveraged by the JFNK preconditioner to accelerate solving the linear subproblem and to reduce the number of Krylov search directions ultimately used.

A. Globalization of Newton's Method with Pseudotransient Continuation

A drawback of Newton's method is that convergence to a root of the PDE is not guaranteed. The initial solution state may be far away from the root of the PDE, and a globalization technique is often required to avoid the algorithm stalling or diverging. The globalization strategy used in this work is PTC with the time step as the continuation parameter. To utilize PTC, the backward Euler time integration can be used to rewrite Eq. 2 introducing a time-based continuation term:

$$\left(\frac{V_i}{\Delta t_i} I + \frac{\partial R}{\partial Q}\right) \Delta Q = -R(Q^n) \tag{5}$$

where V_i is the volume of cell i and Δt_i is the timestep used in cell i. Alternatively, a fixed time step can be used across all cells (simply referred to as Δt). The timestep in Eq. 5 is dynamically controlled during convergence, such that Δt is small during the early stages of convergence and is adjusted to a large value near convergence to recover quadratic Newton convergence. The time step is controlled by adjusting the CFL number and the strategy used to adjust the CFL number is described later in this section.

The addition of the continuation parameter must be included in the residual used in Equation 4 so that the impact of the globalization parameter is included in the automatically computed Frechet Jacobians. The new residual operator used by the Frechet technique is

$$\tilde{R}(\tilde{Q}) = R(\tilde{Q}) + (\tilde{Q} - Q^n)\frac{\Delta t}{V} \tag{6}$$

$\tilde{Q}$ is the perturbation to the solution from the Frechet derivative:

$$\tilde{Q} = Q^n + \epsilon \Delta Q. \tag{7}$$

The final step of the nonlinear solver is a line-search. Solving Eq. 5 yields a potential solution update ΔQ. A line search may improve convergence and robustness by finding a solution update scaling factor, λ, from

$$Q^{n+1} = Q^n + \lambda \Delta Q \tag{8}$$

such that $\tilde{R}(Q^{n+1})$ is minimized.

A good CFL adjustment strategy is key to balancing between having a robust solver, and a fast one. [23**?** –25] The strategy used for this work solves for a proposed solution update ΔQ and scaling factor λ and then categorizes the previous step into three broad categories: a good update, an acceptable update, and a refused update. A step is labeled a good update so long as:

- the Krylov linear solver was able to converge the linear problem to three orders of relative magnitude
- $\lambda \geq 0.1$
- The nonlinear residual was reduced $||R(Q^{n+1})||_2 < ||R(Q^n)||_2$)

A step is labeled acceptable if

- the Krylov based linear solver was able to converge the linear problem to three orders of relative magnitude
- $\lambda \geq 0.1$
- The nonlinear residual did not grow too rapidly $||R(Q^{n+1})||_2 < 2||R(Q^n)||_2$)

Finally, a step is refused if the step is not categorized as either good or acceptable.

If a step is categorized as refused, then the solution is reset to the previous solution and the CFL is reduced by a factor of 20. If the step is categorized as good, then the CFL is increased by a scaling factor, typically 1.5. If the step was categorized as acceptable, the CFL is not changed at this stage.

Once the new CFL is set from the categorization stage there are a few instances where the new CFL is further overwritten. If at any time the new CFL would go out of user specified bounds, it is reset to be within this range. The range used for the cases in this paper were very forgiving with a minimum value of 10^{-3} and a maximum of 10^{8}. If the step was categorized as acceptable, but the number of Krylov search vectors was above a target of 50, then the CFL is reduced by 80% for the next step. Coupling the CFL to the effort required in the linear solver is done to limit the quadratically increasing cost of the Krylov linear solver in regions where full Newton convergence is not expected. If the solver achieves 20 good updates in a row then this rule is disabled. Finally, an optional augmentation is to limit the maximum CFL for the next 20 nonlinear iterations after a refused step. The new, temporary, maximum CFL is set to 80% of the CFL that was used during the failed step.

Even with tuning, the CFL to manage the cost of the linear problem, per nonlinear iteration; the overall cost of the JFNK scheme is much greater than that of VULCAN's existing defect-correction scheme. For each nonlinear iteration, the preconditioner is computed by calculating VULCAN's first-order Jacobians, and then the residual is computed many times. One residual evaluation is required for every Krylov search direction, plus every step of the line search requires an additional residual evaluation. Furthermore, each residual evaluation must be an accurate representation of $R(Q)$ including fully converging any nonlinear equations that are encountered calculating thermodynamic or transport properties or when calculating complex boundary conditions. With the existing defect-correction solver these quantities can be undersolved to further decrease the per-iteration cost. However, for JFNK, if the residual is not accurate with respect to the current solution state then the Newton step will be approximate and further degrade the convergence rate. Because of these increased costs, the JFNK scheme is cost effective only if it can solve the nonlinear problem using significantly fewer nonlinear iterations. On coarse grids, the JFNK scheme often is able to obtain quadratic, or near quadratic convergence. As the grid is refined, the JFNK convergence degrades to linear (or worse) convergence. The additional cost of JFNK is not necessarily a significant drawback on coarse to medium grids since the overall cost to solution is relatively small anyway. And the automated CFL control, as well as the ability to reset the solution to a previous "safe" solution state greatly reduces the human effort involved in the early adaptation cycles.

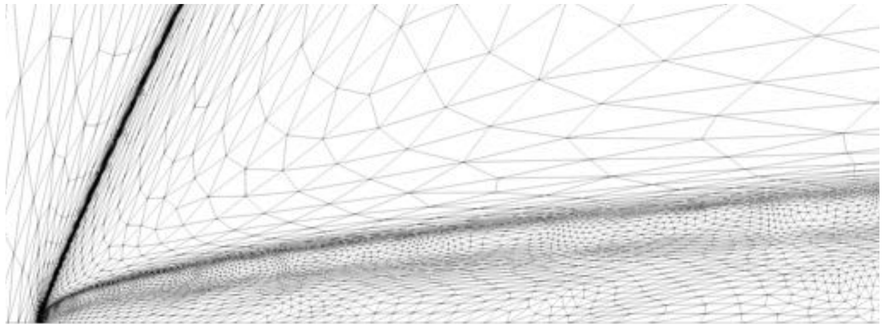
V. Results

Automated simplex mesh adaptation has been used extensively for subsonic and transonic applications where the sensor of interest is often Mach number. However, we are interested in leveraging this technology for hypersonic applications and, in particular, the prediction of wall heating. The first two test cases that follow were selected to closely evaluate wall heating rates on simplex adapted grids for well understood 2D laminar flows. The first is a high unit Reynolds number (26.4 million) flat plate at Mach 6 and the second is a lower unit Reynolds number (0.104) at a high 15.1 Mach number. The final case presented is a 3D configuration with a complex smooth geometry at Mach 6 with a unit Reynolds number of 6.69 million.

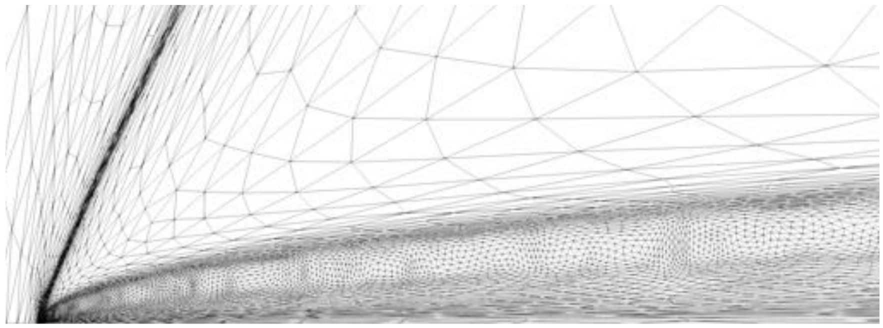
A. 2D Reynolds Number 26.4M Flat Plate at Mach 6

The first case presented is a 2D flat plate at Mach 6 with a Reynolds number of 26.4 million. A slip wall boundary condition is between x = -0.34 and x =0.0 and a 335.83 K isothermal wall starts at x = 0.0 to the extrapolation outflow condition at x = 2.0. The inflow static temperature was fixed at 63.01 K. VULCAN was used to obtain solutions using a second-order upwind-biased Fromme scheme and a multidimensional limiting process based on the Venkatakrishnan smooth limiter [3]. The limiter was not frozen at any point. The viscous gradients were obtained using the F-ANG approach described in Ref. [3].

Two different adaptation trajectories were run starting from the same initial grid with each grid adapted to control L_2 interpolation error in a specific sensor field of either Mach or static temperature. The cell-based sensor fields were averaged to the nodes with a volume-weighted average of the cells surrounding each node in the grid. Then, the Hessian was computed at nodes. The adaptation code *refine* was used to perform adaptation using the multiscale metric technique based on each of the sensor fields. Each series was run for 26 adaptation cycles resulting in grids with 170k triangles. Figure 2 shows a zoomed-in view near the leading edge of the flat plate for the final adapted meshes where anisotropic elements were formed along the diagonal discontinuity that begins at the flat plate leading edge. Figure 2 has been stretched by a factor of 10 in the wall-normal direction. The final Mach-adapted grid has more grid resolution in the streamwise direction, whereas the temperature-adapted grid has more grid resolution in the wall-normal direction.



(a) Final adapted mesh with Mach sensor.

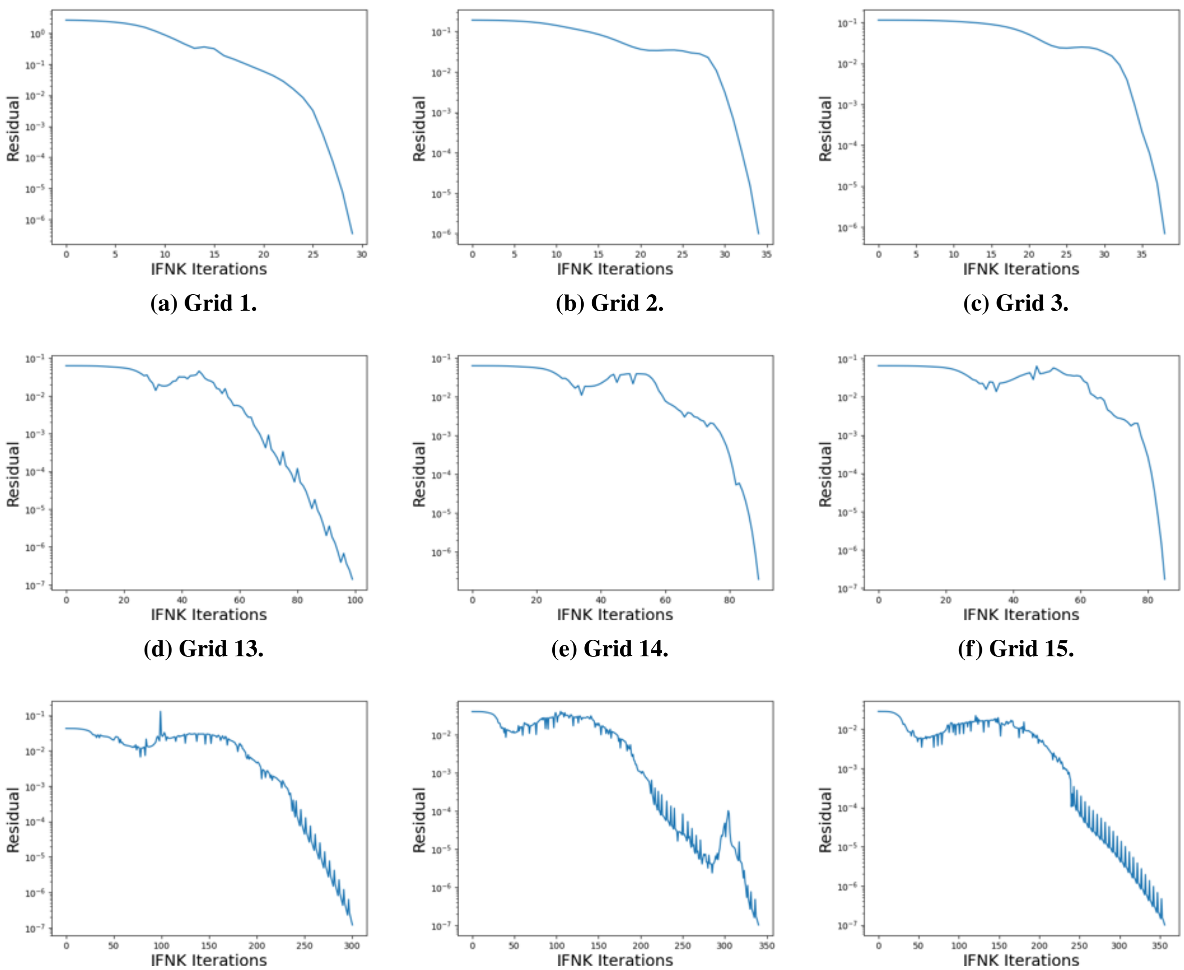


(b) Final adapted mesh with static temperature sensor.

Fig. 2 The final adpated grids for (a) Mach- and (b) static temperature-based adaptation. The coordinate direction normal to the surface has been scaled by a factor of 10. The temperature-based adapted grid has higher aspect ratio cells near the boundary layer than that of the Mach adapted grid.

Figure 3 shows the convergence history of the first three, middle three and final three grids. On the initial coarse grids, full Newton convergence is obtained and near Newton convergence is observed on the medium grids. However, the convergence rate degrades as the grids are refined, showing a mix of quadratic and linear convergence. The CFL controller rule requiring the CFL to stay under previously achieved maximum CFL was not enforced for these cases, since deep convergence was obtained without requiring it, and the number of nonlinear iterations required was small. However, the removal of the rule produces jagged convergence histories.

Figure 4 shows the wall heat flux and y^+ computed on each of the final grids obtained by adapting to control Mach number or static temperature. Both unstructured adapted grids are compared against a highly refined, 1.2 million cell, structured-grid solution obtained from VULCAN. From Figure 2, the Mach number adapted grid placed more grid points along the flat plate. However, it can be seen from Figures 4 (c) and (d) that the grids obtained from static-temperature adaptation has an overall smaller cell height at the wall. The wall heat flux obtained on the static temperature-adapted grid is significantly more accurate, and smoother, compared to the solution obtained from Mach number adaptation. Furthermore, the heat flux obtained from the static-temperature adapted grid compares favorably to the structured grid solution.

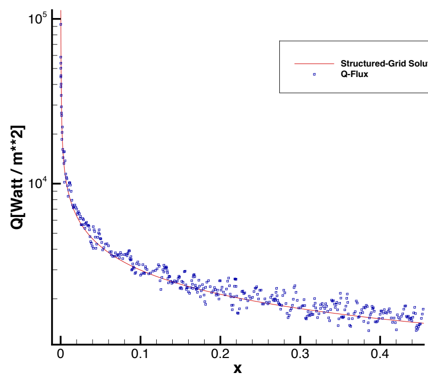


(a) Grid 1. (b) Grid 2. (c) Grid 3.

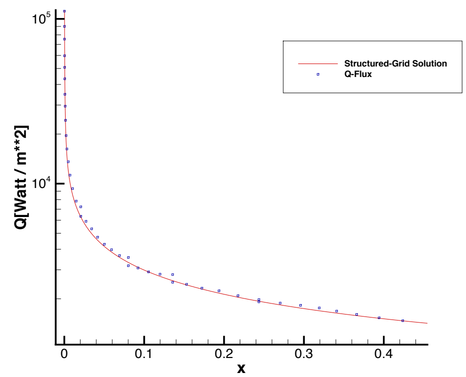
(d) Grid 13. (e) Grid 14. (f) Grid 15.

(g) Grid 24. (h) Grid 25. (i) Grid 26.

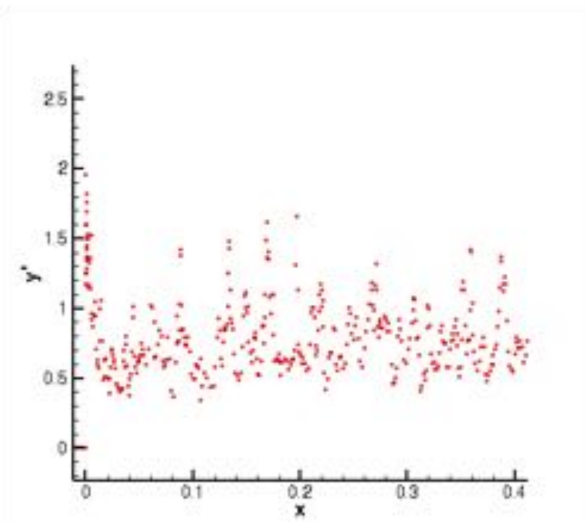
Fig. 3 Mach 6 flat plate convergence history for the first three, middle three, and final three grids in the adaptation sequence using the JFNK nonlinear solver.



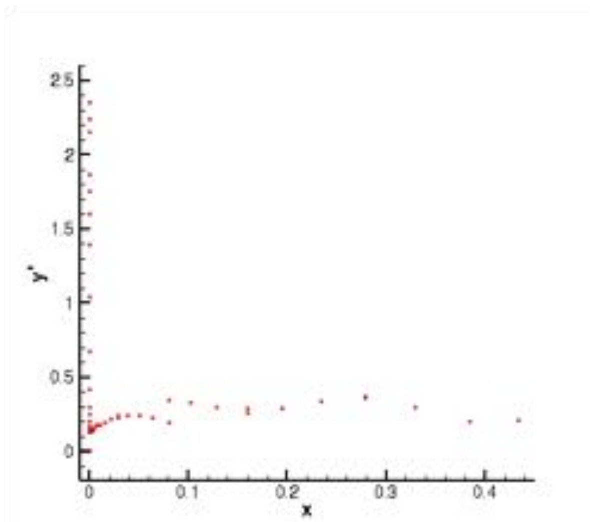
(a) Mach number adapted Q.



(b) Static temperature adapted Q.



(c) Mach number adapted y^+.

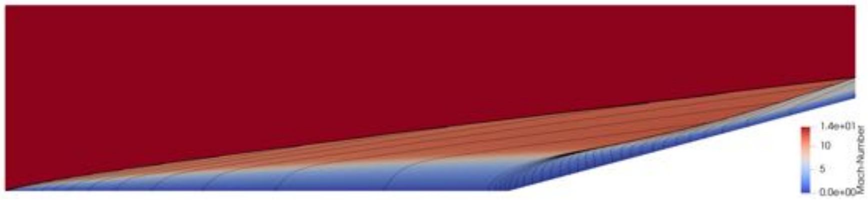


(d) Static temperature adapted y^+.

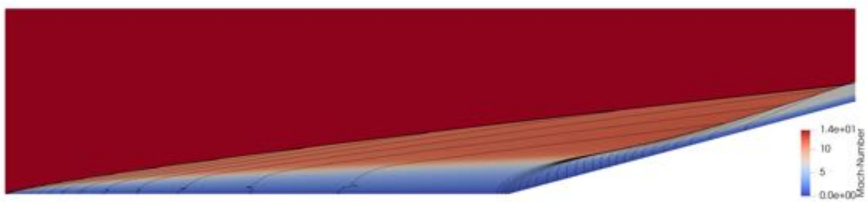
Fig. 4 Results for laminar flow over a flat plate at $M_\infty = 6.0$ and $Re_\infty = 26.4 \times 10^6$ on an adaptive triangular grid adapted based on the Mach number Hessian or the static temperature Hessian. The wall heat flux, Q, and y^+ are plotted at wall face centers. A highly refined structured-grid solution is shown for wall heat flux comparison.

B. Mach 14.1 Compression Corner

The next test case is a 15° laminar compression corner with $M_\infty = 14.1$ and $Re_\infty = 1.04 \times 10^5$ and an inflow static temperature of 72.2 K. The 297.0K isothermal wall starts at x = 0.0 and the corner is at x = 1.143. A fixed inflow boundary condition was placed at x = -0.255 and an extrapolation condition was used at x = 2.42. An overview of the test case is shown in Figure 5, with the Mach field and static pressure contours from the final grid obtained from both sensors.



(a) Mach adapted fine grid.



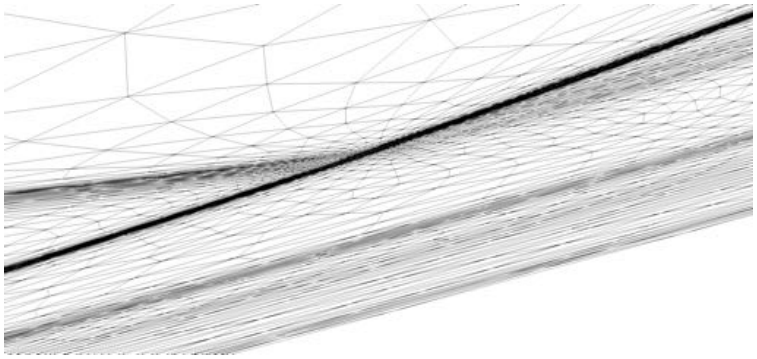
(b) Static temperature adapted fine grid.

Fig. 5 Mach field with contour lines of static pressure on the fine grid from Mach number- and static temperature-based adaptation trajectories.

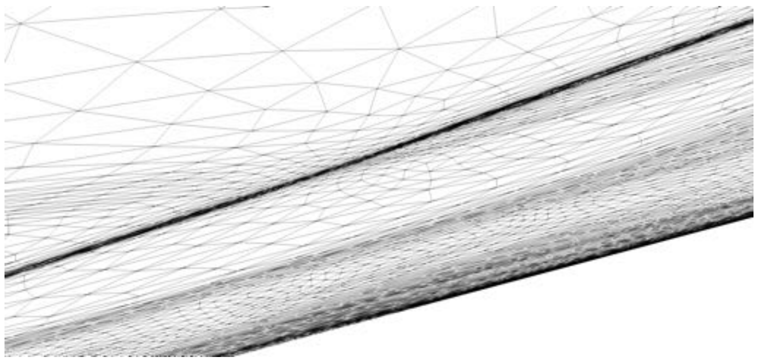
Figure 6 shows the very coarse (21k) triangle grids obtained at iteration 15 for both the Mach number and static temperature sensors. Using the temperature field to drive adaptation leads to more grid being allocated to the boundary layer and the grid is still highly refined at the transmitted shock. The contact discontinuity and the leading edge shocks are not as refined using the temperature-only error estimate. Notice that upon further adaptation many strong and subtle features of the Type VI shock are evident in the grid.

Figure 5 shows Mach number with static pressure contours at logarithmically spaced intervals comparing solutions obtained on the fine grid from the Mach number and static temperature trajectories. The grid obtained via Mach number adaptation in Figure 5a shows smoother static pressure contours upstream of the corner compared to the solution obtained via static temperature adaptation shown in Figure 5b.

Wall heat flux from coarse (42k triangles), medium (84k triangles), and fine (168k triangles) grids is shown in Figure 7 for both Mach number-and static temperature-based adaptation trajectories. Both Mach number- and static temperature-based trajectories show smoother wall heating as the grid is refined. However, static temperature-based adaptation yields much smoother wall heating for the same grid size when compared to Mach number-based adaptation. Both trajectories also show a drop in wall heating at the exit boundary condition that diminishes on finer grids. Computed y^+ values are shown in Figure 8. Both Mach number and static temperature trajectories show a y^+ cell height less than 1.0 for all grids, and under 0.5 on the fine grids with the static temperature-based adaption grids having a smaller y+ than the Mach number-based adaption grids. Finally, Figure 9 shows several fields derived from the VULCAN simulation on the final adapted grid using the static temperature-based adaptation.

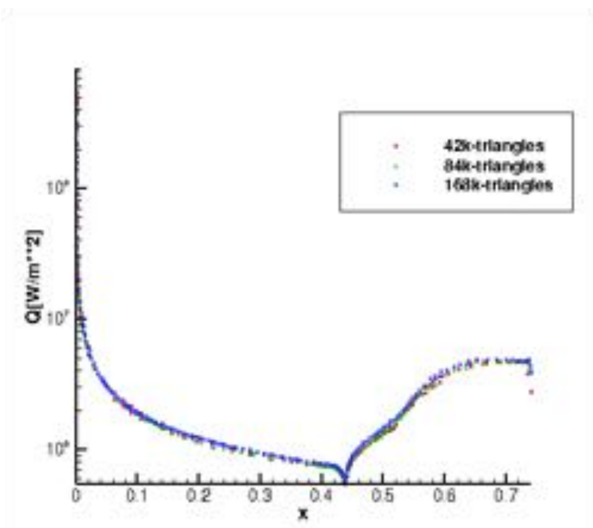


(a) 21k triangle grid with Mach-only error estimate.

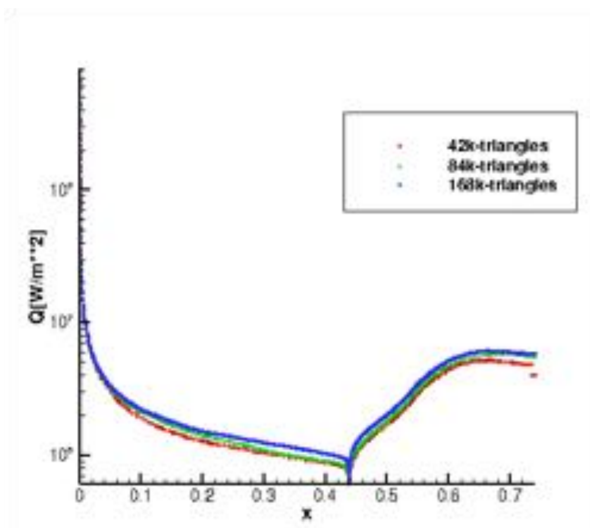


(b) 21k triangle grid with temperature-only error estimate.

Fig. 6 Grids obtained for each sensor field at iteration 15 of the adaptation cycle. Each grid has approximately 21k triangles.

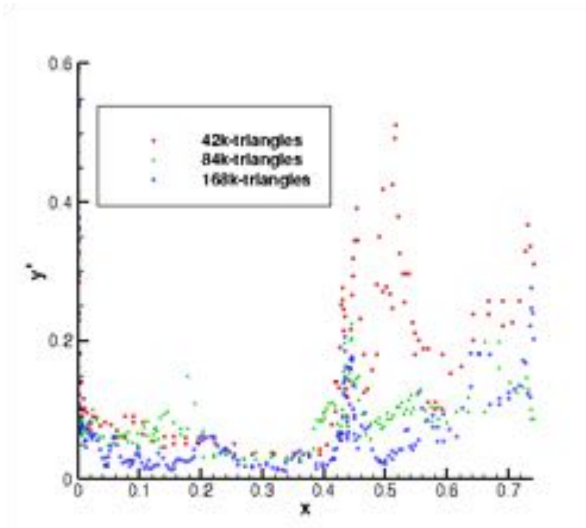


(a) Heat flux on Mach number adapted grid.

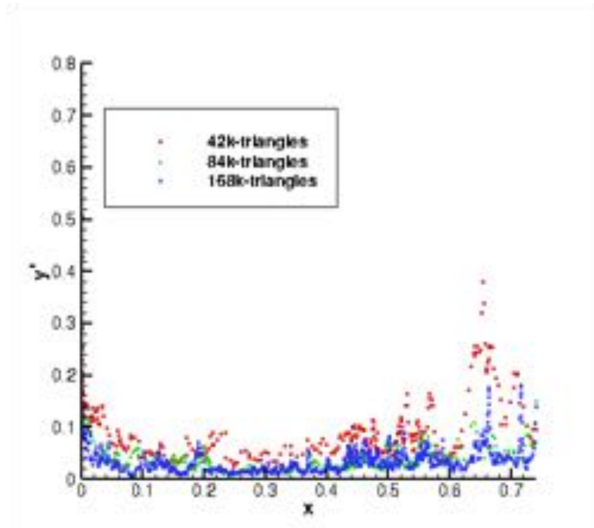


(b) heat flux on static temperature adapted grid.

Fig. 7 Wall heat flux on coarse (42k triangles), medium (84k triangles), and fine (168k triangles) for from Mach number- and static temperature-based adaptation trajectories.

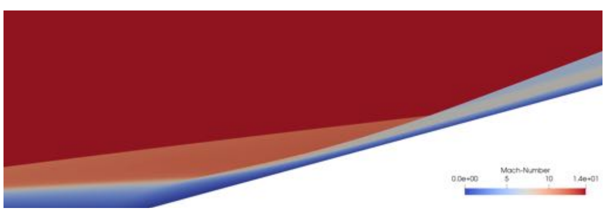


(a) y^+ on Mach number adapted grid.



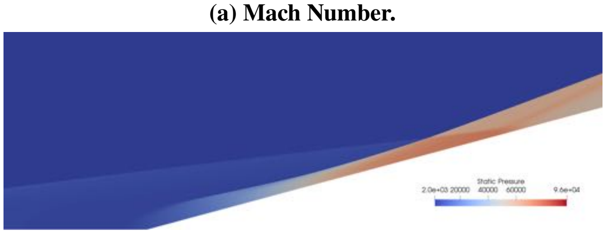
(b) y^+ on static temperature adapted grid.

Fig. 8 Computed y^+ on coarse, medium, and fine grids for Mach number and static temperature trajectories.

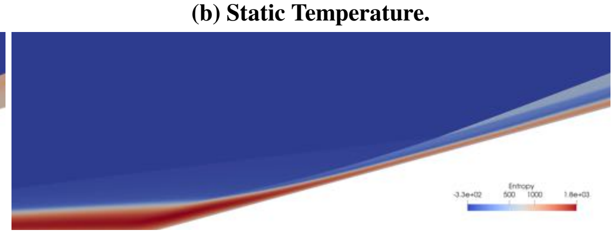


(a) Mach Number.

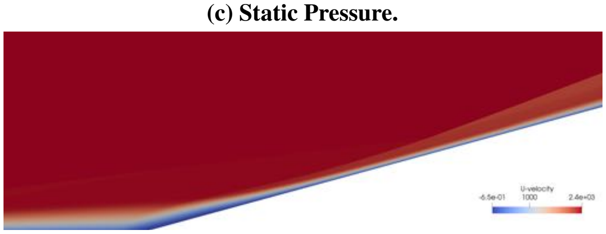
(b) Static Temperature.



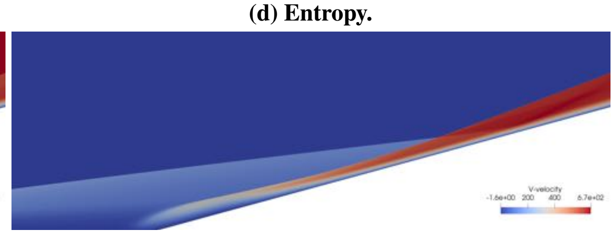
(c) Static Pressure.



(d) Entropy.



(e) X-Velocity.



(f) Y-Velocity.

Fig. 9 Results for a laminar flow over a 15-degree compression corner at $M_\infty = 14.1$ and $Re_\infty = 1.04$ million on an adaptive triangular grid using the static temperature sensor.

C. BOLT at Mach 6 and α 5, Re 6.69 Million

The final test case is a 3D case based on the BOLT geometry [26]. The test case was selected to begin exploration of using 3D geometries with VULCAN and anisotropic tetrahedral adaptation. An overview image of the case can be seen in Figure 10 showing the Mach field as well as the mesh on the vehicle, symmetry, and outflow surfaces. The grid was generated for a half-span domain and only the forebody flow is simulated at Mach 6 with an angle of attack of 5 degrees and a Reynolds number of 6.69 million and an inflow static temperature of 51.6 K. The vehicle surface was simulated using an isothermal viscous wall boundary condition with a temperature of 297.0 K. A fixed inflow condition was used and the exit plane was simulated with an extrapolation boundary condition.

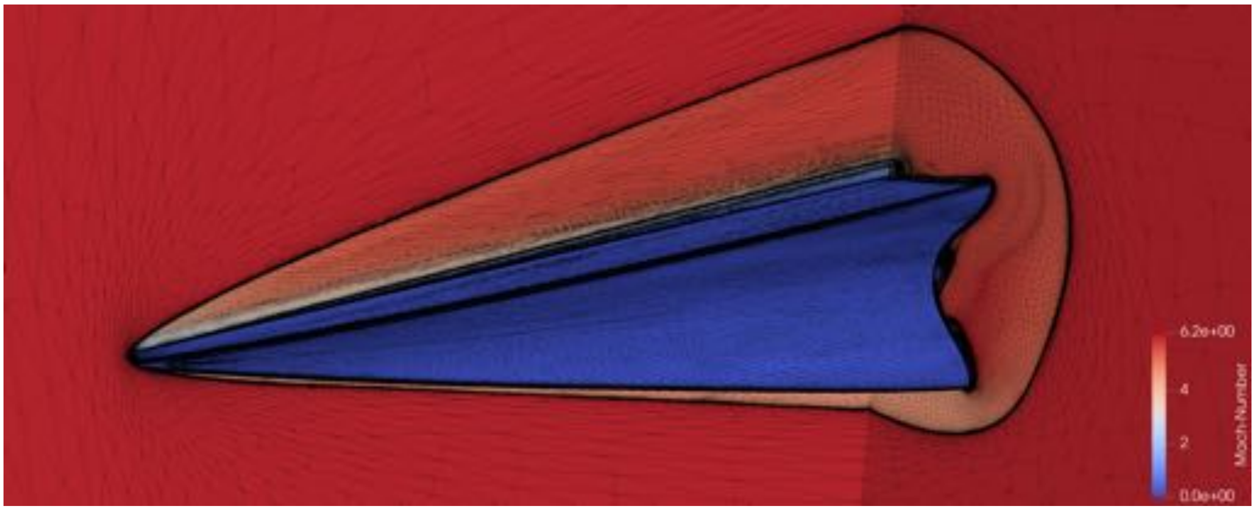


Fig. 10 Vehicle surface, symmetry plane, and outflow plane on the final adapted grid.

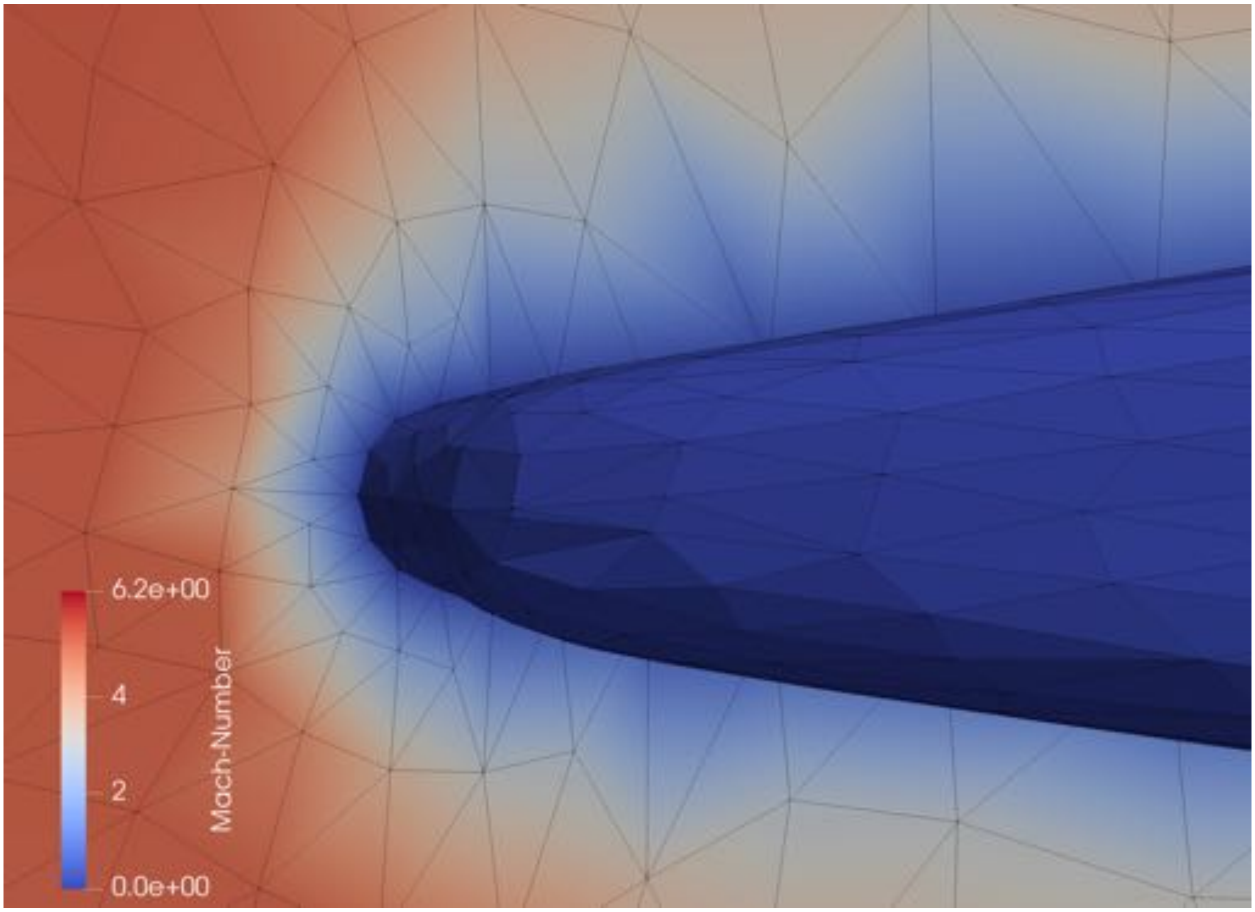
Unlike the previous cases, which were separately adapted to both Mach number and static temperature for comparison, this case was only adapted based on static temperature. The final grid, obtained after 55 adaptation cycles and 24 million tetrahedra, is shown in Figure 10. A closeup comparison of the blunt-body shock from the initial grid to the final grid is shown in Figure 11. A strong bow shock forms off the nose of the vehicle and is maintained to the exit plane. Several vortices form at the nose of the vehicle and are captured via adaptation as they propagate downstream.

This case was run using the JFNK solver for the first 40 adaptation cycles. Early grids demonstrated fast nonlinear convergence that again degraded as the mesh was refined. By adaptation cycle 40, it was clear that the JFNK solver would not be able to make substantial progress without resorting to a very large number of nonlinear iterations and the nonlinear scheme was switched back to Vulcan's existing defect correction based solver starting on adaptation cycle 41.

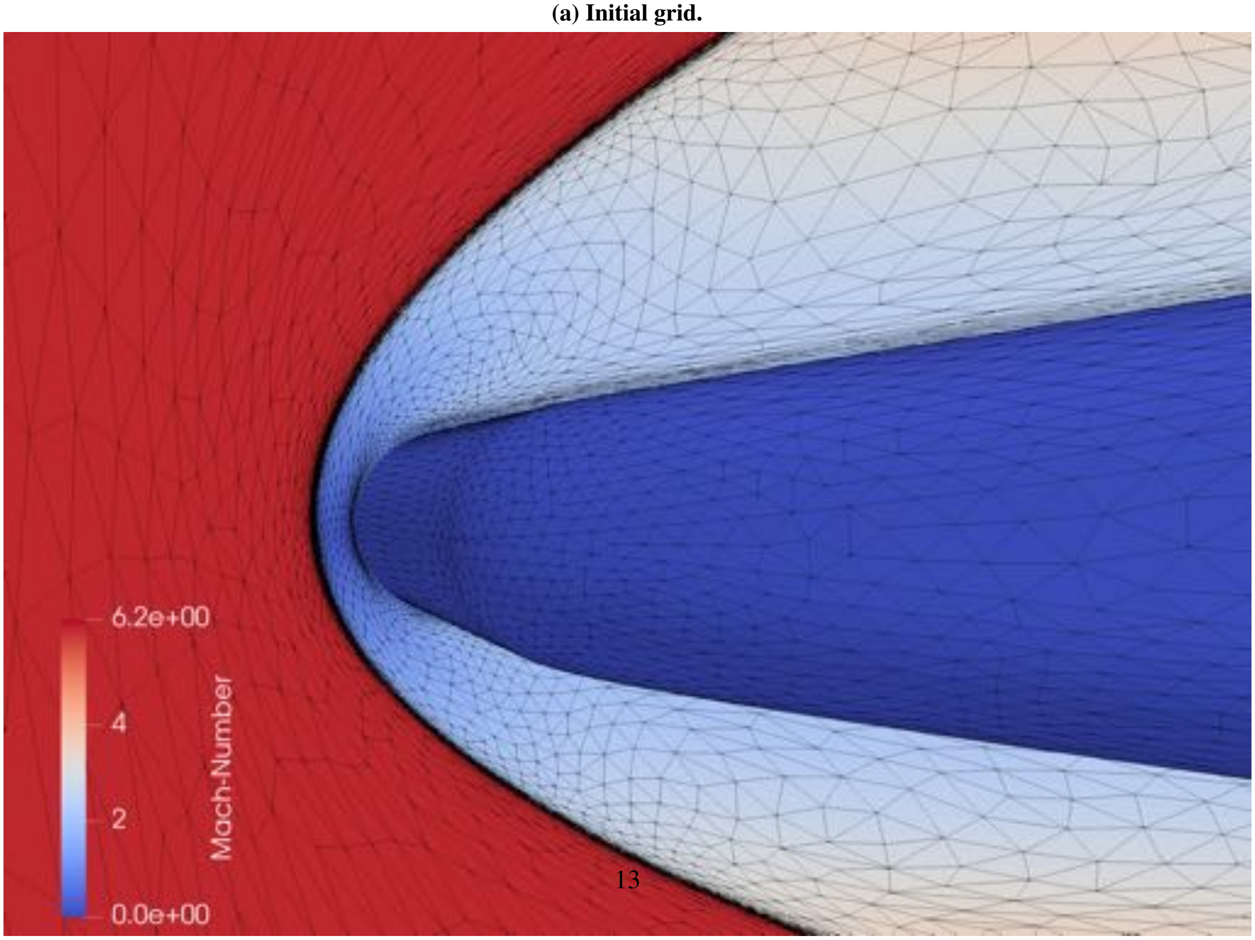
Figure 12 shows false color plots based on quantities of interest on the surface and outflow plane of the final adapted grid along with the grid on those surfaces. The grid was adapted to only control interpolation error in the static temperature field and we see that the final grid is also highly refined where the solution is changing rapidly for entropy, total enthalpy, and Mach number. However, regions of high gradient in the static pressure field are not as well captured by mesh refinement. This result is consistent with the previous compression corner case. Figure 13 shows the predicted heat flux on the final adapted grid along with the computed y^+ value for the first cell height at the vehicle surface. Despite 55 adaptation cycles and using the static temperature based sensor field the y^+ at the nose of the vehicle is well above 1.0 and the heat flux along the curved fin tips is not smooth.

VI. Summary and Future Work

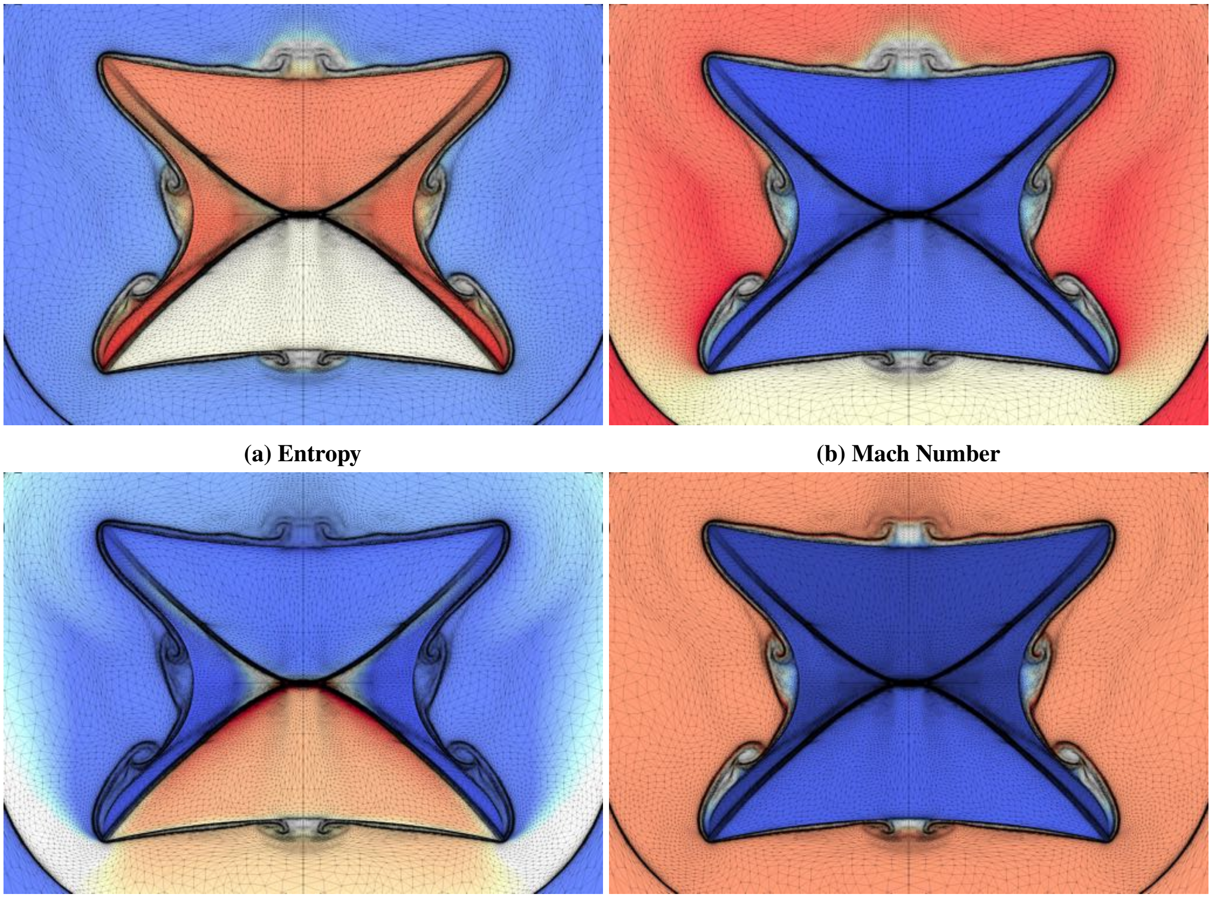
This paper has outlined recent work towards an automated unstructured mesh adaptation workflow with VULCAN for hypersonic applications. VULCAN has been integrated with a Jacobian-Free Newton-Krylov nonlinear solver package previously developed at the NASA Langley Research Center [10]. Two-dimensional calorically perfect laminar-flow test problems were presented for a high Reynolds number 2D viscous flat plate and a 2D compression corner at high Mach number. Anisotropic mesh adaptation was performed via *refine* to obtain anisotropic simplex meshes adapted to control the L_2 of interpolation error in two sensor fields of Mach number and static temperature. Wall heating rates were compared for the high Reynolds number flat plate and the high Mach number compression ramp. In both instances



(a) Initial grid.



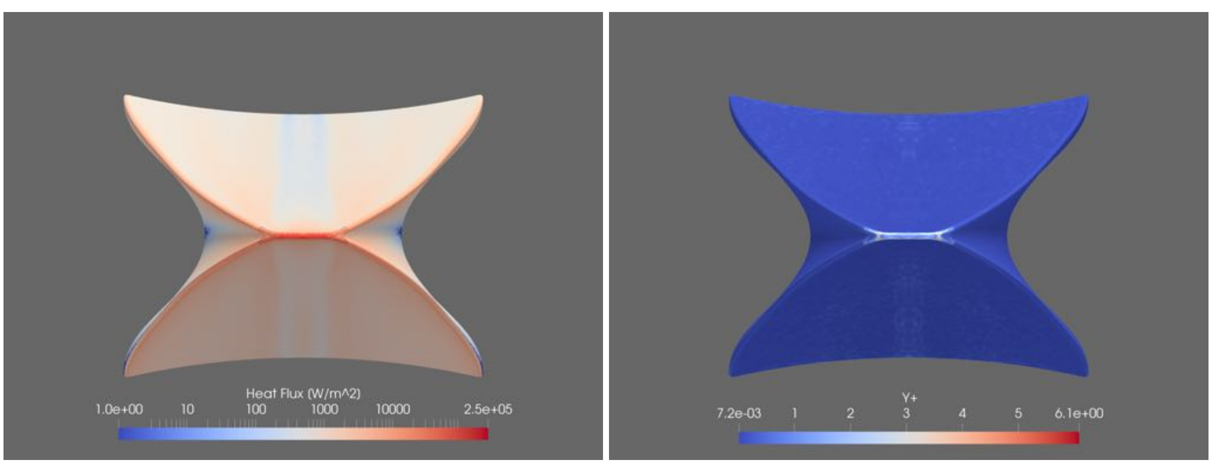
(b) Final grid.



(a) Entropy **(b) Mach Number**

(c) Static Pressure **(d) Total Enthalpy**

Fig. 12 Results for a laminar flow over the BOLT configuration at $M_\infty = 6$ and $Re_\infty = 6.69$ million on an adaptive tetrahedral grid.



(a) Heat flux. **(b) y^+.**

Fig. 13 The computed wall heat flux and y^+ values on the final BOLT grid.

the grids obtained using static temperature as the sensor field produced significantly smoother heating rates compared to the grids obtained via the Mach number sensor. Static pressure contours from the compression ramp case showed that grids obtained via the Mach number sensor were better suited to capture static pressure than grids adapted based on static temperature. A comparison of y^+ values on grids obtained from both sensors showed that for these cases the static temperature sensor produced overall smaller values of y^+ for the same sized grid.

Finally, a three-dimensional calorically perfect laminar flow test case was presented based on the BOLT configuration. Based on the results from the 2D test cases only the static temperature sensor was used for the BOLT case. Qualitative inspection of quantities of entropy, Mach number, static pressure, and total enthalpy were shown, demonstrating that regions of high variation in static pressure is not captured in grids adapted to the static temperature sensor. Finally, preliminary heat flux quantities on the BOLT configuration were shown.

This paper represents recent progress in ongoing work in many areas. The newly added JFNK nonlinear solver in VULCAN enables automated solution process during coarse adaptation cycles. These early grids are generated and solved quickly using an MPI-based compute cluster, often requiring only a few minutes for each grid. On more refined grids, the nonlinear convergence rate degrades even while using the accurate linearizations offered by the JFNK approach. Deep iterative convergence was not routinely achieved for the highly-adapted grids. There may be a number of obstacles in routinely, and automatically, achieving deep iterative convergence. One identified obstacle is the reconstruction limiter failing to converge, even while using the smoothly varying slope limiter of Venkatakrishnan [27], without resorting to freezing the limiter. A robust technique for identifying when to freeze the limiter has so far remained elusive, and the strategy of freezing the limiter is not overall desirable because a frozen limiter can allow non-physical undershoots and overshoots. Michalak and Ollivier-Gooch have identified circumstances where the Venkatakrishnan limiter is not sufficiently smooth for unstructured grids and have proposed an improvement [28]. Work is ongoing for both the nonlinear solution procedure as well as the numerical discretization with the goal of improving robustness and accuracy on complex grid systems.

The choice of adaptation sensor has been demonstrated to have a meaningful impact on heat-flux quantities. Yet, while static temperature appears to be an improvement over the more typical Mach number sensor for the cases presented, it has a few drawbacks. First, all the cases presented were solved using isothermal boundary conditions where the surface temperature was greater than the fluid static temperature. A future study is planned using both isothermal walls colder than the fluid static temperature as well as adiabatic boundaries. Additionally, the static temperature sensor was shown to underperform the Mach number sensor at capturing the static pressure field. Future adaptation strategies with VULCAN may be based on multiple sensor fields simultaneously through the intersection of multiple metric tensor fields [29].

Acknowledgments

The authors thank the Hypersonic Technology Project for their support and funding of this work. And to Michael Park, Hiro Nishikawa, and Robert Baurle for their help during the development of this work.

References

[1] Candler, G., Barnhardt, M., Drayna, T., Nompelis, I., Peterson, D., and Subbareddy, P., "Unstructured Grid Approaches for Accurate Aeroheating Simulations," AIAA Paper 2007–3959, 2007.

[2] Gnoffo, P. A., and White, J. A., "Computational Aerothermodynamic Simulation Issues on Unstructured Grids," AIAA Paper 2004–2371, 2004.

[3] Nishikawa, H., and White, J. A., "Face-Averaged Nodal-Gradient Approach to Cell-Centered Finite-Volume Method on Triangular Grids," AIAA Paper 2019–3636, 2019.

[4] Nishikawa, H., and White, J. A., "An Efficient Cell-Centered Finite-Volume Method with Face-Averaged Nodal-Gradients for Triangular Grids," *Journal of Computational Physics*, Vol. 411, 2020, p. 109423. doi:10.1016/j.jcp.2020.109423.

[5] White, J. A., Nishikawa, H., and Baurle, R. A., "A 3-D Nodal-Averaged Gradient Approach For Unstructured-Grid Cell-Centered Finite-Volume Methods For Application to Turbulent Hypersonic Flow," AIAA Paper 2020–0652, 2020.

[6] Alauzet, F., and Loseille, A., "A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics," *Computer-Aided Design*, Vol. 72, 2016, pp. 13–39. doi:10.1016/j.cad.2015.09.005, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[7] Balan, A., Park, M. A., Anderson, W. K., Kamenetskiy, D. S., Krakos, J. A., Michal, T., and Alauzet, F., "Verification of Anisotropic Mesh Adaptation for RANS Simulations over ONERA M6 Wing," *AIAA Journal*, Vol. 58, No. 4, 2020, pp. 1550–1565. doi:10.2514/1.J059158.

[8] Galbraith, M. C., Caplan, P. C., Carson, H. A., Park, M. A., Balan, A., Anderson, W. K., Michal, T., Krakos, J. A., Kamenetskiy, D. S., Loseille, A., Alauzet, F., Frazza, L., and Barral, N., "Verification of Unstructured Grid Adaptation Components," *AIAA Journal*, Vol. 58, No. 9, 2020, pp. 3947–3962. doi:10.2514/1.J058783.

[9] Michal, T., Krakos, J., Kamenetskiy, D., Galbraith, M., Ursachi, C.-I., Park, M. A., Anderson, W. K., Alauzet, F., and Loseille, A., "Comparing Unstructured Adaptive Mesh Solutions for the High Lift Common Research Model Airfoil," AIAA Paper 2020–3219, 2020.

[10] Thompson, K. B., and O'Connell, M. D., "Streamlined Convergence Acceleration for CFD Codes," AIAA Paper 2018–3709, 2019.

[11] Park, M. A., "Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows," Ph.D. thesis, Massachusetts Institute of Technology, Sep. 2008. doi:1721.1/46363.

[12] Loseille, A., Dervieux, A., Frey, P. J., and Alauzet, F., "Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes," AIAA Paper 2007–4186, 2007.

[13] Loseille, A., and Alauzet, F., "Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 38–60. doi:10.1137/090754078.

[14] Loseille, A., and Alauzet, F., "Continuous Mesh Framework Part II: Validations and Applications," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 61–86. doi:10.1137/10078654X.

[15] Kleb, B., Park, M. A., Wood, W. A., Bibb, K. L., Thompson, K. B., Gomez, R. J., III, and Tesch, S. H., "Sketch-to-Solution: An Exploration of Viscous CFD with Automatic Grids," AIAA Paper 2019–2948, 2019.

[16] Haimes, R., and Dannenhoffer, J. F., III, "The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry," AIAA Paper 2013–3073, 2013.

[17] Dannenhoffer, J. F., III, "OpenCSM: An Open-Source Constructive Solid Modeler for MDAO," AIAA Paper 2013–701, 2013.

[18] Haimes, R., and Drela, M., "On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design," AIAA Paper 2012–683, 2012.

[19] Haimes, R., and Dannenhoffer, J. F., III, "EGADSlite: A Lightweight Geometry Kernel for HPC," AIAA Paper 2018–1401, 2018.

[20] Park, M. A., Kleb, B., Jones, W. T., Krakos, J. A., Michal, T., Loseille, A., Haimes, R., and Dannenhoffer, J. F., III, "Geometry Modeling for Unstructured Mesh Adaptation," AIAA Paper 2019–2946, 2019.

[21] Si, H., "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 41, No. 2, 2015, pp. 11:1–11:36. doi:10.1145/2629697.

[22] Abhyankar, S., Brown, J., Constantinescu, E. M., Ghosh, D., Smith, B. F., and Zhang, H., "PETSc/TS: A Modern Scalable ODE/DAE Solver Library," *Computing Research Repository (CoRR)*, Vol. Numerical Analysis (math.NA), No. arXiv:1806.01437, 2018. URL http://arxiv.org/abs/1302.6066.

[23] Pandya, M. J., Diskin, B., Thomas, J. L., and Frink, N. T., "Assessment of USM3D Hierarchical Adaptive Nonlinear Method Preconditioners for Three-Dimensional Cases," *AIAA Journal*, Vol. 55, No. 10, 2017, pp. 3409–3424. doi:10.2514/1.J055823.

[24] Anderson, W. K., Newman, J. C., and Karman, S. L., "Stabilized Finite Elements in FUN3D," *Journal of Aircraft*, Vol. 55, No. 2, 2018, pp. 696–714. doi:10.2514/1.C034482.

[25] Burgess, N., and Glasby, R., "Advances in Numerical Methods for CREATE-AV Analysis Tools," AIAA Paper 2014–417, 2014.

[26] Berry, S. A., Mason, M. L., Greene, F., King, R., Rieken, E., and Basore, K., "LaRC Aerothermodynamic Ground Tests in Support of BOLT Flight Experiment," AIAA Paper 2019-91, 2019.

[27] Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *Journal of Computational Physics*, Vol. 118, No. 1, 1995, pp. 120–130.

[28] Michalak, "Differentiability of Slope Limiters on Unstructured Grids," *Proceedings of fourteenth annual conference of the computational fluid dynamics society of Canada*, ????

[29] Castro-Diáz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., "Anisotropic Unstructured Mesh Adaptation for Flow Simulations," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 4, 1997, pp. 475–491. doi:10.1002/(SICI)1097-0363(19970830)25:4<475::AID-FLD575>3.0.CO;2-6.