Implementation of Implicit Adaptive Mesh Refinement in an 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 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 method using a simple refinement sensor based on undivided differences is discussed and applied to a practical problem: 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, advances, and future work prepare this method for additional research. The potential for this method 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 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 researcher 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 the optimum for those specific conditions. With a large case matrix or number of vehicle configurations, grid generation quickly becomes an even larger percentage of the researcher's time. Additionally, by hand-crafting grids it is possible to introduce changes in grid topology that might muddle an otherwise clean comparison between two different conditions.

Grid generation is often automated to reduce the time required to generate grids 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 the grid generation process and can help ensure robust comparisons between cases. 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 to 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

[*]Graduate Student, AIAA Student Member.

[†]Research Associate, AIAA Senior Member.

[‡]Professor, AIAA Fellow

flow computation. Finally, no matter how quickly the grids can be made, maintaining a library of specific grids requires a significant allocation of hard disk space for large datasets.

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 resolve developing flow features. Most methods also allow for these locations to move while the code is running in order to track moving or 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). In addition to the above benefits, AMR can remove the requirement of having a large library of generated grids. Geometric changes might still require a large number of grids, but they are coarse and have a reduced storage requirement. Each case is initialized to a coarse grid and tailored for the specific condition. By tailoring each case, grid cells are added only in places where they are needed. This reduces the footprint on the hard disk and the computational cost as well. While there is a computational expense associated with feature detection, it can be limited to a small percentage of the total resources expended.

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, walls require cells with small heights near the surface (and usually large aspect ratios) in order to ensure a linear profile between the wall and the first off-body grid points. 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, researchers turn to implicit time advancement methods.

The authors are interested in exploring the applications for hypersonic applications and reacting flows. This work explores the requirements placed on AMR grids and develops the underpinnings necessary for further expansion of this work. Of key importance are the ability to maintain high-order numerics, provide an avenue for efficient implicit time marching, 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 approach varies 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 code 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 set-up. Flow over this shape is unsteady and can include large sections of subsonic and separated fluid. Implicit methods are required to accelerate the development of the flow structures.

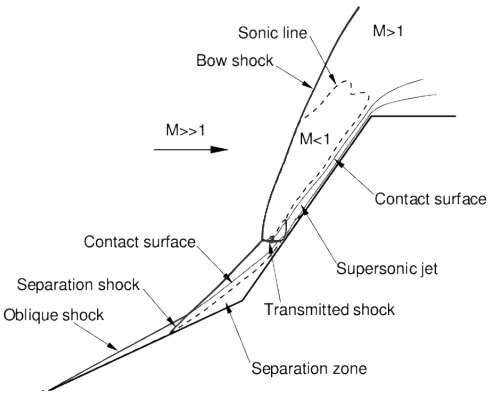
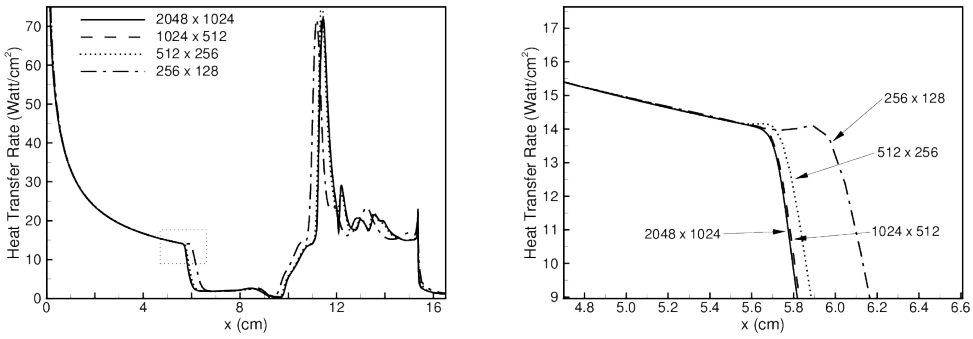


Figure 1. Important flow features in the flow near a double-cone in supersonic flow.[5]

Figure 2 shows the results of a grid sensitivity analysis 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 refinement was not necessary to adequately capture the flow here. At the separation shock, the solutions diverge (see Fig. 2(b) for more detailed view). It is possible that the differences shown downstream of this point in Fig. 2(a) are caused by the divergence in this location. If so, then local refinement in this region, and not though out the entire grid, would be sufficient to greatly improve the prediction of the coarsest grid. This could lower the required grid dimensions from 2048x1024 to something more manageable.



(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. Separation location varies with flight condition and is critical to predicting the surface pressures on the body. In addition, correctly resolving the unsteadiness in the wake and the associated vortices are required to accurately predict aerodynamic quantities. Targeted refinement with implicit AMR will allow much greater fidelity at reduced expense for problems of this type.

III. Computational Methodology

A. Flow Solver

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

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

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, $\rho u, \rho v$, and ρw are the three-dimensional components of momentum and E is the total energy per unit volume. For the inviscid Euler equations, $\vec{F}_v = 0$ and the resulting conservation equation is:

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F} = 0$$

with $\vec{F}$ indicating only the convective fluxes ,$\vec{F}_c$.

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. 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:

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

with V being the cell volume, S the face area, and $\hat{n}$ the 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 vise-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 eigenvectors. If one orients the face normals to be pointing outward from a cell, then 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 is then be written as:

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

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:

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

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.

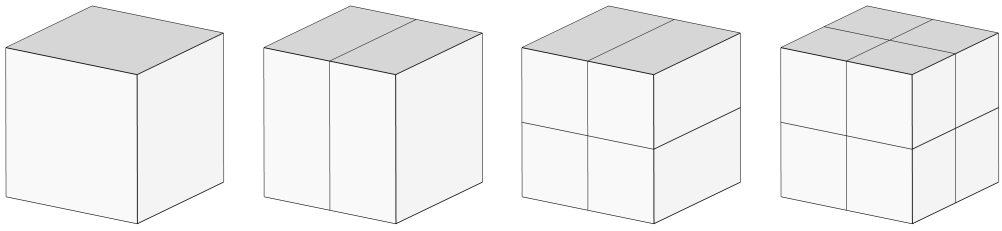
B. Adaptive Mesh Refinement Implementation

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, cells 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.

Grid meshes used in this work are constrained to hexahedral cells and not arbitrary polyhedra. 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 also 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.



(a) Original Cell (b) 1-D Subdivision (c) 2-D Subdivision (d) 3-D Subdivision

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 does not create 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. Unfortunately the added flexibility comes at a cost: there is a great deal of bookkeeping that is required in order to handle connectivity and it needs to be updated and corrected every time that the grid 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 the moving feature and without a mechanism for removing elements, the grid would become cluttered with grid where it was once, but is not currently, 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 locate the feature.

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

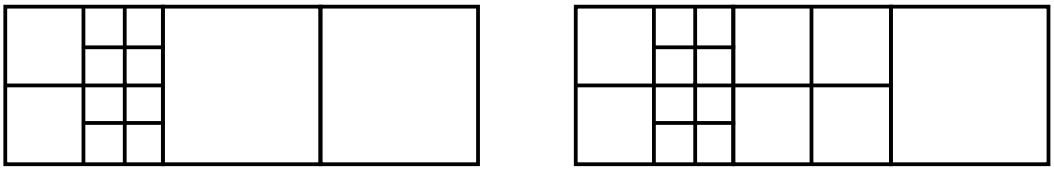
A simple refinement criteria is implemented in this work based on the differences across grid faces. This is a local evaluation and does not consider global effects refinement. Other methods are also popular in the literature. Multiscale methods as originally proposed by Harten have proven successful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

The current looks at the undivided difference of user-specified flow variables (ρ, u, v, w, p, T) to indicate if there is a sufficiently large gradient 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 to the cell-centered quantity.

Equation 1 shows the criteria used for refinement of a flow variable ϕ. Each flow 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.

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

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), not 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 risking unnecessary refinement.

The criteria for coarsening is simple. If none of the children of a parent cell are flagged to be refined or to maintain their current level of refinement by the application of Eq. 1, the requested buffer cells, or to ensure an acceptable level of refinement for face neighbors (see Fig. 4), then the children cells are candidate for removal. 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. 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.

IV. High-Order Flux Verification

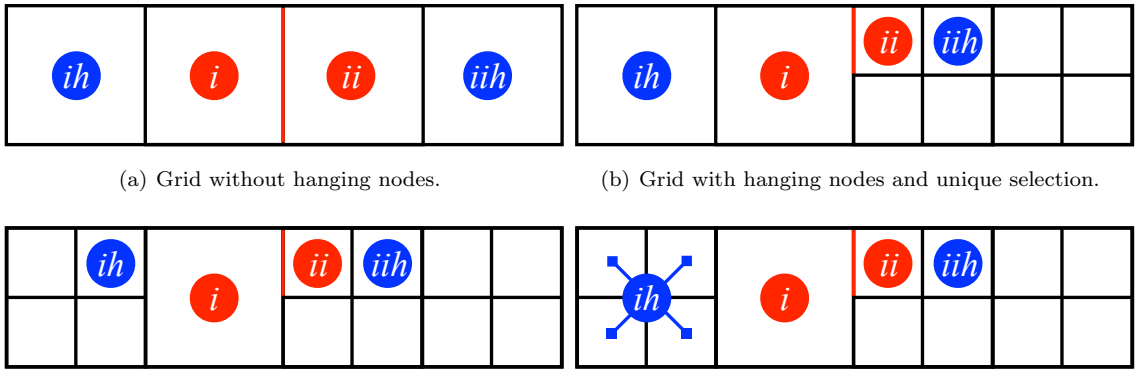
Two distinct flux methods are examined below in order to determine if the AMR method causes a reduction in their accuracy. Each of the methods will be briefly described with their dependencies identified. The methods fall into two categories: Modified Steger-Warming fluxes and Kinetic Energy Consistent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

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 red face, where the numerical flux is evaluated. Grids with hanging nodes may also create high-order partner selection that is unique. As shown in Fig. 5(b), for the red face a unique selection for ih and iih on either side of the cell.

Some ambiguity presents itself with more exotic arrangements of non-conformal cells. For the situation shown in Fig. 5(c), the selection of ih as only the top-right subcell of the originally coarse ih cell seems appropriate. Of course, another way to handle this situation might also include the bottom-right subcell as well since both are neighbors to the cell i. Figure 5(d) presents a third way to handle the situation. In this description, a restriction operation operator averages values from all children of the original coarse cell ih. That value (and the distance to the coarse cell ih) is then used as the properties for the high-order partner.

It is the last approach, Fig. 5(d), that is used in this work. 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 it is ensured that a discontinuity or large gradient does not exist in the subcells of cell ih.



(a) Grid without hanging nodes.

(b) Grid with hanging nodes and unique selection.

(c) Grid with hanging nodes and nearest cell selection.

(d) Grid with hanging nodes and coarse cell selection.

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

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. [? ?]

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 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. [15;16]

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 methods, in regions with discontinuities first-order upwdinded dissipation terms are required to maintain stability.

A. One-Dimensional Gaussian Pulse

To assess the effects of AMR on these numerics, an order of error study was performed comparing solutions on uniformly refined grids to those computed on meshes generated using AMR. Low-dissipation methods are ideal for flows without discontinuous elements (where dissipation is required). 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.[16]

The initial conditions for the 1-D density pulse is 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.

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

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 revolution with the initial density profile. The RMS of the error (weighted and normalized by volume) across all computation 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 3$^{\text{rd}}$-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.

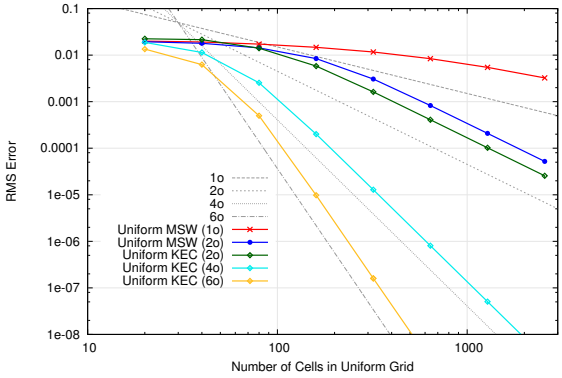
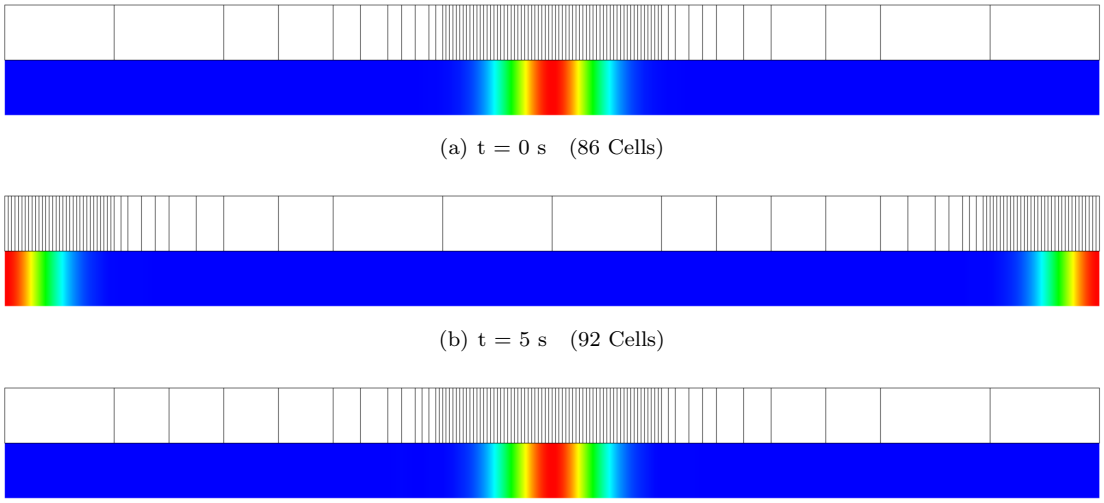


Figure 6. RMS error for 1-D density pulse after one revolution using uniform meshes.

For the comparative study using AMR, the baseline grid uses only 10 cells across the computational domain. This grid was then adapted to the maximum number of grid levels specified. To emulate the results for the uniform case with 320 cells, for example, 5 grid levels are required ($10 \cdot 2^5 = 320$). The cases were initialized on grids that were appropriately refined to capture the gradients in the prescribed density profile (Eq. 2). Furthermore, at a CFL of 0.1 AMR iterations were performed only every 5 solver iterations.The refinement was based on local variations in density governed by a tolerance value, ρ_{tol}.

Figure 7 shows the mesh used in the AMR simulation at the initial time, half-way through the simulation, and at the final time. Coarsening performs well and tracks the pulse in time. It uses 5 levels of refinement and a uniform case would require 320 points (as compared to the values listed in the figure).



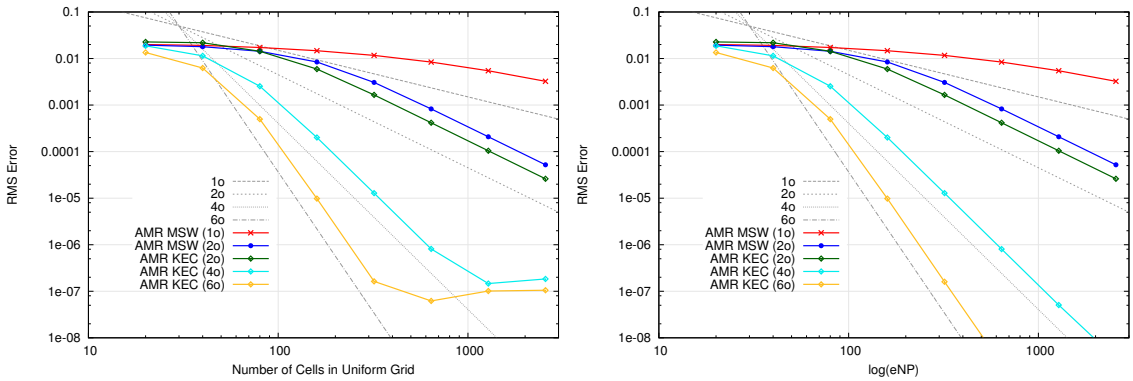
(a) t = 0 s (86 Cells)

(b) t = 5 s (92 Cells)

(c) t = 10 s (89 Cells)

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 ρ_{tol} =1E-3.

Results for the AMR cases are shown in Fig. 8 show an identical study as performed above, but with two different values of ρ_{tol}. The first, 8(a), uses a value of ρ_{tol} = 1.0E-6. It illustrates that the grids with adapted meshes achieve the prescribed spatial order in the flux calculations, but that only a minimum error is attainable. The second plot, 8(b), uses a much lower value for ρ_{tol}: ρ_{tol} = 1.0E-8. It is able to reduce the error further and matches the results shown previously with the uniform grids.



(a) ρ_{tol} = 1.0E-6

(b) ρ_{tol} = 1.0E-8

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

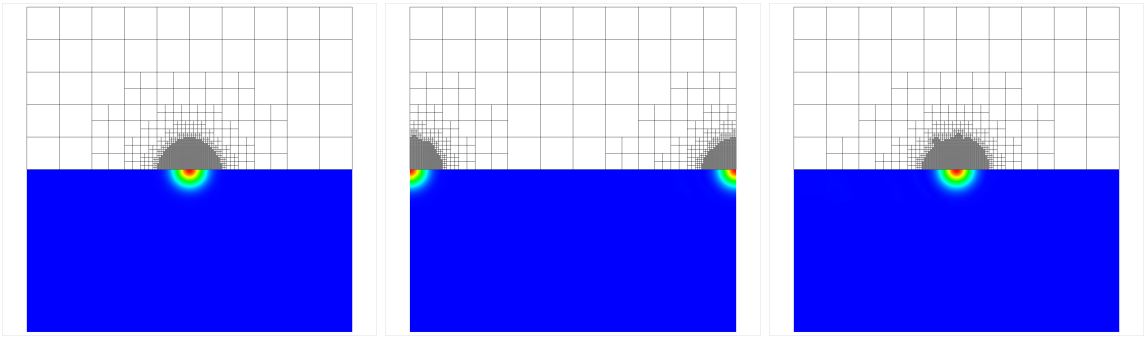
This indicates that the AMR methods presented here are able to preserve the high-order, low-dissipation numerics important to these researchers. The caveat being that the choice of the refinement tolerance parameter plays an integral role in determining the minimum attainable error. This is intuitive; 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 this does introduce a sensitivity whose impact must be assessed.

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 course 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 is identical to the one 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 across all cases.

Figure 9 shows the mesh used in the AMR simulation at the initial time, half-way through the simulation, and at the final time. As with the precious 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.



(a) t = 0 s (3,724 Cells) (b) t = 5 s (4,060 Cells) (c) t = 10 s (4,138 Cells)

Figure 9. Computational mesh, ρ contour, and cell count for 2-D gaussian pulse at three simulation times using AMR with a refinement tolerance of ρ_{tol} =1E-3.

Figure 10 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.

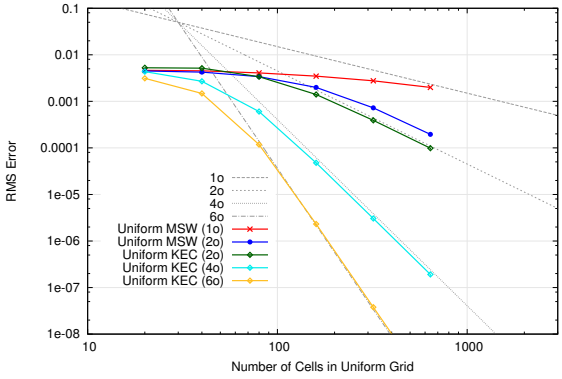
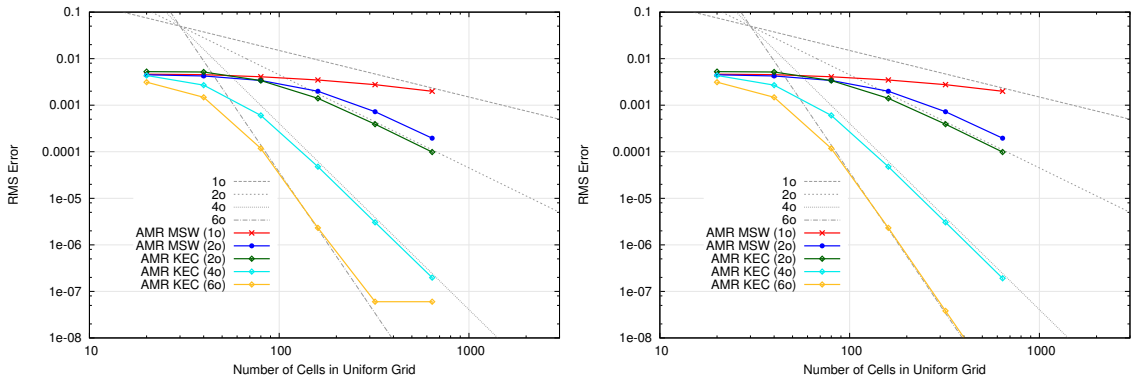


Figure 10. RMS error for 2-D density pulse after one revolution using uniform meshes.

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 (ρ_{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} of 1.0E-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 methods.



(a) $\rho_{tol} = 1.0\text{E-}6$ (b) $\rho_{tol} = 1.0\text{E-}8$

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

V. Implicit Time Integration

Explicit numerical methods are limited in the maximum allowable stable timestep. This timestep is limited by the grid spacing so for grids with a very fine mesh spacing or simulations where a large timestep is desirable, implicit time integration routines are advantageous. For hypersonic applications, usually the stringent requirements on grid spacing near the wall dominates the determination of the maximum stable timestep. 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 in order to more quickly converge the simulations.

The explicit routine 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:

$$\begin{aligned} F^{n+1} &= F^n + \frac{\partial \vec{F}}{\partial \bar{U}}^n \left(\bar{U}^{n+1} - \bar{U}^n\right) \\ &= F^n + A^n_+ \left(\bar{U}^{n+1}_i - \bar{U}^n_i\right) + A^n_- \left(\bar{U}^{n+1}_{ii} - \bar{U}^n_{ii}\right) \\ &= F^n + A^n_+ \partial\bar{U}_i + A^n_- \partial\bar{U}_{ii} \end{aligned}$$

where A^n_+ and A^n_- are the right- and left-running flux Jacobians at the face. $\partial\bar{U}_i$ and $\partial\bar{U}_i$ 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:

$$\partial\bar{U}_i + \frac{\Delta t}{V_i} \sum_{faces} \left[\left(A^n_+ \partial\bar{U}_i + A^n_- \partial\bar{U}_{ii}\right) \cdot \hat{n} S\right] = \Delta\bar{U}_i$$

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

Time-accurate simulations require a non-biases approach to the implicit solve. 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 high-fidelity computations of turbulence. To achieve an efficient implicit solve that is unbiased, full-matrix point-implicit methods are used. This method is called full-matrix point-relaxation. This strategy relaxes the values in all neighboring cells during the implicit solve.

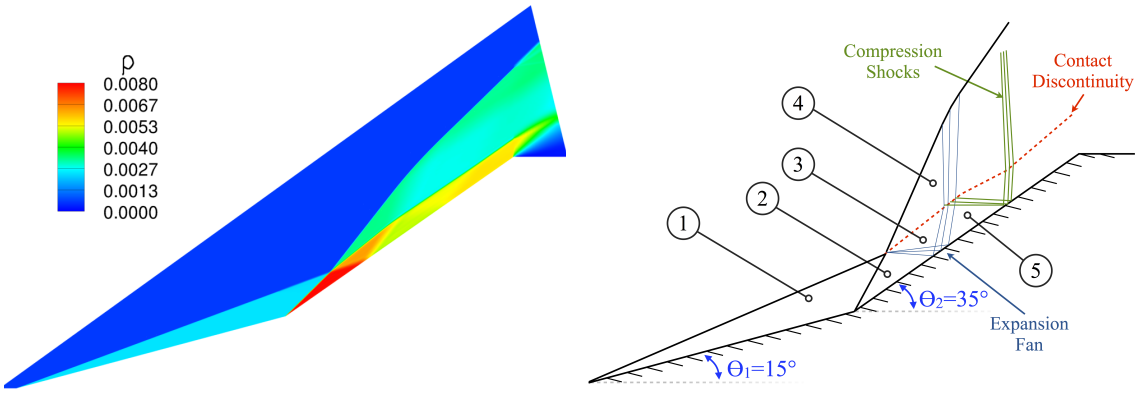
VI. Double Wedge Simulations

An important phenomena of interest to researchers 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 (or biconic). By varying the freestream conditions and the two wedge/cone angles, an entire family of interaction types can be realized (as classified by Edney).[17] The double wedge or double cone provide a simple test case for comparison between computational methods and a means for computational validation.

These 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 computational simulations of implicit double wedges.[18] 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 should present 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) Density contour in solution.

(b) Schematic of flow features.

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

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.[18] 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 with 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 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 this effect. It is only apparent in regions of the flow where the solution is already sensitive to misalignment between the shocks and the grid cells and where the original grid was highly skewed due to the smoothing operations. Figure 13 shows a close-up on 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.

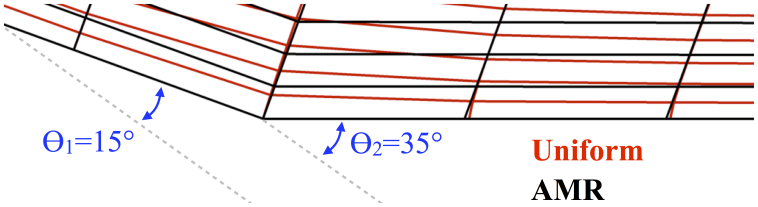


Figure 13. View of mesh for 512×512 grid using smoothed fine grid and coarse grid with AMR.

B. Results

Simple measure for accuracy of the simulations is the surface pressure on the double wedge. For the attached, inviscid case, it is possible to determine many of the pressures on the wedge 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 attained 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 form 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 greatest 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, though.

For the cases with AMR, the case was initially run on a grid of resolution 32×32. Once the solution was deemed converged (RMS residual had dropped four orders of magnitude), the grid was refined

Shown below are the results for a similar set of cases as well as finer ones that were too expensive to prepare for inclusion into this work using a serial code. The AMR procedure was performed once the solution

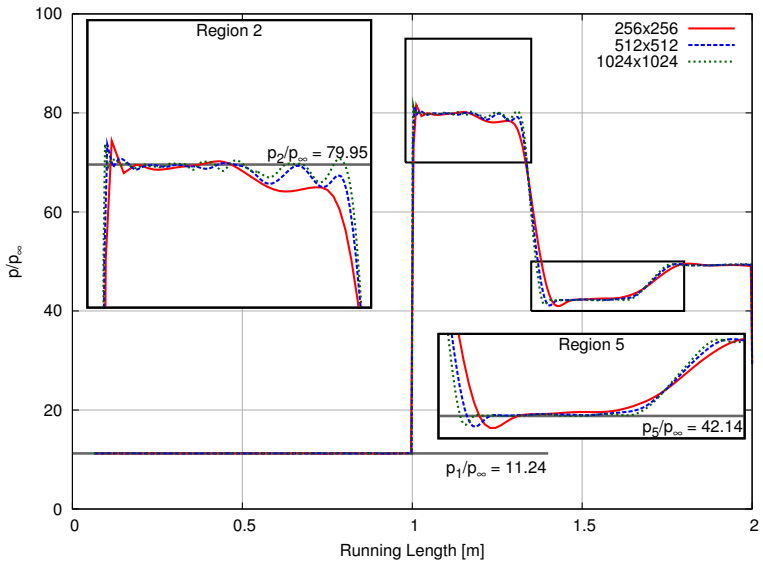
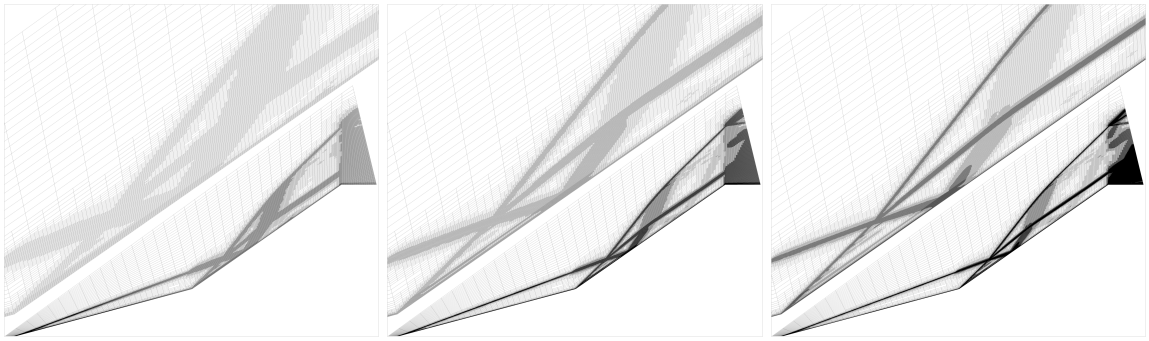


Figure 14. Surface pressure on the wedge for a series of uniform grids with varying grid densities.

was deemed converged (residual magnitude had dropped five orders of magnitude from its initial value). Cells identified for refinement using a joint local tolerance of $\rho_{tol} = 1\text{E-}2$ and $p_{tol} = 1\text{E-}2$ 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.



(a) 4 levels (512 ×512) (b) 5 levels (1024×1024) (c) 6 levels (2048×2048)

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

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 for the pressures in region 2. These results illustrate that grid convergence may be obtained near a resolution of 4096×4096. This is higher than was estimated earlier based on the work by Olejniczak et al. using similar numerics.[18]

Grid 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 reduction in grid quality in the simple cell subdivision (see Fig. 13. With refinement in the AMR, the oscillations in region 2 become much less pronounced and the pressure peak becomes more localized, even though its magnitude remains unchanged. It is interesting to note that the oscillations in region 2 are much less pronounced for the AMR case.

The residual plots for these cases are shown in Fig. 17. These plots are versus simulated time. For this

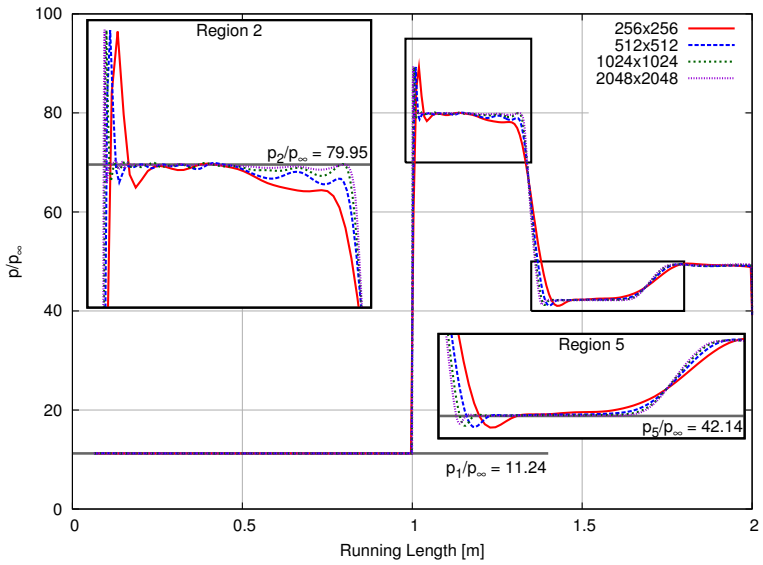
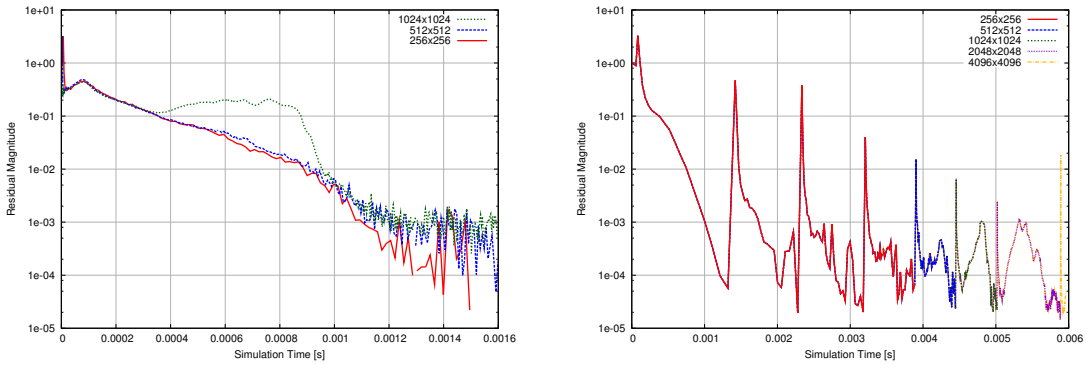


Figure 16. Surface pressure on the wedge for a series of grids with varying levels of AMR.

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. The uniform grid cases converge in slightly more than one flow time. For a completely supersonic flowfield, this is not surprising. The residual plot for the AMR grid requires considerably more simulated time because slightly less than one flowtime is required at each grid level. Fortunately, with dramatically fewer points in the simulation, these calculations require only a small amount of computational time.



(a) Uniform Grid (b) AMR Grid

Figure 17. View of rotated mesh for 512×512 grid using smoothed fine grid and coarse grid with AMR.

C. Computational Efficiency

These steady wedge flows require fewer calls to the routine since features are generally stationary once they develop 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-wedge incorporates a separation bubble at the corner between the two wedges. 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. [5]

With such a flow feature, the AMR routine would have to be called several times throughout the development as the separation region changes size and the shocks resulting from the interactions move. Fortunately, each call to the AMR routine requires computational time on the order of a timestep, so provided it was not called *every* timestep it should still yield considerable improvement in time to solution. In addition to refinement upon residual convergence, our code includes an optional input for a refinement frequency. When included, AMR is performed every n iterations regardless of the residual - n being the frequency chosen.

To understand how the method would perform 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 shown. Since the AMR process has a cost that is equivalent to less than 1 flow solve iteration, it is expected that any change in the performance for different values of n would not be influenced considerably.

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. Cases with three levels of refinement require roughly 26% of the cells and 11% of the wall time as compared to the uniform grid case. When using four levels of refinement, the improvement in cell count and cost is more dramatic (17% and 5%, respectively). The results are even more favorable with additional refinement (10% and 2-3%). For a flow with well-defined, localized features, additional levels of AMR prove very effective.

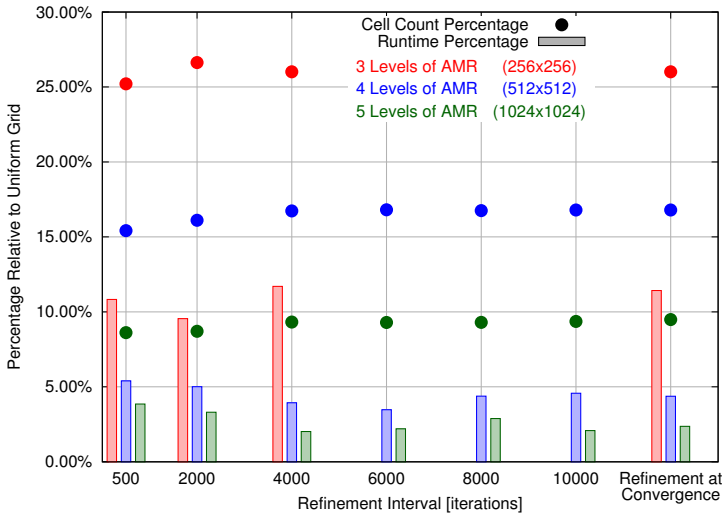


Figure 18. View of rotated mesh for 512×512 grid using smoothed fine grid and coarse grid with AMR.

Another observation is that in general, the more frequent the AMR (lower value of n), the required cell count is lowest but the time to solution is greatest. Fewer cells are required because more cells are coarsened once the flowfield is more resolved. Cases with larger refinement intervals have much fewer opportunities to remove cells that were added in previous refinement iterations. The increased cost is due to the initial transients that are not allowed to fully develop before the grid is refined. A more efficient approach would allow even the lowest values of n to achieve a steady-state solution on the coarsest grid before applying the refinement frequency parameter.

VII. Conclusions

The results from this work indicate that AMR shows promise for highly sensitive problems requiring high-order methods and for shock-dominated flows. Implicit computations are effective and compatible with non-conformal unstructured meshes . Not only does the method provide a mechanism for enriching a mesh without extensive user-input, but it also achieves an accurate answer with a minimum of computational expense.

This work outlines the initial efforts at developing a finite-volume AMR method for future research. 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 addition of the Navier-Stokes viscous fluxes. Incorporating these fluxes into the flow solver is necessary to begin validating against experimental results and to simulate real-world conditions. Inclusion of the numerical mechanics necessary for multi-species, non-reacting flow is also underway.

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.

Acknowledgements

This work was sponsored by the NASA Johnson Space Center Academic 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

[1] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, pp. 484–512, 1984. 2, 5, 6

[2] M. J. Berger and P. Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, vol. 84, pp. 64–84, 1989. 2

[3] C. 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. 3

[4] Z. J. Zhang and C. P. T. Groth, "Parallel High-Order Anisotropic Block-Based Adaptive Mesh Refinement Finite-Volume Scheme," No. AIAA 2011-3695, 2011. 3, 5

[5] I. 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. 3, 16

[6] G. V. Candler, M. D. Barnhardt, T. W. Drayna, I. Nompelis, D. M. Peterson, and P. K. Subbareddy, "Unstructured Grid Approaches for Accurate Aeroheating Simulations," in *Evaluation*, no. AIAA 2007-3959, 2007. 5

[7] F. 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. 5

[8] F. 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. 5

[9] A. Harten, "Adaptive Multiresolution Schemes for Shock Computations," *Journal of Computational Physics*, vol. 115, pp. 319–338, 1994. 6

[10] S. 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. 6

[11] M. Nemec, M. J. Aftosmis, and M. Wintzer, "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," No. AIAA 2008-0725, 2008. 6

[12] M. A. Park, "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, vol. 42, no. 9, pp. 1854–1862, 2004.

[13] S. 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. 6

[14] M. J. Aftosmis and M. J. Berger, "Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries," No. AIAA 2002-0863, 2002. 6

[15] P. 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. 8

[16] M. D. Bartkowicz, *Numerical Simulations of Hypersonic Boundary Layer Transition*. PhD thesis, 2012. 8

[17] B. Edney, "Anomalous Heat Transfer and Pressure Distributions on Blunt Bodies at Hypersonic Speeds in the Presence of an Impinging Shock," 1968. 12

[18] J. 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. 12