

Implementation of Adaptive Mesh Refinement in an Implicit Unstructured Finite-Volume Flow Solver

Alan M. Schwing ^{*}, Ioannis Nompelis [†], and Graham V. Candler [‡]

Department of Aerospace Engineering and Mechanics

University of Minnesota, Minneapolis, MN, 55455

This paper explores the implementation of adaptive mesh refinement in an implicit unstructured finite-volume solver. Unsteady and steady problems are considered. The effect on the recovery of high-order numerics is explored and the results are favorable. Important to this work is the ability to provide a path for efficient, implicit time advancement. A refinement metric using a simple sensor based on undivided differences is discussed and applied to a model problem with practical applications: a shock-shock interaction on a hypersonic, inviscid double-wedge. Cases are compared to uniform grids without the use of adapted meshes in order to assess error and computational expense. Discussion of difficulties and future work prepare this method for additional research. The potential for this approach in more complicated flows is described.

I. Introduction

Ideal computational fluid dynamic (CFD) calculations require a grid that efficiently discretizes the spatial domain. For complex geometries, grid generation frequently demands a significant percentage of a project's effort. Generating a grid is as much an art as a science and can be the most subjective portion of a given analysis. Changing the prevailing flow characteristics (Mach number, Reynolds number, or angle of attack) affect the resulting solution and the location of flow separation, shock standoff distance, and other critical features often change. These changes necessitate a modification to the grid with cell resolution added, removed, or moved in order to efficiently capture all relevant flow phenomena. While adapting a grid to changing conditions, it is necessary to avoid adding grid elements where they are not needed and unnecessarily inflating computational cost.

To handle these sometimes competing requirements - adding more elements where needed without adding superfluous ones - the most obvious course of action is for the user to generate grids specific to each flow condition that will be considered in a given analysis. These grids contain adequate densities of cells in locations where the researcher feels is optimum for those specific conditions. With a vast case matrix or large number of vehicle configurations, grid generation quickly becomes an even larger percentage of the user's time. Additionally, hand-crafting grids may introduce changes in grid topology that muddle an otherwise clean comparison between two different conditions.

Grid generation is often automated to reduce the time required and to handle the logistics involved in enforcing similarity between distinct grid systems. Several methods are popular: automated grid translation and smoothing, using overset grids, and/or leveraging unstructured elements to provide topological flexibility. Each of these solutions remove some of the subjectivity from the grid generation process and can help ensure robust comparisons. Unfortunately, they do not completely 'solve' all problems encountered with grid generation and can introduce issues of their own. In most cases, there is still a non-trivial amount of time required to properly set-up the mechanisms for automation. Some methods reduce the time required for grid generation, but require more involved numerics and additional time for the flow computation.

^{*}Graduate Student, AIAA Student Member.

[†]Research Associate, AIAA Senior Member.

[‡]Professor, AIAA Fellow

Another issue is that these methods still require the researcher to have a priori knowledge about where to best place grid points for accurate results. For simple flows this might not be an issue, but for non-trivial flowfields or complicated vehicle geometries this can be a significant problem. With an unsteady flow, important features are likely moving in space with time. To properly resolve these features, adequate grid is required throughout the region containing the unsteadiness. An example of these two issues is flow over a bluff body shedding an oscillating wake. It might require a large number of grid points to resolve flow separation or shock shape and location. These locations will change as freestream parameters are modified and can be unsteady at a particular condition. In addition, the wake's shear layer and its myriad vortical structures need to be resolved as they convect downstream.

Research has progressed in the area of dynamic grid adaptation in order to help remedy some of these issues. It is not a new technique and was pioneered by a number of authors over many years, beginning with the work of Berger, Oliger, and Colella.^{1,2} The theme of the method is to select portions of the computational volume that contain important features or have a measurably high error. These locations are then locally refined by the addition of more grid elements during the course of the CFD run. This reduces the error and better resolves developing flow structure. Most methods also allow for these locations to move while the code is running in order to track unsteady phenomena. If appropriate feature detection or error estimation is used, this can remove or relax the strict requirement for a priori knowledge of the flow behavior. This allows the use of an initially coarse, more general grid that will be refined as needed by the flow solver. Local adaptation can improve grid quality to maintain numerical fidelity without an arduous demand on the researcher's time or expertise.

This technique is frequently referred to as Adaptive Mesh Refinement (AMR). Grids generated using AMR can be categorized by the presence or absence of hanging nodes. Hanging nodes are generated on the boundaries of refinement when a cell is refined, but its neighbor is not. This results in a face that is subdivided and a grid line that is truncated at the face. These grids are also described as being conformal (without hanging nodes) or non-conformal (with hanging nodes). As will be discussed later, the finite-volume approach lends itself to application on both conformal and non-conformal meshes.

When dealing with non-conformal meshes and multiple grid levels, there can be ambiguity and with how to construct stencils for high-order accurate numerical fluxes. Such methods frequently require accurate reconstruction of the spatial gradients and larger numerical stencils. For sensitive flowfields with a large range of frequency content, this is an important concern. It must be demonstrated that the AMR technique does not contribute to the error and that the improved efficiency does not come at the cost of solution quality. A portion of this work is devoted to ensuring that the order of accuracy can be maintained between uniformly refined grids and selectively refined, AMR grids.

For flows at high Reynolds numbers, viscous interaction can be limited to regions near the body. A boundary layer develops in the immediate vicinity of the vehicle due to the shearing associated with a non-slip wall. To resolve this strong shear, cells near the wall boundaries are constrained to having small heights (and usually large aspect ratios). This is done to ensure that the boundary layer is resolved appropriately. These cells typically have the smallest spacing of all cells in the computational domain and limit the maximum explicit time step allowable in a numerical time advancement scheme. In order to improve throughput and reduce the amount of time required for a simulation, it is desired to take larger time steps than those dictated by these small cells. For this reason, users turn to implicit time advancement methods.

The authors are interested in AMR for applications involving hypersonic, reacting, and turbulent flows. This work explores the requirements placed on AMR grids and develops the underpinnings necessary for further expansion. Of key importance are the ability to maintain high-order numerics, provide an avenue for efficient implicit time integration, and include AMR refinement criteria relevant for shock-dominated flows.

Many other researchers are also exploring this space. Some particularly relevant recent work has explored the use of implicit AMR in a finite-volume framework.^{3,4} This work employed structured, multi-block grids with a block-based refinement technique. Implicit time advancement was achieved by using an iterative Newton method coupled with a generalized minimum residual (GMRES) solver. The resulting method is parallel and shows success on simulating two-dimensional partial differential equations with applicability to aerospace problems. Our work differs in three points: the specifics of the implicit solve, the relaxation of the requirement for block-based refinement, and the fundamentally unstructured nature of the approach and future grid smoothing.

II. Motivational Problems

One problem that illustrates an important use for the AMR method in an implicit CFD solver is that of a double cone in supersonic flow. Major flow features are shown in Fig. 1. The solution for flow over this shape is detailed and contains many fine-scale features that require significant time to develop. Flow over this shape can be unsteady and can include large sections of subsonic and separated fluid. Implicit methods are required to accelerate the development of flow structures.

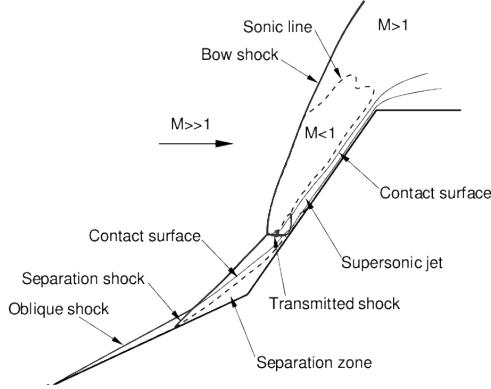
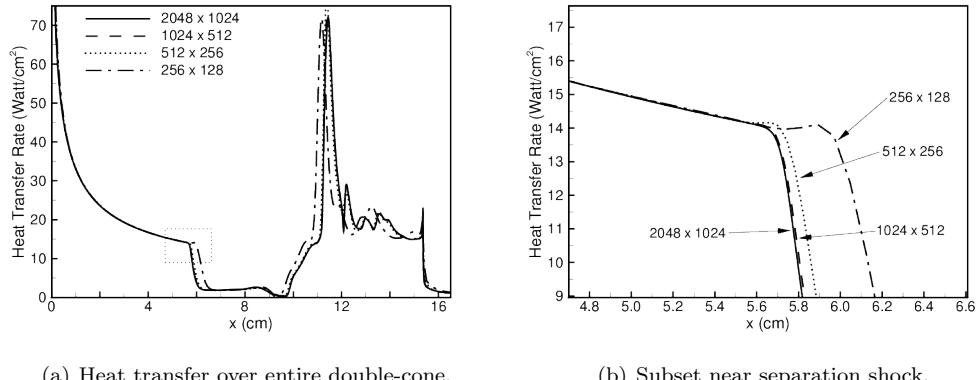


Figure 1. Important features in hypersonic flow near a double-cone.⁵

Figure 2 shows the results of a grid sensitivity analysis using fixed, unadapted grids for this problem. As is shown in Fig. 2(a), the solutions are identical for all grid resolutions upstream of about 5.5 cm. This indicates that physics were resolved adequately on the coarser grids. At the separation shock, the solutions diverge (see Fig. 2(b) for more detailed view) and motivate the additional refinement.

This model problem can be compared to a control surface on an aerospace vehicle. For such a control surface, accurately predicting the location and magnitude of the separated region will determine control efficiency and design requirements. Furthermore, it is unlikely that the entire vehicle will require as fine of grid resolution as the region containing the ramp. The ability to use adaptive, local refinement could lower grid requirements and enable much more rapid analysis.



(a) Heat transfer over entire double-cone.

(b) Subset near separation shock.

Figure 2. Illustration of grid convergence study on uniformly refined meshes.⁵

A second problem that highlights the need and utility of this method is that of a bluff body capsule. Such designs are important for nearly all planetary reentry vehicles currently under development at NASA and private aerospace companies. As mentioned previously, there are unsteady quantities near the vehicle and in the wake due to the aerodynamics and potential plume sources on the body. Targeted refinement with implicit AMR could allow much greater fidelity at reduced expense for problems of this type.

The wake of the vehicle is highly sensitive to separation location and can be an unsteady phenomenon. Separation location is influenced by a number of factors including grid resolution. Vehicle aerodynamics and aerothermodynamic requirements are dependent on the resolution of the wake. Not only are surface pressures and integrated loads affected, but hot gas can be entrained in recirculation and drive thermal considerations.

Wake environments are dominated by vortical structures, shear layer breakdown, and turbulence - all unsteady phenomena. For static grid generation, unsteady structures require adequate resolution across a large volume that encompasses the range of locations a feature may exhibit. The ability to track moving structures and adjust the grid accordingly has the potential to greatly lower the fixed cost incurred by clustering high resolution cells throughout the wake.

III. Computational Methodology

A. Flow Solver

The solution to the inviscid, compressible Navier-Stokes equations (Euler equations) is computed using a finite volume scheme. In conservation law form, the compressible Navier-Stokes equations can be written as:

$$\frac{\partial U}{\partial t} + \nabla \cdot (\vec{F}_c - \vec{F}_v) = 0$$

where $U = (\rho, \rho u, \rho v, \rho w, E)^T$ is the array of conserved variables and \vec{F}_c and \vec{F}_v are the convective and viscous fluxes, respectively. ρ is the density, ρu , ρv , and ρw are the three-dimensional components of momentum, and E is the total energy per unit volume. For the inviscid, Euler equations, $\vec{F}_v = 0$. We will now use \vec{F} to indicate only the convective fluxes, \vec{F}_c . The convective fluxes exhibit many important features of the Navier-Stokes equations for high Reynolds number flows. There are well established flux formulations each with associated levels of accuracy. Numerical time integration and handling of the non-linear terms are important and determine the overall success of a computational approach. For this reason, recovering a high-order, accurate representation of the convective fluxes on grids generated using AMR is important to the authors.

Cell averaged values (\bar{U}) are obtained using explicit and implicit numerical time integration from an initial condition. The finite volume formulation allows for arbitrary polyhedra defined by a cell volume and a number of bounding faces. We solve a weak form of the governing equations with these cell averaged values. Numerical fluxes are calculated at each of the faces and by employing the divergence theorem over one such computational cell, a discrete representation of the Euler equations can be derived. The Euler equations for a cell with an arbitrary number of faces is:

$$V \frac{\partial \bar{U}}{\partial t} + \sum_{faces} [\vec{F} \cdot \hat{n} S] = 0 \quad \text{or} \quad \frac{\partial \bar{U}}{\partial t} = -\frac{1}{V} \sum_{faces} [\vec{F} \cdot \hat{n} S]$$

with V being the cell volume, S the face area, and \hat{n} the outward pointing unit normal to the face. This form of the governing equations lends itself to unstructured grids. As long as sufficient connectivity exists that links cells to their surrounding faces or vice-versa, it is independent of any defined ordering of the discrete cells.

A useful deconstruction of the face fluxes represents \vec{F} as $\vec{F} = \vec{F}_- + \vec{F}_+$ where \vec{F}_- and \vec{F}_+ are upwinded components of the total flux in the direction of the positive and negative running eigenvalues. With outward pointing normals, as in the equation above, all of the upwinded fluxes \vec{F}_+ depend on cell-averaged quantities in the current cell and then sum of \vec{F}_- depend on neighboring cells. The discrete equation representing a first-order accurate method can be written as:

$$\frac{\partial \bar{U}}{\partial t} = -\frac{1}{V} \sum_{faces} [(\vec{F}_- + \vec{F}_+) \cdot \hat{n} S]$$

Higher-order methods can be developed by augmenting the numerical stencil used to generate \vec{F}_+ and \vec{F}_- or by changing the flux deconstruction entirely. Aspect of this are briefly discussed in a following section.

Using a first-order, explicit time integration scheme between time level n and $n+1$, $\frac{\partial \bar{U}}{\partial t}$ is approximated as $\frac{\bar{U}^{n+1} - \bar{U}^n}{\Delta t}$. For such a scheme, the fluxes are all calculated at time level n . This method can be written as:

$$\bar{U}^{n+1} = \bar{U}^n - \frac{\Delta t}{V} \sum_{faces} \left[(\vec{F}_- + \vec{F}_+) \cdot \hat{n} S \right] = \bar{U}^n + \Delta \bar{U}$$

with $\Delta \bar{U}$ being shorthand for the explicit update to the cell-averaged value.

For a conformal grid with hexahedral cells, there are six faces to every element. By generalizing the method using a summation over the faces of a given cell it is clear to see how this method can be extended not only to non-hexahedral cells, but to non-conformal grids as well. It is this flexibility that is leveraged when applying the finite-volume formulation to grids obtained with AMR.

B. Adaptive Mesh Refinement Implementation

AMR builds on a given grid and adds successively finer representations of that grid in order to reduce truncation error and improve computational resolution. To differentiate between the varying generations of cells, those that are created by a subdivision are considered to be one ‘level’ finer than their corresponding parent cell. This is consistent with the terminology initially laid out by Berger.¹ The initial grid contains cells of level 0. Cells created by the subdivision of a level 0 cell are categorized as level 1 cells, for example.

Grids used in this work are constrained to include only three-dimensional, six-sided polyhedra (hexahedrals). The hexahedral cells are generally not cubic and may have an arbitrary skewness. For hypersonic applications, numerical error is introduced when trying to capture discontinuities using non-hexahedral shapes.⁶ Also, by including only hexahedral cells, it greatly simplifies the generation of high-order numerical flux stencils.

In a general sense, there is no limit to the number of cells that a parent cell can be divided into. An obvious choice for subdivision is to refine a cell by reducing its size by a factor of two in each solution direction. This is also referred to as isotropic subdivision. In a three-dimensional cell, this would equate to dividing a cell into eight smaller cells. Depending on the skewness of a hexahedral element, these new cells need not have equal size. Anisotropic subdivision allows for more flexibility in the generation of the grid and is used by a number of researchers in the field.^{4,7,8} It does, however, increase the complexity of the AMR procedure and can lead to large changes in cell aspect ratio and size in localized regions.

The work presented here constrains refinement to be isotropic in each of the solution directions such that grid density in a parent cell is doubled along each edge. Since the flow solver is written in 3-D, all cells are hexahedrals. When running cases with boundaries that indicate 1-D or 2-D problems, the code only subdivides the cell into the relevant directions. See Fig. 3 for an illustration of the three types of subdivision considered.

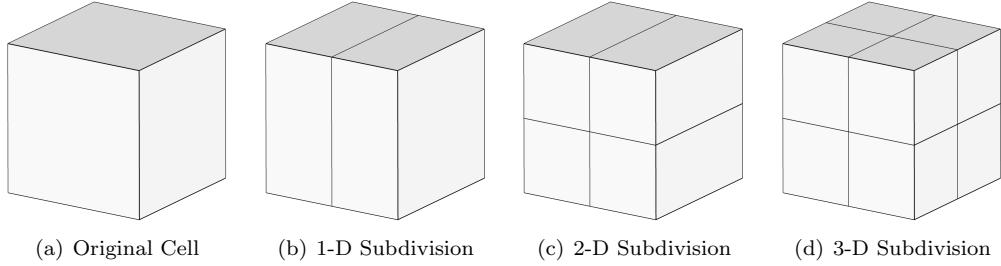


Figure 3. Illustration of three types of AMR subdivision from an original hexahedral cell.

Working with an unstructured grid, each cell can be refined independent of the others. This provides a great deal of flexibility and reduces the creation of superfluous elements. Octree data structures for each cell and face keep track of the parent child relationships and connectivity arrays are updated at the time of cell creation. This added flexibility comes at a cost: there is a great deal of bookkeeping required in order to handle connectivity and it must be updated every time a cell is refined or coarsened.

In addition to refining the grid, it is often necessary to coarsen as well. For unsteady problems, this requirement is due to the physical movement of a feature that requires increased grid resolution. AMR tracks moving features and without a mechanism for removing elements, the grid becomes cluttered with cells that are no longer required. Steady problems can benefit from coarsening as well. With improved

resolution, features may move to a more accurate location. Unfortunately, this can cause oscillations if the refined region was necessary in order to correctly determine flow structure.

Solution quantities are conserved during the refinement and coarsening operations. For refinement, each of the child cells are initialized to the solution variables in the coarse parent cell. When replacing a set of child cells with their parent cell during coarsening, the parent cell's solution variables are overwritten with the volume-averaged quantities from the child cells.

C. Refinement Criteria and Procedure

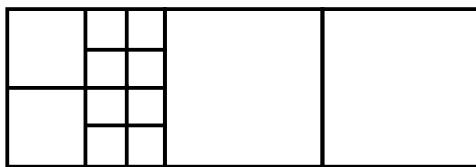
A simple refinement criteria is implemented in this work based on the differences across grid faces. This is a local evaluation and does not consider global effects of refinement. Other methods are also popular in the literature. Multiscale methods as originally proposed by Harten have proven successful for hypersonic problems.^{9,10} Global adjoint-based refinement has also been shown to be effective for a variety of flow conditions.^{11,12,13} Local truncation error and residual estimates are also common.^{1,14} In future work, we intend to more fully investigate some of these methods and their application for problems of interest.

This paper employs an undivided difference of user-specified flow variables (ρ, u, v, w, p, T) to indicate if there is a sufficiently large delta between adjacent cells to merit subdivision. Differences indicate that there is a variation or discontinuity in the flow that requires additional spatial resolution. In this work, differences are normalized based on cell-centered quantities.

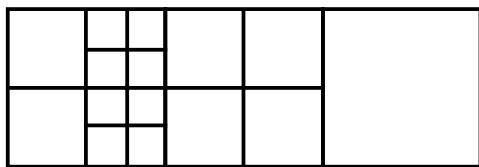
Equation 1 shows the criteria used for refinement of a flow variable, ϕ . Each variable may have a separate tolerance, ϕ_{tol} . The code loops over all faces and evaluates if the tolerance is exceeded for any targeted variable. If it is, then both neighbors are flagged for refinement. The maximum grid level is specified by the user and cells that are already at the maximum level are still flagged for refinement, but no subdivision is actually performed. This ensures that for physical discontinuities that would otherwise create an infinite number of grid levels, grid sizes remain bounded to a specified resolution.

$$\phi_{tol} < \frac{|\phi_i - \phi_{ii}|}{\min(\phi_i, \phi_{ii})} \quad (1)$$

With a face-based unstructured numerical method, neighboring cells may be subdivided an arbitrary number of times relative to the neighbor cell. Figure 4(a) shows an example of a cell (far-left) that contains subcells two levels higher than the neighbor cell (the center cell). The center cell (in 2-D), now has seven faces defining its perimeter. Such an arrangement would not impact the finite volume formulation, but the current method seeks to avoid dramatic changes in cell sizes. To accomplish this, the code enforces that adjacent cells not have more than one grid level difference between them. The accompanying figure, Fig. 4(b), presents an acceptable configuration.



(a) Rejected refinement of face neighbors.



(b) Accepted refinement of face neighbors.

Figure 4. Computational meshes with unacceptable and acceptable refinement of adjacent cells.

Also included in this work is a concept of ‘buffer cells’. Cells adjacent to those flagged for refinement are also refined. This creates a region of refined cells that allow flow features to propagate over several time steps without drifting into a coarser region of the grid. By conservatively choosing the size of this buffer region, the researcher can confidently reduce the frequency at which grid quality is assessed. This will further reduce the overhead associated with AMR, but comes at the cost of additional refinement.

When coarsening cells, the child cells are not removed from memory, but are marked as ‘blanked’ and unused until they are ‘unblanked’ by subsequent refinement of the parent cell. This reduces the computational requirement since unblanking is less expensive than recreating the geometry and connectivity, but it does incur an additional memory requirement. The criterion for coarsening is simple: to coarsen the grid and unblank a parent cell, all children of the parent must be active and cannot be flagged to refine or maintain

its current grid level. If any one child cell is flagged to be refined by Eq. 1, the requested number of buffer cells, or to ensure an acceptable transition between face neighbors (see Fig. 4), then no coarsening is allowed. Before the cell is actually removed, the restricted solution value on the parent cell is compared to all of its face neighbors. If the new value would not trigger refinement of the parent cell, then it is flagged to be coarsened.

IV. High-Order Flux Verification

Two distinct flux methods are examined below in order to determine if the AMR method causes a reduction in their accuracy. Each of the methods are briefly described with their dependencies identified. The methods fall into two categories: Modified Steger-Warming fluxes and Kinetic Energy Consistent fluxes.

The high-order implementations require gradients of flow quantities. To calculate these gradients, the code uses a weighted-least squares approach using all cells that share a face with a given cell. The solution to the least squares calculation involves a matrix solve with the right-hand side updated every timestep. Fortunately, the left-hand matrix is only a function of the geometry and can be inverted once and stored. This matrix does need to be re-inverted if the mesh changes due to refinement or coarsening, however.

Another requirement for many of the flux methods is knowledge of high-order partners to each face. These high-order partners are the second-neighbors to the face in question. For sections of hexahedral mesh without hanging nodes, there exists a unique set of neighbors and high-order partners for each face. Figure 5(a) shows the face neighbors (*i* and *ii*) and the high-order partners (*ih* and *iih*) to the face highlighted in red.

Grids with hanging nodes may present obvious selections for the high-order partners. Suppose we subdivide the cells on the left-hand side of the previous case and generated a new set of faces, as shown in Fig. 5(b). The new, highlighted face still has a unique and easily determined set of neighbors. For this situation, the high-order partners are also unique and very little changes from the uniform grid case. Unfortunately, not all configurations provide ideal circumstances for high-order partner selection.

Some ambiguity presents itself with more exotic arrangements of non-conformal cells. For the situation shown in Fig. 5(c), the cell on the far-left has also been subdivided. The selection of neighbors remains unchanged and so does the right-hand high-order partner (*iih*). One possible choice for the left-hand high-order partner (*ih*) is to select the new cell that is closest and in the face-normal direction, as shown in the figure. Other selections for the partner *ih* exist. One option would be to include both of the new cells that share a face with the coarse neighbor, cell *i*.

Figure 5(d) presents a third way to handle this situation. In this description, a restriction operator averages values from all children of the original coarse cell *ih*. Those values are then used as the properties for the high-order partner.

It is the last approach, Fig. 5(d), that is used in this work when such ambiguity arises. The connectivity and search algorithm is simpler and the mechanisms for restriction are already available in the solver. Finally, due to previously mentioned requirement for at least one buffer cell surrounding cells flagged in the refinement criteria, we ensure that a discontinuity or large gradient does not exist in the subcells of cell *ih*.

To calculate the gradient in this new cell, we use a restriction of the gradients in the child cells. A more exact gradient might be determined by using a least-squares solve based on the values in the child cells. There is potential for a more rigorous investigation into best practices for the construction of numerical stencils on grids with hanging nodes. However, we will show that our selection does not appear to have a deleterious effect on solution quality.

A brief survey of the flux methods examined in the following section is provided below:

Modified Steger-Warming (MSW) ($\mathcal{O}(\Delta x)$) This upwinded flux depends on the flow quantities in the cell on either side of a face. To calculate the flux, the split Jacobian is calculated using average values at the face. No gradients or high-order partners are required.^{15,16}

Modified Steger-Warming (MSW) ($\mathcal{O}(\Delta x^2)$) This is a second-order implementation of the MSW method that uses a linear reconstruction to calculate the quantities at the face. No gradients are required for the linear projection, but high-order partners are required.

Central Kinetic Energy Consistent (KEC) ($\mathcal{O}(\Delta x^2)$) This central flux depends only on the flow quantities in the neighboring cells. First-order dissipative fluxes are added in regions with sharp gradients

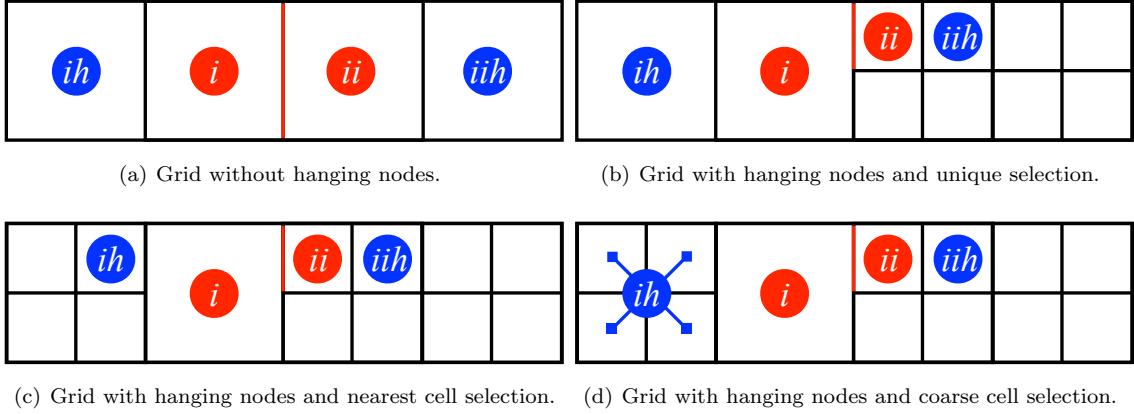


Figure 5. Illustrations of high-order partner selection for various grid topologies.

as identified by a user-defined switch. The flux evaluation does not depend on gradients or high-order partners, but some dissipation switches do require them.^{17,18}

Central Kinetic Energy Consistent (KEC) ($\mathcal{O}(\Delta x^4)$) A central stencil that requires the gradients in the neighboring cells improves the order of the KEC method. This method depends on the scalar quantities and the gradients at the neighboring faces but not the high-order partners.

Central Kinetic Energy Consistent (KEC) ($\mathcal{O}(\Delta x^6)$) Using the gradients in the high-order partners in addition to those in the neighbors, a sixth-order formulation of the KEC fluxes is possible. As with the other two KEC methods, in regions with discontinuities first-order upwind-dinded dissipation terms are required to maintain stability.

A. One-Dimensional Gaussian Pulse

To assess the effects of AMR on these numerics, a study into the order of the error was performed by comparing solutions on uniformly refined grids to those computed on a set of elements generated using AMR. Low-dissipation methods are ideal for flows without discontinuities. Taking this into account, convection of a Gaussian density pulse is considered. This is an identical test case to the one used in previous investigation into the KEC numerics.¹⁸

The initial conditions for the 1-D density pulse are shown in Eq. 2 with \bar{x} being the x -coordinate of the computational element. Simulations were performed on three-dimensional domain with a width of a single element in the y - and z -directions and a varying number of cells in the x -direction. Bounds in the x -direction extended from -5 to 5. The boundary conditions enforced zero flux on the y - and z -aligned faces and periodicity in the x -direction.

$$\begin{array}{lll} u = 1.0 & w = 0.0 & p = 1.0 \\ v = 0.0 & \rho = 1.0 + \frac{1}{10}e^{-\frac{(\bar{x}-5.0)^2}{2}} & T = \frac{p}{\rho R} \end{array} \quad (2)$$

In an ideal, inviscid simulation, the initial density pulse will advect with the freestream velocity. Error is easy to assess by comparing the density in the simulation after one complete cycle with the initial density profile. The RMS of the error (weighted and normalized by volume) across all computational cells provides a scalar result for the accuracy of the numerical flux on a specific grid. To minimize the effect of any discretization error due to the timestep, all simulations use a timestep consistent with a CFL of 0.1 and a 3rd-order explicit Runge-Kutta method.

Figure 6 presents the error as measured using a range of numerical fluxes across several grids with uniform cell distributions. Also plotted are lines showing theoretical convergence of first-, second-, fourth-, and sixth-order. As expected, these methods on uniform grids recover the predicted order as the grids are refined.

For the comparative study using AMR, the baseline grid uses only 10 cells across the computational domain. This grid is then adapted to the maximum number of grid levels specified. To emulate the results

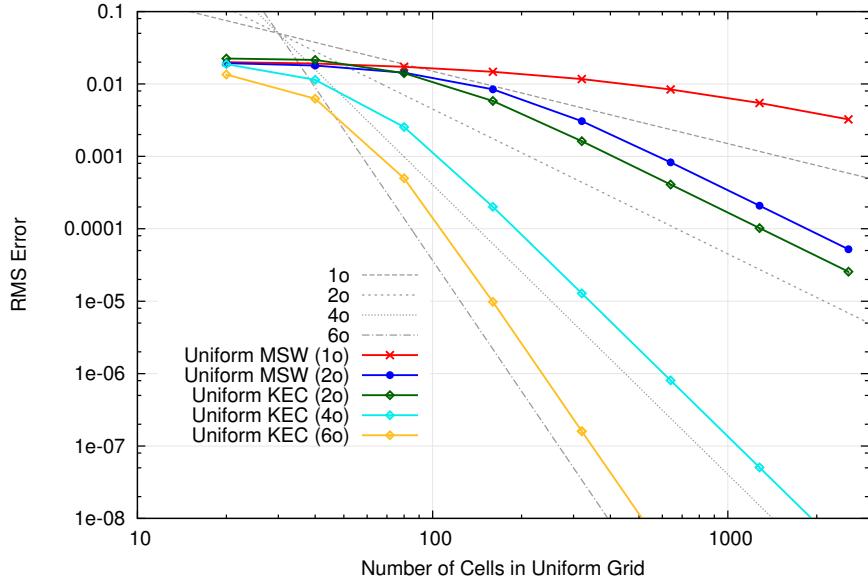


Figure 6. RMS error for 1-D density pulse after one cycle using uniform meshes.

for the uniform case with 320 cells, for example, 5 grid levels are required ($10 \cdot 2^5 = 320$). The solutions are initialized on grids refined to capture the gradients in the prescribed density profile (Eq. 2). Furthermore, at a CFL of 0.1, AMR iterations are performed every 5 solver timesteps. The refinement is based on local variations in density governed by a tolerance value, ρ_{tol} (see Eq. 1).

Figure 7 shows the mesh used in an AMR simulation at the initial time, half-way through the simulation, and at the final time. Both the refining and coarsening operations perform well for this problem. The adapting grid tracks the pulse in time and fine cells are removed from the domain when no longer needed. It uses 5 levels of refinement on a baseline grid of 10 cells. The accompanying contours of density show the solution obtained using the sixth-order accurate KEC fluxes.

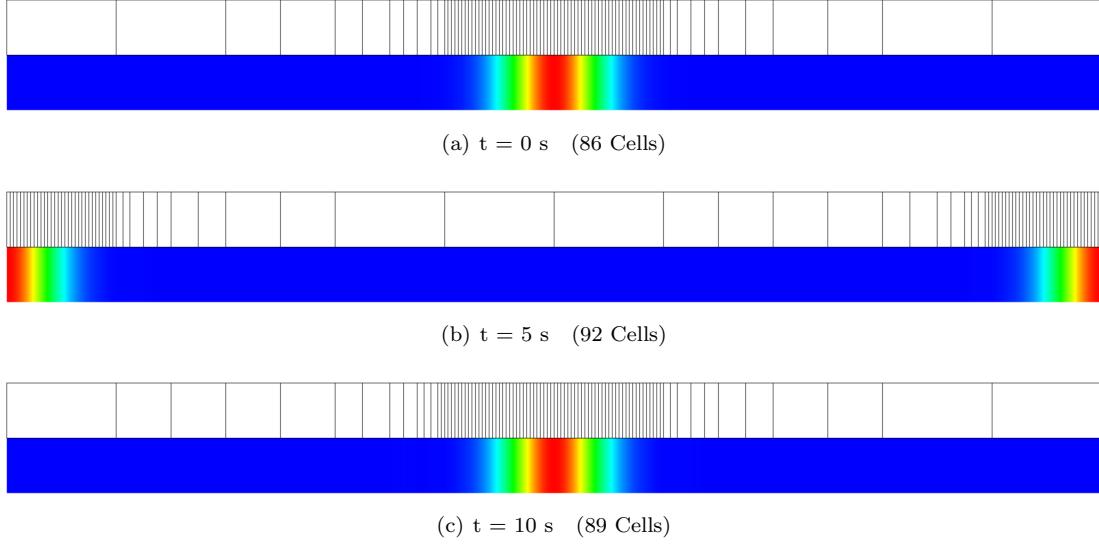


Figure 7. Computational mesh, ρ contour, and cell count for 1-D Gaussian pulse at three simulation times using AMR with a refinement tolerance of $\rho_{tol}=1E-3$.

Results for a spatial convergence study using AMR are shown in Fig. 8. The initial conditions and error metrics are identical to the one shown previously for the uniform grid. Each successive level of refinement on the 10 cell baseline grid increases the ‘effective number of points’. For instance, one level of refinement

creates fine cells identical to those found in a uniform grid of 20 cells. The error for that case is plotted on Fig. 8 at an x -axis location of 20 (corresponding to the number of cells in a uniform grid required to get an identical fine-cell resolution).

Two different values of ρ_{tol} are used in order to illustrate sensitivity to the AMR refinement parameter. The first, shown in Fig. 8(a), uses a value of $\rho_{tol} = 1.0E-6$. It illustrates that the grids with adapted meshes achieve the prescribed spatial order in the flux calculations, but only a minimum error is attainable. The second plot, Fig. 8(b), uses a much lower value for ρ_{tol} : $\rho_{tol} = 1.0E-8$. It is able to reduce the error further and matches the results shown previously with the uniform grids.

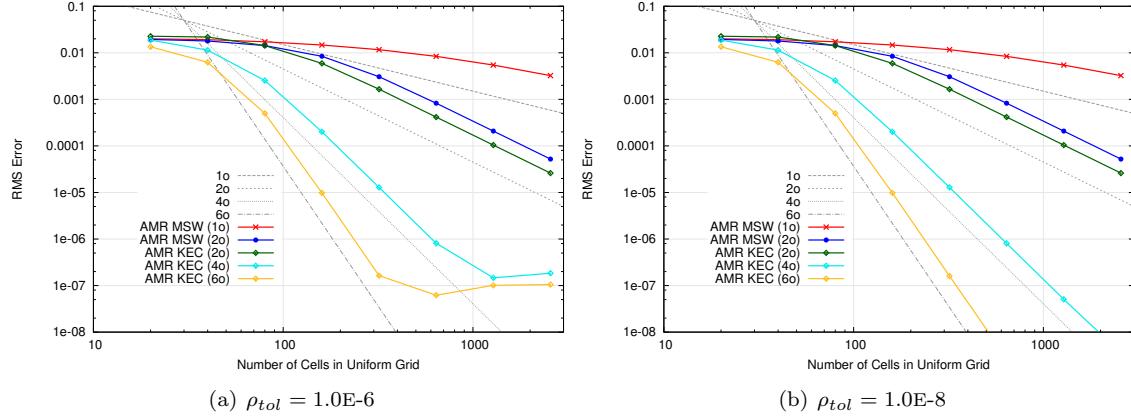


Figure 8. RMS error for 1-D density pulse after one cycle using AMR and two different values of ρ_{tol} .

The agreement between Fig. 8(b) and Fig. 6 indicates that the AMR methods presented here are able to preserve the high-order, low-dissipation numerics important to the authors. The caveat is that the choice of the refinement tolerance parameter plays a critical role in determining the minimum attainable error. This is intuitive; the method cannot achieve an accuracy that is much beyond the prescribed threshold for refinement. It is important to note that the tolerance required for a simulation may not be known a priori and introduces a sensitivity which must be assessed. This is an aspect of the formulation that requires further investigation.

B. Two-Dimensional Gaussian Pulse

There are additional degrees of freedom introduced when moving beyond the one-dimensional problem. The choice for the second neighbors to a face become non-unique and in order to ensure that the approach chosen here does not adversely impact the accuracy of the methods, the order of error assessment is repeated for a two-dimensional case. In addition, since the motivating problems are inherently two- or three-dimensional, it is necessary to ensure that their accuracy does not suffer as result of the AMR procedure.

This case uses a two-dimensional Gaussian pulse convecting for one cycle on a periodic grid (in the x - and y -direction) with no fluxes in the z -direction. The initial conditions for the 2-D density pulse are identical to those for the 1-D simulation (Eq. 2), but with $\bar{x} = \sqrt{x^2 + y^2}$. The grid is comprised of a varying number of cells, uniform in size and number in the x - and y -directions. Again, a CFL of 0.1 is used for all computations and the volume-weighted RMS error is compared.

Similar to the 1-D case considered earlier, Fig. 9 shows the RMS error for simulations using a uniformly distributed number of cells. The x -axis shows number of cells in the x -direction (identical in the y -direction). As with the one-dimensional case, the numerical fluxes perform to their analytical order of accuracy as the grid is refined.

Figure 10 shows the mesh used in an AMR simulation at the initial time, half-way through the simulation, and at the final time. As with the previous case, 5 levels of refinement are allowed. Only half the grid is shown - cell counts are for the entire mesh, as compared to 102,400 (320×320) for a uniformly refined grid.

The comparative cases using AMR on a coarse grid with 10 cells in the x - and y -directions (100 total elements). A maximum of six grid levels are considered, corresponding to an effective grid with 409,600 elements (640×640). Figure 11 shows the results for two values of ρ_{tol} . Again, the larger tolerance ($\rho_{tol} = 1.0E-6$) results in a minimum error in excess of the desired result seen in the uniform grid case. With a ρ_{tol}

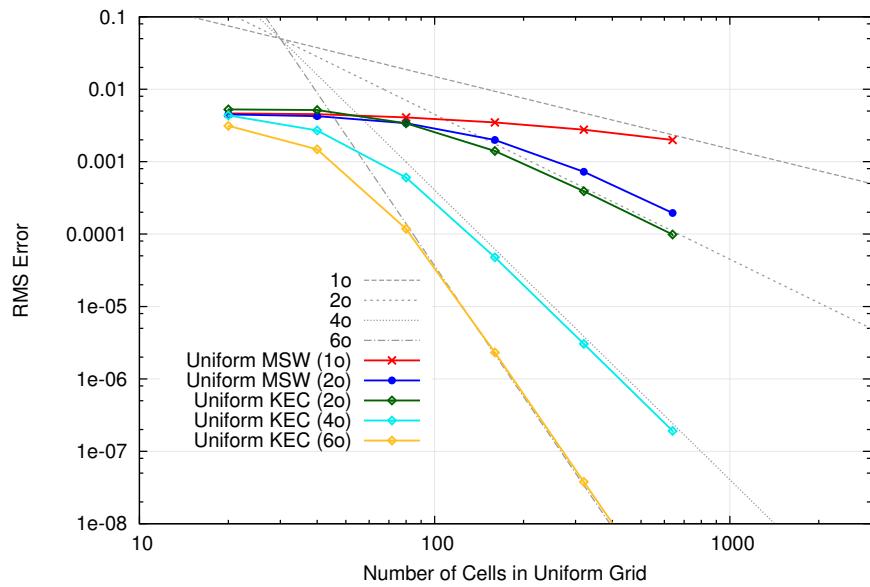


Figure 9. RMS error for 2-D density pulse after one cycle using uniform meshes.

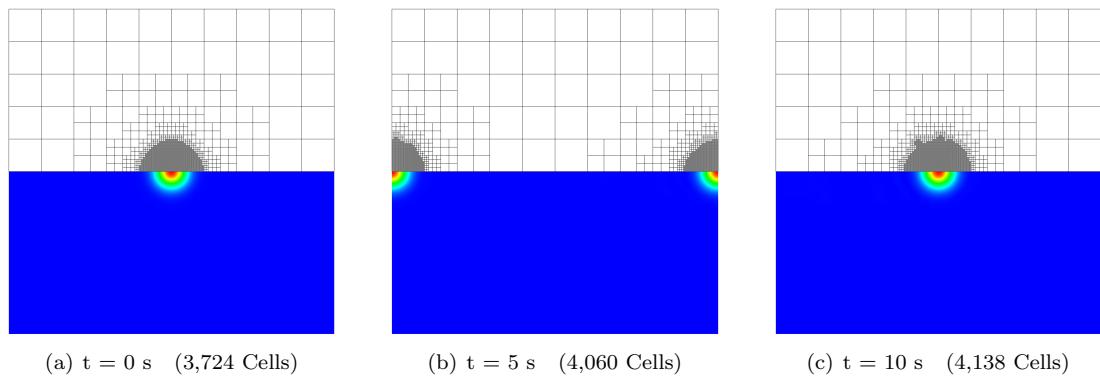


Figure 10. Computational mesh, ρ contour, and cell count for 2-D Gaussian pulse at three simulation times using AMR with a refinement tolerance of $\rho_{tol}=1E-3$.

of $1.0\text{E-}8$, the analytical order of error is recovered for the range of grid sizes examined. As seen with the one-dimensional case, with an appropriate value for the refinement tolerance the AMR method is amenable to these high-order fluxes.

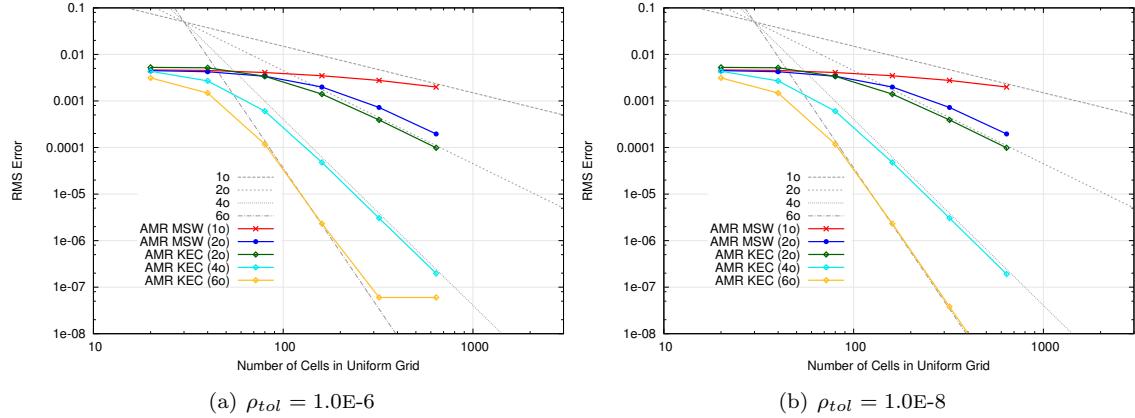


Figure 11. RMS error for 2-D density pulse after one cycle using AMR and two different values of ρ_{tol} .

While promising, these results still leave some ambiguity at the appropriate value for the refinement tolerance necessary for a practical problem. To achieve a specified level of accuracy for this test problem, it seems best to select a value of ρ_{tol} that is at or below the desired RMS of the error. The same guideline might prove useful for non-linear flowfield as well, but it is likely that more complicated problems will require more involved heuristics to drive refinement.

A more laborious alternative is to perform a sensitivity problem for each specific class of problem. Currently, engineers and researchers perform grid sensitivity studies in order to guide grid generation. As implemented in this work, our process requires two such studies: one to determine the tolerance, ϕ_{tol} , and another to determine the required minimum cell size. For this reason, more focused work is still required into more advanced metrics for refinement.

V. Implicit Time Integration

Explicit numerical methods are limited in the maximum allowable stable timestep. This timestep is limited by the grid spacing. For grids with a very fine cell spacing or simulations where a large timestep is desirable, implicit time integration methods are advantageous. The stringent requirements on grid spacing near the wall usually dominate the determination of the maximum stable timestep for hypersonic applications. This work does not include viscous fluxes and as such does not have onerous demands on the near-wall grid and a reasonable explicit time-step is still possible. Even so, for steady-state computations where temporal error does not influence the result, it is efficient to iterate using a timestep much larger than the maximum stable explicit value.

The explicit time integration presented earlier can be made implicit by evaluating all of the numerical fluxes at the future time level, $n+1$, instead of the current time level, n . To do this, a first-order linearization is performed:

$$\begin{aligned} F^{n+1} &= F^n + \frac{\partial \bar{F}^n}{\partial \bar{U}} (\bar{U}^{n+1} - \bar{U}^n) \\ &= F^n + A_+^n (\bar{U}_i^{n+1} - \bar{U}_i^n) + A_-^n (\bar{U}_{ii}^{n+1} - \bar{U}_{ii}^n) \\ &= F^n + A_+^n \delta \bar{U}_i + A_-^n \delta \bar{U}_{ii} \end{aligned}$$

where A_+^n and A_-^n are the right- and left-running flux Jacobians at the face. $\delta \bar{U}_i$ and $\delta \bar{U}_{ii}$ are the implicit updates to the conserved variables with i being the index of the current cell and ii the index of the face neighbor. The linear system that results from combining this linearized flux with the governing finite volume formulation is:

$$\delta\bar{U}_i + \frac{\Delta t}{V_i} \sum_{faces} [(A_+^n \delta\bar{U}_i + A_-^n \delta\bar{U}_{ii}) \cdot \hat{n} S] = \Delta\bar{U}_i$$

By storing the Jacobian matrices and performing an iterative solution procedure, the values $\delta\bar{U}_i$ can be obtained.

Primarily for unsteady problems, it is undesirable to bias the solution of this linear system. Similarly, flows without strong coupling in a preferred direction should not be handled with numerics that artificially induce correlations. Examples of these flows include mixing in jet plumes, shear-layer instabilities and wake dominated flows, as well as direct simulations of turbulence.

To obtain a solution to the implicit system, a Full-Matrix Point-Implicit method is used.¹⁹ This block-diagonal implicit solver maintains tight coupling for all degrees of freedom within a computational cell. Furthermore, since the method has no bias in how the implicit linear system of equations is relaxed to convergence, it is suitable for unsteady, time-accurate simulations.

Implementation of this implicit method within an AMR framework is natural and straightforward due to the unstructured nature of the approach. Connectivity between cells in grids generated using AMR is arbitrary. Employing a more sophisticated implicit solver would require searches to recreate appropriate adjacency structures following each invocation of the AMR process. The possibility of using a more advanced formulation of the implicit operator, perhaps using multigrid or coupled line solves, is reserved for future work.

VI. Double Wedge Simulations

An important phenomenon of interest in the field of hypersonic aerodynamics is the shock-shock interaction. Common experimental and computational geometries used to investigate the effects of this interaction are the double wedge and the axisymmetric double cone. By varying the freestream conditions and the two wedge/cone angles, an entire family of interaction types can be realized (as classified by Edney).²⁰ The double wedge or double cone provide a simple test case for comparison between computational methods and a means for computational validation.

Double wedge geometries provide a tractable two-dimensional problem with which to assess the AMR procedures presented in this paper for a hypersonic application. Several of the resulting interactions are steady-state and enable aggressive implicit time-stepping. Additionally, the domains are easily discretized and refinement on the boundaries requires only linear subdivision of edges. Since smoothing and surface projection are not currently implemented these shapes provide an ideal test problem for our purposes.

Researchers have performed implicit computational simulations of double wedges using methods similar to those described in this work.²¹ These results provide a guide for expected behavior and aid in selection of an appropriate test case. Several shock-shock interactions create relatively large portions of subsonic flow with unsteady flow characteristics. Based on these results, a double wedge with cone half angles of 15° and 35° is selected as a steady interaction for analysis here. At a freestream condition of Mach 9.0 and $\gamma = 1.4$ this geometry produces a Type VI shock-shock interaction.

Shock-shock interactions can generate a highly-complex arrangement of flow structures. For this method of adaptive refinement, this provides a test case exhibiting characteristics found in real-world hypersonic simulations. Solutions obtained by using AMR are compared to those obtained using uniform grids at identical conditions. Figure 12 shows a schematic of the important flow features in an interaction of the type examined below.

A. Grid Generation

Work by Olejniczak et al. showed that a two-dimensional grid of 1024×1024 elements was necessary to capture the smallest-scale features in the double wedge simulations.²¹ To better estimate grid convergence and allow for comparison on higher density grids, a 2048×2048 grid was generated using a similar approach. This grid was then coarsened evenly in each direction to create successively smaller domains with the minimum grid being 32×32 . This minimum grid was then refined using AMR to recreate a resolution identical to the fine grids.

One consideration that should be mentioned is grid cell alignment. The finest grid (2048×2048) was smoothed using an elliptic smoother. With the coarser representations of the grid, the AMR routine does

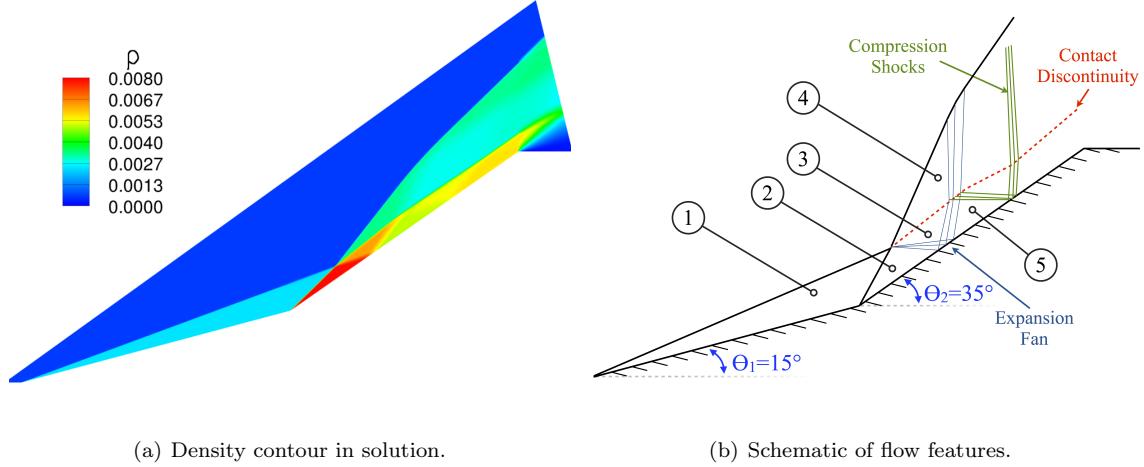


Figure 12. Density contour and feature schematic for 15° - 35° type VI shock-shock interaction.

not perform any additional smoothing steps and simply subdivides the faces. For this reason, the 2048×2048 grid generated by AMR is not as smooth as the fine grid from the grid generation code. The grid lines are not necessarily normal to the boundary near regions of high curvature (in the 15° - 35° corner). Finally, the original grid used geometric stretching in the cells near the wall in order to cluster points to the boundaries. The AMR method does not include this clustering.

As will be seen below, there is a noticeable difference between the solutions due to these effects. It is only apparent in regions of the flow where the solution is already sensitive to misalignment and where the original grid was highly skewed due to the smoothing operations. Figure 13 shows a close-up of the grid near the 15° - 35° corner for a uniform mesh of 512×512 points and an initially coarse mesh refined to the same global cell density. Most noticeable is the difference in wall spacing on the 35° ramp.

The impact of these differences in grid quality is not explicitly examined in the following presentation of the results. Subsequent discussion identifies significantly altered pressure predictions between the uniform grid and the grids generated using AMR specifically in this portion of the geometry. Additional simulations (not shown) were performed by uniformly refining a grid using AMR in order to remove the possibility of hanging nodes causing these differences. Those simulations agreed with the AMR solutions with hanging nodes, strongly implicating these grid quality issues in the observed discrepancies in region 2.

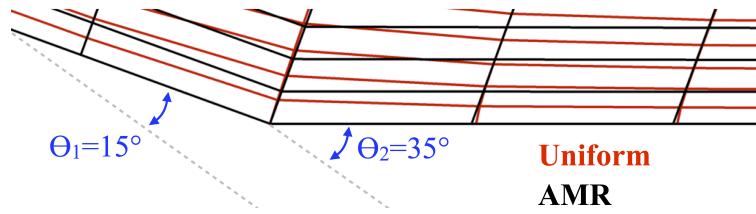


Figure 13. View of cells for 512×512 uniform grid and coarse grid refined to 512×512 with AMR.

B. Results

A simple measure for accuracy is the surface pressure on the double wedge. For the attached, inviscid case, it is possible to determine several of these pressures analytically. Referring back to Fig. 12, the pressure in region 1 is easily attainable from the oblique shock relations. Likewise, region 2 is derived from the conditions at region 1 and the oblique shock relations. For regions 3 and 4, the derivation is a bit more complex. The relationship between regions 2 and 3 is governed by isentropic expansion (Prandtl-Meyer). Region 4 can be described by the freestream conditions and the oblique shock relations; provided the flow direction in regions 3/4 is known. Based on these relations, a single solution exists that maintains $p_3 = p_4$ as required. Finally,

the pressure in region 5 is obtained by isentropic expansion from region 3. Table 1 shows conditions for the regions shown in Fig. 12(b).

Region	1	2	3	4	5
Mach	5.04	3.04	3.25	1.73	3.48
$\frac{p}{p_\infty}$	11.24	79.95	58.61	58.61	42.14

Table 1. Conditions for regions in double wedge show in Fig. 12 with Mach 9, $\gamma = 1.4$.

For a steady-state problem like this one, there is no requirement to track features as they migrate from areas of higher cell density to coarser regions. With unsteady calculations, the timestep should be reasonable when considering the convective (or acoustic) speeds. The solutions presented below were run at a timestep that was 40-times greater than the largest stable explicit timestep.

Figure 14 shows a grid convergence study using uniform grids. The figure shows computational results for pressure relative to the freestream value (p_∞) on the surface of the double wedge. Also shown on the figure are the analytical predictions for the pressure ratio in regions 1, 2, and 5 (gray lines). Insets highlight the pressures in regions 2 (upper-left) and 5 (lower-right).

In general, the pressure comparisons are good when compared to the theory. Region 2 shows a non-physical jump in pressure at the shock and a great deal of ringing in the solution downstream. At the impingement of the expansion fans between regions 2 and 5, the ringing disappears and the solution looks very similar to what the theory predicts. Grid convergence is not entirely realized with this series of grids: the specifics in region 2 are becoming a better match to the theory and the width of the expansion fan and compression shocks are tightening. All grid levels capture the gross flow features.

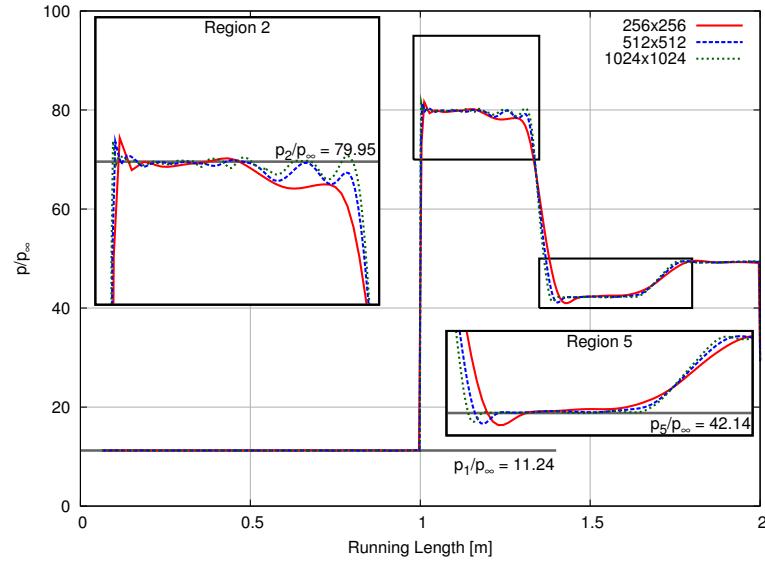


Figure 14. Surface pressure on the wedge for a series of uniform grids with varying grid densities.

Shown below are the results for a similar set of cases including several that were too expensive to create with uniform grids. The AMR procedure was performed once the solution was considered converged (residual magnitude had dropped five orders of magnitude from its initial value). Cells identified for refinement using a joint local tolerance of $\rho_{tol} = 1E-2$ and $p_{tol} = 1E-2$ were refined once. This was continued until the solution was allowed to converge on the finest allowable grid. A progression of grid levels is shown in Fig. 15 with inset images providing detail near the corner.

Figure 16 shows the resulting surface pressures. The values are very similar to those seen previously. In fact, when plotted together, the only differences that exist when using identical effective resolutions are the pressures in region 2. These results illustrate that grid convergence may be obtained near a resolution of

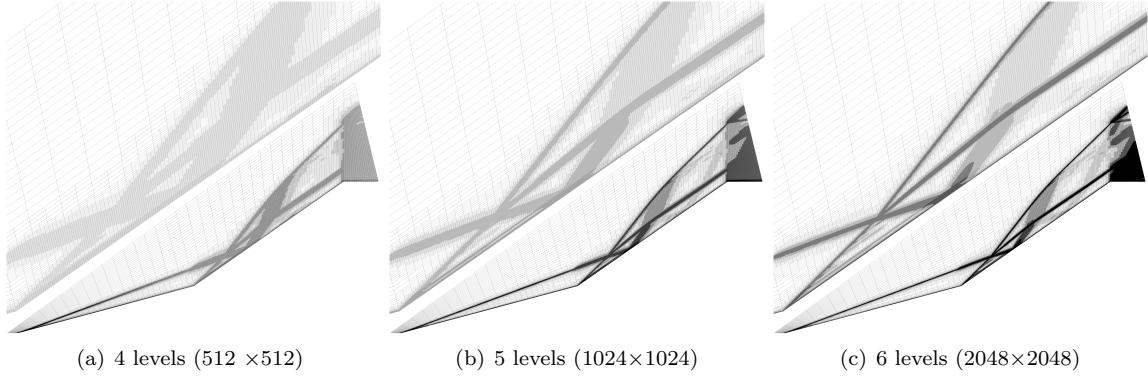


Figure 15. Images of the mesh for varying levels of refinement with their effective resolutions.

4096×4096. This is higher than was estimated earlier based on the work by Olejniczak et al. using similar numerics.²¹

Solutions on grids obtained using AMR have a more pronounced peak and present a different pattern of oscillations prior to the impingement of the expansion fan. As was mentioned earlier, this is attributed to the difference in the grids based on simple cell subdivision (see Fig. 13). With refinement in the AMR, the oscillations in region 2 become less pronounced and the pressure peak becomes more localized, even though its magnitude remains unchanged. Downstream of the large pressure peak, the cases with AMR show a more accurate pressure profile in region 2 with reduced oscillations as compared to the uniform grid cases.

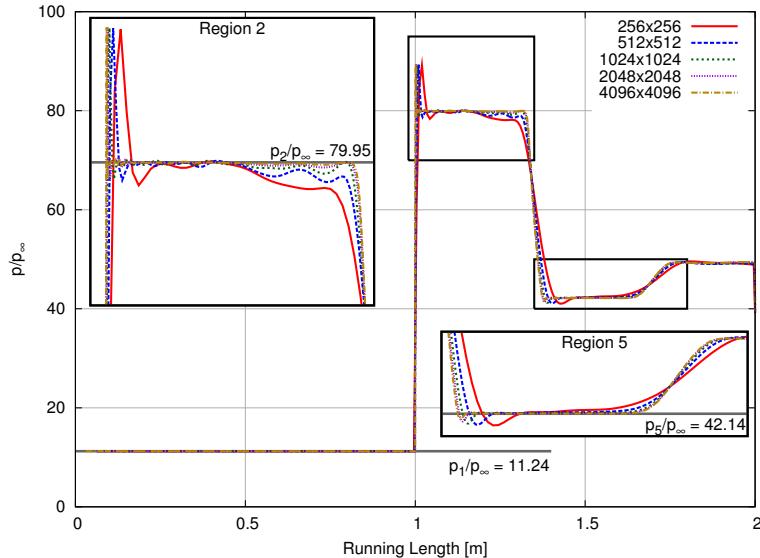


Figure 16. Surface pressure on the wedge for a series of grids with varying levels of AMR.

Plots showing the residual for the cases examined previously are shown in Fig. 17. These plots are versus simulated time. For this test case, one flow time (the time it takes for a fluid element on the upstream portion of the wedge to travel to the most downstream point) is slightly less than 0.0016 seconds. We expect a supersonic flowfield to converge in roughly a flow time due to the hyperbolic character of the Euler equations. Focusing on Fig. 17(a) for the uniform grid case, we recover this expected behavior regardless of the grid resolution.

For the grids created with AMR, Fig. 17(b) presents a slightly different behavior. The residual behaves (on the coarsest grid) similar to the uniform cases until one flow time. At this point, there is a sharp jump

in residual (caused by invocation of AMR). Each subsequent call to the AMR subroutine causes a similar jump in solution residual. The refined grids show a similar trend as they converge over the subsequent flow time. What is interesting is that on the finer grids, it does not require a complete flow time to converge to the refinement threshold. Observations show that the error caused by refinement in the shock near the tip of the geometry dissipates as it passes through coarse cells in region 1. In effect, this reduces the flow time for these simulations and accelerates convergence.

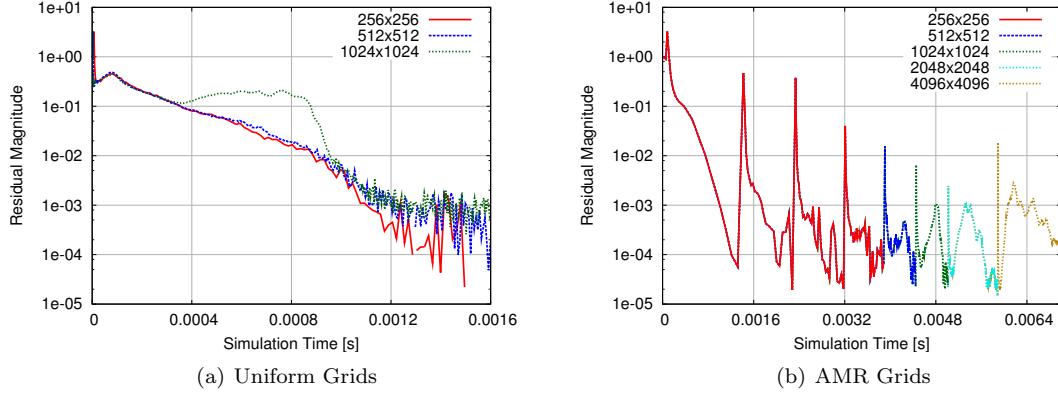


Figure 17. Plots of residual for solutions obtained with uniform grids and those refined with AMR.

C. Computational Efficiency

For the cases considered here, the solutions performed using AMR are much less expensive than those converged on uniform grids and achieve nearly identical results. To quantify this savings and investigate sensitivities, several additional cases were run. In addition to refinement upon residual convergence, our code includes an optional input for a refinement frequency. When included, AMR is performed every n timesteps regardless of the residual - n being the interval chosen.

To understand how the method performs for these cases, a number of refinement intervals were considered for three different values for maximum refinement. Figure 18 shows the results as compared to the uniform grid case. The percentage of both cell count and runtime are plotted. The cost of the AMR process is less than 1 flow solve iteration, and it is expected that additional AMR iterations will not cause significant computational cost. For this reason, there should be minimal change in the performance for different values of n .

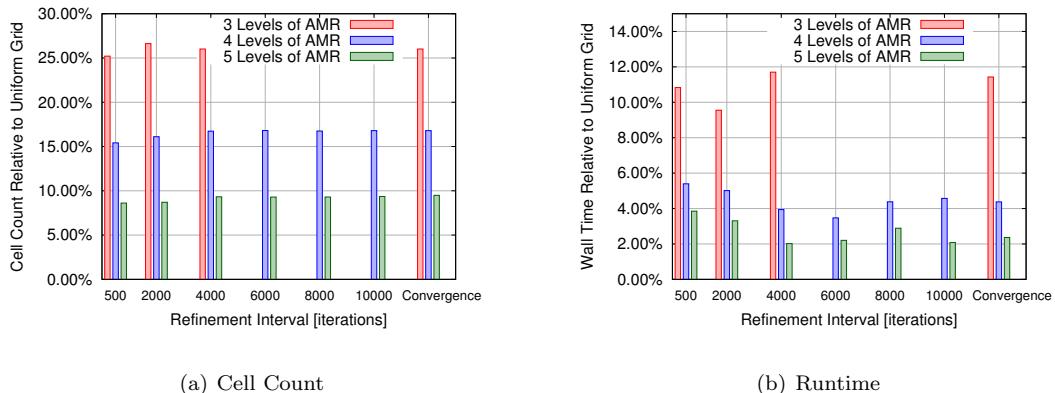


Figure 18. Percentage of cells and wall time used by AMR simulations compared to uniform grids. Several values of refinement frequency considered.

Examining the figure, it is clear that for a given refinement level, the refinement frequency does not have a dramatic effect on either the cell count or the time-to-solution. Figure 18(a) shows the comparison for cell count across many refinement intervals and for the case that was refined only at residual convergence. In general, the more frequent the AMR (lower value of n), the lower the number of cells in the final mesh. Fewer cells are required because more cells are coarsened once the flowfield is better resolved. Cases with larger refinement intervals have much fewer opportunities to remove cells that were added in previous refinement iterations.

Even with the slight trend, the cell counts are mostly independent of the refinement interval chosen. Compared to the uniform grid case, three levels of refinement require roughly 26% of the cells. The improvement is more dramatic for four and five mesh levels - 17% and 10%, respectively. For a flow with well-defined, localized features, additional levels of AMR prove very effective.

Figure 18(b) highlights the savings in computational time for this model problem relative to the uniform grid solutions. It might be expected that the cost for these simulations be directly proportional to the relative mesh sizes since most operations scale with the number of cells or number of faces. This turns out to not be the case. As mentioned earlier when looking at the residual, the effective flow time is reduced on the grids using AMR due to dissipation in region 1.

In general, the cases with the more frequent refinement (lower refinement interval) require slightly greater walltime. This is caused by the initial AMR process adapting to a solution prematurely - before the flow solver adequately resolves features on the coarser grids. Looking across all cases, the cost savings was still considerable. For three levels of refinement, the AMR solutions were about 11% of the wall time as compared to the uniform grid case. Four and five levels of AMR provided more considerable benefit - 5% and 2-3%, respectively.

These steady wedge flows require few calls to the AMR subroutine since features are stationary and refinement only at convergence is convenient. For more complicated and viscous flows, features may develop slower or be coupled to the resolution of the finest grid cells. A viscous double cone incorporates a separation bubble at the corner between the two cone angles. This separation bubble has a profound effect on the shock structure and produces a region of subsonic flow. Previous work has shown that this effect is very dependent on the resolution in the corner.²²

It might be advantageous to call the AMR subroutine several times throughout the development as the separation region changes size and the shocks resulting from the interactions move. Fortunately, each AMR call is inexpensive, so provided it was not called *every* timestep it should still yield considerable improvement in time to solution. Further studies are planned for computational efficiency on pseudo-steady and unsteady flows.

VII. Conclusions

The implicit unstructured finite-volume method is compatible with AMR and grids containing hanging-nodes. Results from this work indicate that AMR shows promise for high-order methods and sensitive, unsteady problems. With proper selection of refinement criteria, accuracy previously demonstrated on uniform grids is realized on grids containing hanging-nodes. This work shows that low-dissipation, KEC flux functions developed for hexahedral cells can be successfully applied to the more exotic grids explored here.

Our model problem of a shock-dominated flow also demonstrates comparable accuracy at a much reduced computational cost. Presented was an application of Full-Matrix Point-Implicit method to unstructured meshes with hanging-nodes. Implicit time stepping enables a much larger stable timestep and reduces the cost for many problems. Due to the large aspect ratios, implicit methods are necessary when working with grids designed for viscous solutions. It is possible that future investigation will indicate that a more advanced implicit method is required for improved performance.

As has been mentioned earlier, AMR has a long history and proven success on a large range of problems. It is an active research topic with a range of approaches currently being pursued. Our approach is intended to progress towards a completely unstructured framework that enables a flexible AMR process for hypersonic applications. There are several important avenues that are still being pursued in order to increase capabilities, enable more interesting problems, and assess applicability to a broader range of problems. Several major improvements are described in the paragraphs that follow.

Near-term goals include investigation of the Navier-Stokes viscous fluxes. These fluxes are currently implemented in the code and appear to perform well for grids with hanging nodes. The viscous terms are

necessary in order to begin validating against experimental results and to simulate real-world conditions. Inclusion of the numerics necessary for multi-species, non-reacting flow is underway.

Further work is required in identifying an ideal refinement parameter that does not require arduous user input. The strategy employed here requires appropriate selection of a refinement tolerance as well as a maximum grid refinement level. Selection of these parameters is not always intuitive and requires sensitivity studies to ensure efficient operation.

The work shown in this paper was performed entirely in serial. Parallelization is required for larger problems and 3-D cases will require additional computational power to achieve converged solutions in a reasonable amount of time. Dynamic partitioning is required since load balancing based on the coarse grid will certainly prove non-ideal for the final, refined mesh. This might require a significant change to data management as well.

Grid smoothing and surface projection are also necessary to move beyond the simple shapes examined in this work. Most aerodynamic shapes include non-linear surfaces and simple subdivision of boundary faces is insufficient to properly resolve the flow features. For viscous cases with very small body-normal spacings near the wall, smoothing is required in order to ensure that no negative volumes develop and to help ensure that grid cells remain aligned with the surface. Development of a robust method to project and smooth unstructured grids with hanging nodes is underway.

Acknowledgments

This work was sponsored in part by the NASA Johnson Space Center Academic Fellowship and the Defense National Security Science and Engineering Faculty Fellowship. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NASA, Johnson Space Center, or the U.S. Government.

References

- ¹M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, pp. 484–512, 1984.
- ²M. J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, vol. 84, pp. 64–84, 1989.
- ³C. P. T. Groth and S. A. Northrup, "Parallel Implicit Adaptive Mesh Refinement Scheme for Body-Fitted Multi-Block Mesh," No. AIAA 2005-5333, pp. 1–17, 2005.
- ⁴Z. J. Zhang and C. P. T. Groth, "Parallel High-Order Anisotropic Block-Based Adaptive Mesh Refinement Finite-Volume Scheme," No. AIAA 2011-3695, 2011.
- ⁵I. Nompelis, G. V. Candler, and M. S. Holden, "Effect of Vibrational Nonequilibrium on Hypersonic Double-Cone Experiments," *AIAA Journal*, vol. 41, no. 11, pp. 2162–2169, 2003.
- ⁶G. V. Candler, M. D. Barnhardt, T. W. Drayna, I. Nompelis, D. M. Peterson, and P. K. Subbareddy, "Unstructured Grid Approaches for Accurate Aeroheating Simulations," No. AIAA 2007-3959, 2007.
- ⁷F. Ham, F. Lien, and A. Strong, "A Cartesian Grid Method with Transient Anisotropic Adaptation," *Journal of Computational Physics*, vol. 179, no. 2, pp. 469–494, 2002.
- ⁸F. Bramkamp, P. Lamby, and S. Müller, "An Adaptive Multiscale Finite Volume Solver for Unsteady and Steady State Flow Computations," *Journal of Computational Physics*, vol. 197, no. 2, pp. 460–490, 2004.
- ⁹A. Harten, "Adaptive Multiresolution Schemes for Shock Computations," *Journal of Computational Physics*, vol. 115, pp. 319–338, 1994.
- ¹⁰S. Frauholz, M. Behr, B. U. Reinartz, and M. Siegfried, "Numerical Simulation of Hypersonic Air Intake Flow in Scramjet Propulsion Using a Mesh-Adaptive Approach," No. AIAA 2012-5976, pp. 1–22, 2012.
- ¹¹M. Nemec, M. J. Aftosmis, and M. Wintzer, "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," No. AIAA 2008-0725, 2008.
- ¹²M. A. Park, "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, vol. 42, no. 9, pp. 1854–1862, 2004.
- ¹³S. R. Copeland, A. Lonkar, F. Palacios, and J. J. Alonso, "Adjoint-Based Goal-Oriented Mesh Adaptation for Nonequilibrium Hypersonic Flows," No. AIAA 2012-0552, 2013.
- ¹⁴M. J. Aftosmis and M. J. Berger, "Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries," No. AIAA 2002-0863, 2002.
- ¹⁵R. W. MacCormack and G. V. Candler, "The Solution of the Navier-Stokes Equations Using Gauss-Seidel Line Relaxation," *Computers and Fluids*, vol. 17, no. 1, pp. 135–150, 1989.
- ¹⁶G. V. Candler and R. W. MacCormack, "The Computation of Hypersonic Ionized Flows in Chemical and Thermal Nonequilibrium," *Journal of Thermophysics and Heat Transfer*, vol. 5, no. 3, pp. 266–273, 1991.
- ¹⁷P. K. Subbareddy and G. V. Candler, "A Fully Discrete, Kinetic Energy Consistent Finite-Volume Scheme for Compressible Flows," *Journal of Computational Physics*, vol. 228, no. 5, pp. 1347–1364, 2009.

- ¹⁸M. D. Bartkowicz, *Numerical Simulations of Hypersonic Boundary Layer Transition*. PhD thesis, 2012.
- ¹⁹M. J. Wright, G. V. Candler, and M. Prampolini, “Data-Parallel Lower-Upper Relaxation Method for the Navier-Stokes Equations,” *AIAA Journal*, vol. 34, pp. 1371–1377, July 1996.
- ²⁰B. Edney, “Anomalous Heat Transfer and Pressure Distributions on Blunt Bodies at Hypersonic Speeds in the Presence of an Impinging Shock,” *FFA Report 115*, 1968.
- ²¹J. Olejniczak, M. J. Wright, and G. V. Candler, “Numerical Study of Inviscid Shock Interactions on Double-Wedge Geometries,” *Journal of Fluid Mechanics*, vol. 352, pp. 1–25, 1997.
- ²²M.-C. Druguet, G. V. Candler, and I. Nompelis, “Effect of Numerics on NavierStokes Computations of Hypersonic Double-Cone Flows,” *AIAA Journal*, vol. 43, no. 3, pp. 616–623, 2005.