

Parallel High-Order Anisotropic Block-Based Adaptive Mesh Refinement Finite-Volume Scheme

Z. J. Zhang * and C. P. T. Groth †

*University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, Ontario, M3H 5T6, Canada*

A novel parallel, high-order, anisotropic, block-based, adaptive mesh refinement (AMR), finite-volume scheme is proposed and developed herein for the numerical solution of physically complex flow problems having disparate spatial and temporal scales, and with strong anisotropic features. A block-based AMR approach is used which permits highly efficient and scalable implementations on parallel computing architectures and the use of multi-block, body-fitted computational grids for the treatment of complex geometries. Rather than adopting a more usual isotropic approach to the refinement of the grid blocks, the proposed approach uses a binary hierarchical tree data structure that allows for anisotropic refinement of the grid blocks in each of the coordinate directions in an independent fashion. This allows for the more efficient and accurate treatment of narrow layers, discontinuities, and/or shocks in the solutions which occur, for example, in the thin boundaries and mixing layers of high-Reynolds-number viscous flows and in the regions of strong non-linear wave interactions of high-speed compressible flows with shocks. The anisotropic AMR approach is combined with a computationally efficient, high-order, central, essentially non-oscillatory (CENO), cell-centered, upwind, finite-volume scheme and an efficient parallel, implicit and explicit scheme for the solution of general systems of partial differential equations governing both steady and time-varying problems. The CENO upwind scheme makes use of Riemann-solver based flux functions and a solution smoothness indicator to provide robust, accurate, and monotonic treatment of shocks and under-resolved solution content. The finite-volume scheme is applied to the solution of both a model system, the advection-diffusion equation, as well as the Euler equations governing compressible, inviscid, gaseous flows in two space dimensions. The potential of the parallel adaptive scheme for dealing with flows having disparate scales is clearly demonstrated.

I. Introduction and Motivation

I.A. Algorithm Design Needs for Physically Complex Flows

Computational fluid dynamics (CFD) has proven to be an important enabling technology in many areas of science and engineering. In spite of its relative maturity and widespread successes in aerospace engineering, there remain a variety of physically-complex flows, which are still not well understood and have proven to be very challenging to predict by numerical methods. Such flows would include but are not limited to: (i) multiphase, turbulent, and combusting flows encountered in propulsion systems (e.g., gas turbine engines and solid propellant rocket motors); (ii) compressible flows of conducting fluids and plasmas; and (iii) micro-scale non-equilibrium flows, such as those encountered in micro-electromechanical systems. These flows present numerical challenges for they generally involve a wide range of complicated physical/chemical phenomena, exhibit strong anisotropic solution features, as well as involve complex flow geometries.

Hand-in-hand with CFD algorithm development, the rapid increase in high-performance computing systems in the last 10–15 years has led to terascale and, more recently, petascale parallel distributed memory clusters ranging in size from a few thousand to hundreds of thousands of cores and capable of up to more than one petaflop/s. These advances in computing hardware are, in turn, creating significant opportunities

*M. A. Sc. Candidate, jenmyzm.zhang@utoronto.ca

†Professor, Senior Member AIAA, groth@utias.utoronto.ca

for CFD of physically-complex flows. Nevertheless, advances in numerical methods are required to fully exploit terascale and petascale, computing platforms and thereby enable the routine solution of physically-complex flows for practical engineering applications. A recent assessment of the needs for large-scale and high-performance scientific computing indicates that a number of fundamental issues in discretization design must be addressed.¹ Issues and challenges identified included: (i) greater automation of mesh generation via adaptive mesh refinement (AMR) to reduce the time to generate high-quality meshes and for the treatment of complex geometries; (ii) efficient parallel implementations of fully implicit time-marching methods for use in combination with AMR; (iii) efficient high-order temporal and spatial discretizations for reduced computational cost for a specified level of accuracy; and (iv) accurate and robust treatments of multi-scale anisotropic physics.

I.B. Block-Based Adaptive Mesh Refinement

One approach to reducing the computational costs of physically-complex flow simulations is to use AMR.²⁻⁷ Computational grids that automatically adapt to the solution are very effective in treating problems with disparate length scales, providing the required spatial resolution while minimizing memory and storage requirements. The use of AMR in conjunction with finite-volume schemes has produced some very powerful methods for the treatment of a wide variety of physically-complex flows with complex and/or moving geometries. Moreover, block-based variants of these techniques have been proposed for parallel implementation.^{6,8,9} In particular, Groth and co-researchers¹⁰⁻¹⁶ have developed a block-based parallel AMR method for body-fitted multi-block mesh. Their block-based approach has been shown to enable efficient and scalable parallel implementations for a variety of complex flows, as well as allow for local refinement of body-fitted mesh with anisotropic stretching. In spite of these successes, fully reliable and accurate AMR strategies for high-order schemes and viscous flows, as well as efficient parallel implementations for unsteady flows with dynamic adaptation, have yet to be devised.

I.C. High-Order Finite-Volume Schemes

The potential of high-order methods to reduce the cost of simulations for physically-complex flows is also recognized. Standard lower-order methods (i.e, methods up to second order) can exhibit excessive numerical dissipation for multi-dimensional problems and are often not practical for physically-complex flows. Improved numerical efficiency may be achieved by raising the order of accuracy of the spatial discretization, thereby reducing the number of computational cells required to achieve the desired accuracy. For hyperbolic conservation laws and/or compressible flow simulations, the challenge has been to achieve accurate discretizations while coping in a reliable and robust fashion with discontinuities and shocks.¹⁷ High-order schemes for partial differential equations (PDEs) governing diffusion processes and having an elliptic nature have also been considered. It is then desirable that the discretization of the elliptic operator remains accurate while satisfying a maximum principle, even on stretched/distorted meshes. In the last decade, there have been a number of studies of high-order schemes, including finite-volume,¹⁷⁻²⁶ discontinuous Galerkin,²⁷⁻³¹ and spectral finite-difference and finite-volume methods,³²⁻³⁶ for both structured and unstructured mesh. In spite of these advances, there is still no consensus for a robust, efficient, and accurate scheme that fully deals with all of the aforementioned issues and is universally applicable to arbitrary meshes. For this reason, development of effective high-order methods remains an active area of research.

In the essentially non-oscillatory (ENO) high-order finite-volumes schemes proposed by Harten *et al.*,¹⁷ the use of computational stencils that contain discontinuities is avoided to achieve solution monotonicity. Although a weighted ENO (WENO) scheme attempts to simplify the ENO procedure by adopting a stencil-weighting approach, both ENO and WENO variants encounter difficulties when selecting appropriate stencils for general multi-dimensional unstructured meshes,^{19,20,23,37} and they can result in poor conditioning of the linear systems that are involved in the solution reconstruction using such stencils.^{23,37} These, along with the associated computational cost and complexities involved in the ENO and WENO finite-volume schemes, have limited the range of applications where they may be applied. Ivan and Groth^{38,39} have recently proposed a high-order central ENO (CENO) finite-volume scheme which attempts to deal with these computational issues. As a variant of the ENO scheme, the CENO scheme avoids the complexity associated with the other schemes by using a fixed central stencil. A central stencil in general provides the most accurate reconstruction due to cancellation of truncation errors. This hybrid scheme first performs an unlimited, high-order, k -exact reconstruction using the least-squares technique,¹⁸ and then automatically reverts to a monotonicity

preserving limited, piecewise, linear, least-squares reconstruction procedure in cells near shocks or with under-resolved solution content. The switching is controlled by a solution smoothness indicator. The CENO reconstruction is effective in eliminating the appearance of $O(1)$ numerical oscillations in under-resolved regions and in solutions that contain strong discontinuities and/or shocks. Although uniform accuracy is not achieved for non-smooth solutions, the method is easily extendable to both multi-dimensions and unstructured mesh.

I.D. Proposed Algorithm

With the preceding viewpoints in mind, a somewhat novel parallel, high-order, anisotropic, block-based, AMR, finite-volume scheme is proposed and developed herein. In most isotropic AMR methods, the mesh resolution is doubled in all directions during each refinement. This may not be efficient in applications where the solution is rapidly changing in one or more directions but not in the others. Therefore, rather than adopting a more usual isotropic approach to the refinement of the grid blocks, the proposed approach uses a binary hierarchical tree data structure that allows for anisotropic refinement of the grid blocks in each of the coordinate directions in an independent fashion. Unlike the anisotropic extensions to AMR procedures considered by other researchers in the past (see, for example, the paper by Ham *et al.*⁴⁰), the proposed anisotropic AMR approach allows block-based refinement in a preferred direction, while still preserving the parallel efficiencies of block-based isotropic methods. This allows for the more efficient and accurate treatment of narrow layers, discontinuities, and/or shocks in the solutions which occur, for example, in the thin boundaries and mixing layers of high-Reynolds-number viscous flows and in the regions of strong non-linear wave interactions of high-speed compressible flows with shocks. In addition, the anisotropic AMR method allows the aspect ratios of the computational blocks to vary unlike in isotropic AMR methods, allowing the resulting mesh to better conform to the variation of the solution, even for problems that are not strongly anisotropic.

This increase in flexibility for refinement leads to a further reduction in the number of computational cells required to attain desired spatial accuracy. The anisotropic AMR approach is combined with the computationally efficient, high-order, CENO, upwind scheme of Ivan and Groth^{38,39} and a parallel, implicit and explicit scheme for the solution of general systems of partial differential equations governing both steady and time-varying problems. As was previously shown by Jameson,⁴¹ the combination of high-order spatial discretizations with AMR may provide very effective means of obtaining high solution accuracy. The proposed finite-volume scheme is applied to the solution of both a model system, the advection-diffusion equation, as well as the Euler equations governing compressible, inviscid, gaseous flows in two space dimensions. The potential of the parallel adaptive scheme for dealing with flows having disparate scales is clearly demonstrated.

II. High-Order CENO Finite-Volume Scheme

II.A. Two-dimensional Advection-Diffusion Equation

A model advection-diffusion equation with a scalar solution variable, u , is first considered for the evaluation of the proposed high-order anisotropic AMR scheme. This model equation is given by

$$\frac{\partial u}{\partial t} + \vec{\nabla} \cdot (\vec{V}u) = \vec{\nabla} \cdot (\kappa \vec{\nabla} u) + \phi(x, y, u). \quad (1)$$

The first term of the equation is the time rate of change of u ; the advective flux is represented by $\vec{\nabla} \cdot (\vec{V}u)$, where \vec{V} is the prescribed advective velocity field; the diffusive flux is represented by $\vec{\nabla} \cdot (\kappa \vec{\nabla} u)$, where κ is the diffusion coefficient. The Peclet number, which is defined as $Pe = aL/\kappa$, offers an indication of whether a given problem is advection-dominated or diffusion-dominated. The last term of Equation (1) is a source term, which may be non-linear if dependent on u .

II.B. Two-dimensional Euler Equations for Inviscid Compressible Gasdynamics

Application of the proposed numerical framework in solving the Euler equations allows the scheme to be evaluated in the context of more interesting and realistic problems. The conservation form of the Euler equations is given as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \vec{\nabla} \cdot \vec{\mathbf{F}} = 0, \quad (2)$$

which, in two-dimensions (2D), could be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0. \quad (3)$$

The vector of conserved solution variables, \mathbf{U} , is given by

$$\mathbf{U} = [\rho, \rho u, \rho v, \rho e]^T, \quad (4)$$

where ρ is the gas density, u and v are x and y components of the velocity vector, $e = p/(\rho(\gamma - 1)) + u^2/2$ is the specific total energy, $p = \rho RT$ is the pressure, T is the gas temperature, R is the ideal gas constant and γ is the specific heat ratio. Furthermore, the flux vectors \mathbf{F} and \mathbf{G} , associated with the x - and y -direction respectively, are each defined as

$$\mathbf{F} = [\rho u, \rho u^2 + p, \rho uv, u(\rho e + p)]^T, \quad (5)$$

and

$$\mathbf{G} = [\rho v, \rho uv, \rho v^2 + p, v(\rho e + p)]^T. \quad (6)$$

II.C. Semi-Discrete Form

After applying a finite-volume spatial discretization to Equations (1) and (3) for a 2D quadrilateral cell indexed (i, j) , the following semi-discrete form can be obtained:

$$\frac{d\bar{\mathbf{U}}_{i,j}}{dt} = -\frac{1}{A_{i,j}} \sum_{l=1}^{N_f} \sum_{m=1}^{N_G} \left(\omega \vec{\mathbf{F}} \cdot \vec{\mathbf{n}} \Delta l \right)_{i,j,l,m} + \mathbf{S}_{i,j} = \mathbf{R}_{i,j}, \quad (7)$$

where

$$\bar{\mathbf{U}}_{i,j} = \frac{1}{A_{i,j}} \int_A \mathbf{U}_{i,j}^k dA. \quad (8)$$

This is achieved by integrating each term in the PDEs over the control volume of the cell, which is its area $A_{i,j}$, and subsequently applying the divergence theorem. The cell solution average is denoted $\bar{\mathbf{U}}_{i,j}$. Solution state obtained by evaluating the k -order reconstruction polynomial, $\mathbf{U}_{i,j}^k$, is used in calculating the interface flux. The reconstructed solution $U_{i,j}^k$ is determined using the CENO approach,^{38,39} which is constrained to preserve $\bar{\mathbf{U}}_{i,j}$ when integrated over cell (i, j) of area $A_{i,j}$. The total flux through each face, $\vec{\mathbf{F}}$, is computed using an N_G -point Gauss Quadrature numerical integration procedure, after which the fluxes through $N_f = 4$ faces of the cell are summed. Sufficient number of Gauss points are chosen per face such that the accuracy of the high-order scheme is maintained. Finally, $\mathbf{R}_{i,j}$ is the cell residual.

II.C.1. Advection-Diffusion Equation

For the 2D advection-diffusion problem, Equation (7) reduces to a scalar equation, so that $\bar{\mathbf{U}}_{i,j}$ is replaced by $\bar{u}_{i,j}$. The source term $S_{i,j}$ for the advection diffusion equation is given by $1/A_{i,j} \int_A \phi_{i,j}(x, y, u) dA$. The scalar flux is given by $\vec{F} = \vec{V}u - \kappa \vec{\nabla}u$, which has an advective and a diffusive component. The hyperbolic, or advective, flux is defined as $\vec{F}_a \cdot \vec{\mathbf{n}} = u \vec{\mathbf{V}} \cdot \vec{\mathbf{n}}$. An upwinding procedure is used such that

$$\vec{F}_a \cdot \vec{\mathbf{n}} = \begin{cases} u_l(\vec{\mathbf{V}} \cdot \vec{\mathbf{n}}) & \text{if } \vec{\mathbf{V}} \cdot \vec{\mathbf{n}} \geq 0, \\ u_r(\vec{\mathbf{V}} \cdot \vec{\mathbf{n}}) & \text{if } \vec{\mathbf{V}} \cdot \vec{\mathbf{n}} < 0, \end{cases} \quad (9)$$

where u_r and u_l are the left and right solution states evaluated at each Gauss Quadrature point of the cell interface. The elliptic, or diffusive, flux is defined as $\vec{F}_d \cdot \vec{\mathbf{n}} = -\kappa \vec{\nabla}u \cdot \vec{\mathbf{n}}$. At every cell interface, it is taken as the arithmetic average of the diffusive fluxes from the left and right states. Therefore, it is expressed as

$$\vec{F}_d \cdot \vec{\mathbf{n}} = -\kappa \left[\frac{1}{2} \left(\vec{\nabla}u_l^k(\vec{r}) + \vec{\nabla}u_r^k(\vec{r}) \right) \right] \cdot \vec{\mathbf{n}}, \quad (10)$$

and \vec{r} is the position vector to the Gauss quadrature point. $\vec{\nabla}u_l^k(\vec{r})$ and $\vec{\nabla}u_r^k(\vec{r})$ refer to the solution gradient computed analytically from the k -order accurate CENO reconstructions for the left and right states. Gradient evaluation in the elliptic flux evaluation using a k -order CENO reconstruction is k -order accurate, hence the scheme is globally k -order accurate.

II.C.2. Euler Equations

The source term $\mathbf{S}_{i,j}$ is zero for the Euler equations. The flux term $\vec{\mathbf{F}}$ is given by $\vec{\mathbf{F}} = \mathbf{F}\hat{i} + \mathbf{G}\hat{j}$, where \mathbf{F} and \mathbf{G} are the flux vectors defined previously. The numerical flux could be written as follows in terms of the left and right states:

$$\vec{\mathbf{F}} \cdot \vec{n} = \vec{\mathbf{F}}(\mathbf{U}_{int}(\mathbf{U}_l, \mathbf{U}_r, \vec{n})) \cdot \vec{n}. \quad (11)$$

The interface solution at every Gauss Quadrature point of all cell faces, \mathbf{U}_{int} , is obtained by solving a Riemann problem in the direction indicated by the face normal \vec{n} , and with initial states defined by \mathbf{U}_l and \mathbf{U}_r on either side of the cell interface. CENO reconstruction of k -order is used to evaluate \mathbf{U}_l and \mathbf{U}_r , in a similar fashion as the hyperbolic flux evaluation in the advection-diffusion equation. Since Euler equations are purely hyperbolic, a $k+1$ -order accurate spatial discretization could be achieved with a k -order reconstruction, assuming a smooth solution. Both approximate⁴²⁻⁴⁵ and exact Riemann solvers⁴⁶ have been implemented. The Roe linearized Riemann solver⁴² has been used for studies in this paper unless otherwise stated.

II.D. CENO Reconstruction

A piecewise k -order polynomial reconstruction within each computational cell is performed using a high-order central ENO (CENO) method. The CENO method is a hybrid solution reconstruction procedure which combines the high-order k -exact least-squares reconstruction technique by Barth¹⁸ using a fixed central stencil, with a monotonicity preserving limited piecewise linear least-squares reconstruction algorithm.¹⁸ Switching in the hybrid procedure is determined by whether the solution is sufficiently resolved on the computational mesh as indicated by a solution smoothness indicator.

The k -exact reconstructed solution of variable u in cell (i, j) assumes a polynomial form of the following:³⁸

$$u_{i,j}^k(\vec{r}) = \sum_{p_1=0}^{N_1} \sum_{p_2=0}^{N_2} (x - \bar{x}_{i,j})^{p_1} (y - \bar{y}_{i,j})^{p_2} D_{p_1 p_2}^k, \quad (12)$$

where N_1 and N_2 satisfy $N_1 + N_2 \leq k$; $\bar{x}_{i,j}$ and $\bar{y}_{i,j}$ are coordinates of the cell centroid; and $D_{p_1 p_2}^k$ are the coefficients for each polynomial term.³⁸ To determine the values of $D_{p_1 p_2}^k$, constraints are imposed requiring that the average solutions of cell (i, j) and cells in its reconstruction stencil are conserved, or that $\bar{u}_{i,j} = (1/A_{i,j}) \iint_{A_{i,j}} u_{i,j}^k(\vec{r}) dx dy$. Additional neighboring cells are included in the reconstruction stencil beyond what is required to uniquely determine all coefficients, so that reconstruction is more robust for stretched meshes or meshes that are not aligned with the solution gradients. The current reconstruction scheme uses a fixed central stencil with 8 neighboring cells for $k=1$ and 24 neighboring cells for $k=2$, $k=3$ and $k=4$.³⁸

The above formulation results in an overdetermined system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{B}$, which is solved using orthogonal decomposition by SVD method.⁴⁷ This method involves the computation of a pseudo-inverse matrix \mathbf{A}^{-1} , where \mathbf{A} is unchanged for a fixed stencil in CENO.³⁸ Therefore, the pseudo-inverse matrix could be stored and reused to speed up the computation. This is another advantage of the CENO method, in addition to the avoidance of reconstruction on multiple stencils and poorly conditioned coefficient matrices \mathbf{A} which may occur in the other ENO and WENO schemes. Preservation of solution average in cell (i, j) is explicitly enforced by expressing D_{00}^k as a function of the other unknowns.³⁸

II.D.1. Smoothness Indicator

The CENO scheme preserves solution monotonicity in regions of large gradients or discontinuities by reverting the high-order k -exact reconstruction to a limited piecewise linear ($k = 1$) reconstruction. The slope limiters of Barth-Jespersen and Venkatakrishnan could be used.^{18,48} The switching is triggered by a smoothness indicator, which identifies under-resolved regions that may require a drop in order. The smoothness indicator, \mathcal{S} , is calculated in terms of a smoothness parameter α in the following manner:³⁸

$$\mathcal{S} = \frac{\alpha}{\max((1-\alpha), \varepsilon)} \frac{(SOS - DOF)}{DOF - 1}, \quad \text{where} \quad \alpha = \frac{\sum_{\gamma} \sum_{\delta} (u_{\gamma,\delta}^k(\vec{r}_{\gamma,\delta}) - u_{i,j}^k(\vec{r}_{\gamma,\delta}))^2}{\sum_{\gamma} \sum_{\delta} (u_{\gamma,\delta}^k(\vec{r}_{\gamma,\delta}) - \bar{u}_{i,j})^2}. \quad (13)$$

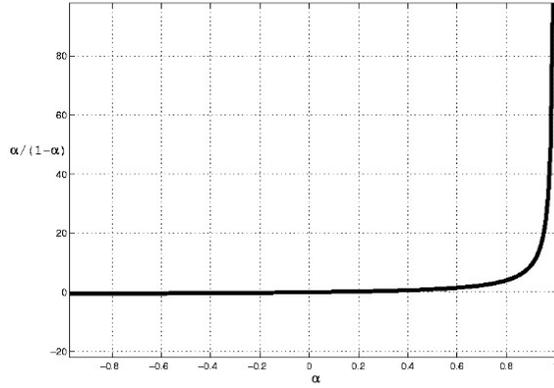


Figure 1: The graph of $f(\alpha) = \frac{\alpha}{(1-\alpha)}$.

In the above equation, *SOS* and *DOF* refer to the reconstruction stencil size and number of unknowns respectively, whereas ε is a tolerance added to prevent division by zero. The computation of α involves comparing values of reconstructed solution at the centroids, $\vec{r}_{\gamma,\delta}$, of all stencil cells indexed by γ and δ .³⁸ Figure 1 illustrates the behavior of $\frac{\alpha}{(1-\alpha)}$, which rapidly increases to ∞ as α approaches 1, indicating a smooth and well-reconstructed solution.³⁸ Finally, a smoothness cutoff value \mathcal{S}_c is specified, so that if $\mathcal{S} > \mathcal{S}_c$, the reconstruction is considered smooth, otherwise the order of reconstruction is dropped locally.

II.D.2. Reconstruction at the Boundaries

Correct high-order treatment of boundary conditions is crucial in maintaining the accuracy of the scheme. In the current approach, extra rows of ghost cells are added beyond the geometric boundary of the computational domain to impose high-order boundary conditions. Least-squares reconstruction in control volumes adjacent to the boundary are constrained as described by Olivier-Gooch and Van Altera.²⁶ Furthermore, geometric data are computed to the same order of accuracy as that of the interior scheme. The Boundary constraints in addition to the reconstruction conditions lead to an enlarged system of equations, which is solved using Gauss elimination with pivoting followed by a Householder QR factorization for the remaining least-squares problem.

II.E. Solution of the Semi-Discrete Form

For time-accurate simulation of unsteady problems, multi-stage explicit time-marching schemes have been used to advance the ordinary differential equation described by Equation (7) in time. A fourth-order Runge-Kutta scheme is used for $k > 2$ to preserve the global accuracy of the scheme. For most steady-state problems, an implicit NKS method is used to speed up the solution process, which will be described in a later section.

III. Anisotropic Adaptive Mesh Refinement

III.A. Overview of Block-Based Adaptive Mesh Refinement

Adaptive mesh refinement (AMR) techniques ensure all regions on the domain are sufficiently resolved without over-resolving low-gradient regions, and is therefore very effective in treating problems with disparate spatial scales. The AMR technique described in this paper follows the approach by Groth *et al.* for computational magnetohydrodynamics.^{6,9} A block-based hierarchical data structure is used in conjunction with the finite-volume scheme described above to facilitate automatic solution-directed mesh adaptation on multi-block mesh according to physics-based refinement criteria. “Blocks” here refer to patches of adjacent cells that are refined or coarsened as a group during AMR. The proposed AMR formulation borrows from previous work by Berger and co-workers,^{2,3,5,8} Quirk,^{49,50} and De Zeeuw and Powell⁴ for Cartesian mesh. It is similar to the block-based approaches described by Quirk and Hanebutte,⁴⁹ and Berger and Saltzman.⁸

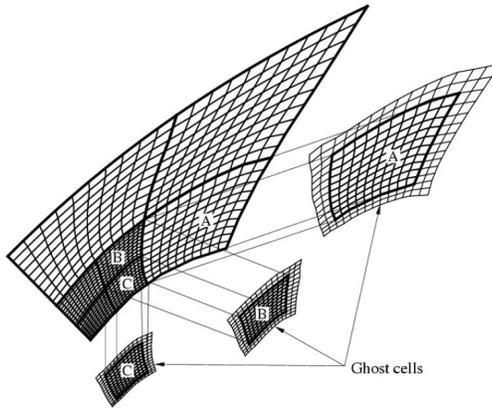


Figure 2: Multi-block quadrilateral mesh in block-based AMR with layers of overlapping ghost cells to facilitate inter-block communication.

Cell-based Cartesian mesh adaptation procedures with more arbitrary quadrilateral and hexagonal mesh have been considered by Davis and Dannenhoffer,⁵¹ and Sun and Takayama.⁵² In comparison to a cell-based approach, block-based AMR may at times be less flexible and efficient. However, a block-based approach involves communications overhead required per block of cells instead of per cell, and less connectivity also results in a simpler data structure. More importantly, the block-based approach most easily lends itself to parallel implementation, and may thus be the preferred alternative especially for large-scale problems.

In the current approach, every block consists of a fixed number of $N_i \times N_j$ cells. Both N_i and N_j must be even, but not necessarily equal. Cell-averaged solution states within each block are stored in an indexed array which corresponds to the physical arrangement of the cells. In addition to these interior cells, there are layers of “ghost” cells which store solution data of interior cells from neighboring blocks, as shown in Figure 2. These overlapping data storage allow the finite-volume calculations on each block to be carried out in a more independent manner. A message passing routine updates the solution in ghost cells as solution in the corresponding interior cells evolve. This may or may not involve restriction or prolongation of the solution depending on the resolution change between adjacent blocks. Message passing procedures will be discussed in more details later in this section. Additional inter-block communication is required to correct interface fluxes computed on coarser neighbors using fluxes from their finer neighbors, so that the flux conservation properties of the finite-volume scheme is strictly enforced across the interface.^{2,3}

The adaptive procedure is directed by one or more refinement criteria for each block. Percentage thresholds are specified by the user, such that blocks with refinement criteria greater than the refinement threshold are flagged to refine, while blocks with criteria lower than the coarsening threshold are flagged to coarsen. To ensure the accuracy of the scheme, mesh refinement is constrained such that the grid resolution changes only by a factor of 2 between any two adjacent blocks, and the minimum resolution is no less than that of the initial mesh. Therefore, a check must be performed prior to refinement to eliminate any cases that violate such constraints. During mesh refinement, each “parent” block is divided into multiple “children” blocks with the same number of cells, thereby increasing the spatial resolution in the region of interest. This process is reversed for over-resolved regions, where multiple “sibling” blocks are coarsened into their parent. Standard multigrid-type restriction and prolongation operators are used to evaluate solution on all blocks resulting from the coarsening or division processes.

Lastly, a hierarchical tree-like data structure is used to keep track of mesh refinement and connectivity between solution blocks. Example of such tree structure is depicted in Figure 4 for anisotropic mesh refinement. Leafs on the trees represent all blocks on the current mesh, while the ancestor nodes and the branches represent the division processes or refinement histories which lead to the current mesh. The root of each tree, therefore, represents a block on the initial mesh. The data structure may contain multiple trees, whose roots are stored in an indexed array data structure. Recursive traversal of the multi-tree structure can be used to determine block connectivity on the physical domain. However, in order to reduce overhead associated with accessing solution information from adjacent blocks, the neighbors of each block are computed and stored after each AMR so that direct access is possible. The hierarchical tree data structure has the advantages in

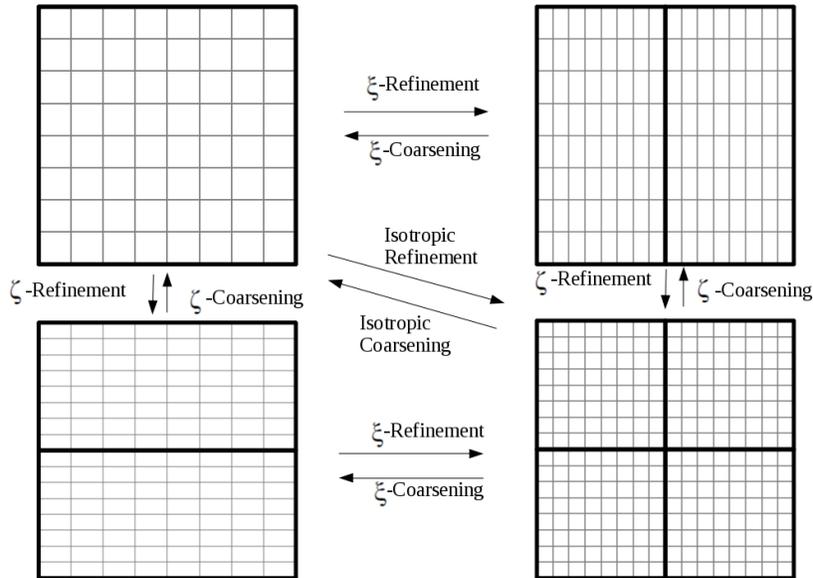


Figure 3: Illustration of refinement and coarsening of an 8×8 block, during (i) anisotropic AMR in ξ , (ii) anisotropic AMR in ζ and (iii) isotropic AMR cell division. Their geometrical relationship are also represented.

readily permitting local mesh refinements. Local modifications to the multi-block mesh can be performed without re-gridding the entire mesh and re-calculating all solution block conductivities.

III.A.1. Anisotropic Adaptive Mesh Refinement

The proposed block-based AMR technique readily permits anisotropic mesh refinement, yet most AMR procedures in the past were carried out in an isotropic manner, where the mesh resolution doubles in all directions. However, this is less flexible, and certainly not the most efficient for flows where solution varies much rapidly in one or more directions than the others. In comparison, the anisotropic AMR technique proposed in this research permits a directional preference in the refinement process. Anisotropic AMR divides each parent block into two children by splitting the parent blocks in either the ξ - or ζ -direction for general body-fitted meshes, as illustrated in Figure 3. As a result, cell resolution doubles in a coordinate direction of choice, but remains the same in the other direction. Implementation of anisotropic AMR follows the general block-based AMR framework discussed above, with some new procedures and data structures to handle the added complexities.

III.B. Hierarchical Tree Data Structure

A flexible binary tree data structure is used to keep track of block connectivity during anisotropic mesh refinement, as compared to a quadtree data structure used in isotropic mesh refinement. Figure 4 provides an example of such binary tree structure. Each branch in the tree may indicate a split in either ξ or ζ . The sequence of “splits” leading to specific ξ and ζ levels is not unique, although the number of splits is unique. This property of the binary tree structure is exploited in a connectivity rearrangement algorithm which allows anisotropic AMR to be more robust.

Ham *et al.*⁴⁰ pointed out a disadvantage of using a structured hierarchical data-structure for anisotropic AMR, in that coarsening is not always permitted unless it exactly reverts the refinement history. This is the case with blocks B and C as represented by the first tree structure in Figure 4, which could not be coarsened in ζ because they are not linked by a ζ -split. This problem is overcome if the first ζ -split which connects the

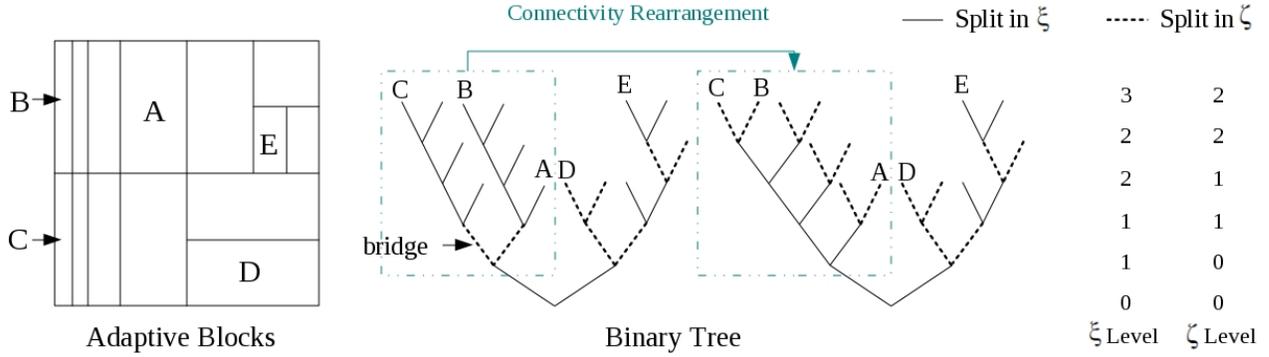


Figure 4: Computational mesh with 6 anisotropic refinement levels originating from one initial block, identically represented by two different binary tree structures.

branches containing blocks B and C, denoted as the “bridge”, could be moved up the tree and made into the immediate linkage between B and C, effectively turning the first tree structure into the second one. This is in essence the goal of connectivity rearrangement. By recursively modifying the internal tree structure, refinement history is equivalently re-written without altering the current mesh.

A new neighbor searching algorithm has also been developed in conjunction with the binary tree structure to compute connectivity of blocks in the physical space. The algorithm follows a similar approach as those described in other literatures^{53,54} and for isotropic AMR.⁵⁵ A common ancestor between the current block and the neighbor is first identified as the “bridge”, which the algorithm crosses to reach the branch containing the neighbor, and subsequently descends to the appropriate leaf node. The bridge is often found as a sibling in the direction of the neighbor. However, there are no corresponding siblings for corner neighbors in anisotropic AMR, in which case the search is carried out in two parts. Take the search of a northwest neighbor for example, the first sibling encountered in either north or west will be treated as an intermediate block, from which a search for a bridge in the remaining direction commences.

III.C. Anisotropic Mesh Refinement Procedures

Anisotropic division and coarsening of blocks in ξ and ζ may be carried out sequentially, but this could lead to the “deadlock” situation illustrated in Figure 5. The deadlock situation is a by-product of connectivity rearrangement, where the refinement history is re-written in such a way that it could not be reversed without exceeding the maximum resolution difference between two neighbors in the intermediate step. The deadlock situation could be avoided by eliminating the intermediate step and simultaneously coarsening in both ξ and ζ . Hence the following anisotropic AMR procedure is proposed:

1. Directional refinement criteria in both ξ and ζ are calculated for all blocks, according to which appropriate blocks are flagged to either refine or coarsen in the respective direction;
2. Conflict-checking procedure is carried out to ensure refinement and coarsening for all flagged blocks are permissible;
3. Connectivity rearrangement is performed to ensure parents of all blocks flagged coarsen are split in the appropriate direction;
4. Reduce all blocks flagged coarsen into their parent blocks;
5. Split all blocks flagged refine in ξ into their west and east children;
6. Split all blocks flagged refine in ζ into their south and north children;
7. Compute and store the new neighbor information;
8. Update geometry and solution information in ghost cells through message passing.

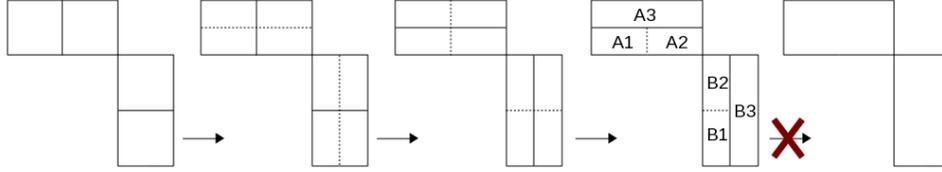


Figure 5: Formation of a deadlock situation from connectivity rearrangement, where blocks A1, A2, and A3 are meant to be coarsened into one block, while B1, B2 and B3 into another, but it would not be permitted if coarsening in ξ and ζ take place sequentially. Block boundaries of siblings are represented by dashed lines.

Note that the constraint of maximum resolution change across block interfaces could be violated before all flagged blocks are refined or coarsened. However, this constraint only needs to be reinforced during message passing at the end. Connectivity rearrangement relies on the use of correct neighbor information, hence it must be performed before refinement or coarsening of any block. Anisotropic AMR could lead to isotropic refinement if a block is flagged to refine in both ξ and ζ . Lastly, the efficiency η of both isotropic AMR and anisotropic AMR is calculated as $1 - N_{cells}/N_{uniform}$, where $N_{uniform}$ is taken as the total number of cells on an isotropic uniform mesh, which consists of cells at the same level as the higher of the maximum ξ - or ζ -level on the current mesh.

III.C.1. Refinement Criteria

For Euler equations for gas-phase flows, isotropic AMR has been directed by refinement criteria based on the magnitude of the density gradient, compressibility and vorticity, which would provide indications of any shock waves, contact surfaces or shear waves present. These measures are scaled by the area of the cell and normalized by the magnitude of the solution variables involved. The refinement criteria for anisotropic AMR must provide a separate indicator for the solution behavior in each direction. Partial derivatives of the primitive solution variables with respect to each coordinate direction would be an intuitive choice. Following a similar scaling and normalization in calculating the isotropic refinement criteria, expressions for the directional refinement criteria would be as follows:

$$\epsilon_{\xi} = \frac{1}{|u|} \left(\vec{\nabla} u \cdot \Delta \tilde{\mathbf{X}} \right), \quad \epsilon_{\zeta} = \frac{1}{|u|} \left(\vec{\nabla} u \cdot \Delta \tilde{\mathbf{Y}} \right). \quad (14)$$

In Equation (14), u could represent any solution variable in general, while $\Delta \tilde{\mathbf{X}}$ is the vector difference between the mid-points of the east and west faces of the cell, and $\Delta \tilde{\mathbf{Y}}$ is the vector difference between the mid-points of the north and south faces of the cell. For body-fitted meshes, the gradient vector $\vec{\nabla} u$ is projected onto the average dimensions of the cell in curvilinear coordinates to indicate the normalized solution change across the cell along each coordinate direction. Therefore, ϵ_{ξ} and ϵ_{ζ} could provide a good indication to the needs of refinement by relating the physics of the solution to spatial resolution in the direction of interest. For the single-variable advection-diffusion equation, Equation (14) simply involves the gradient of the solution u . For Euler equations, gradient of the density is used as the refinement criteria by default unless otherwise specified. Approximations to the gradient vectors are obtained via a second-order unlimited least-squares reconstruction.

III.C.2. Conflict Checking

The conflict checking procedure essentially revises the list of refinement flags to eliminate any cases of refinement or coarsening that is non-permissible. Refinement or coarsening would not be permitted if it introduces resolution changes of more than a factor of two between adjacent blocks. This type of conflict is identified through level checks.

Level checks for ξ and ζ are independent, but are each carried out in a similar manner. By assigning values of “+1”, “0” and “-1” to denote “refine(R)”, “no change(N)” and “coarsen(C)”, level difference resulting from AMR could then be calculated for each direction, such that if the resultant level difference is greater than 1, there is a resolution conflict. There are situations where the conflict could not be resolved unless

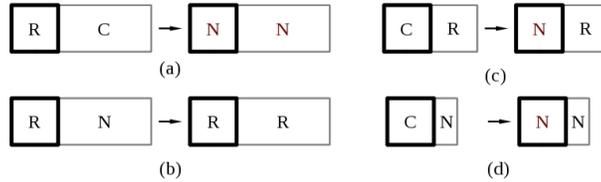


Figure 6: Examples of conflicting scenarios during ξ -refinement. The block with bold boundaries represent the block being checked in relation to its neighbors, while changes resulted from conflict resolution are marked in red.

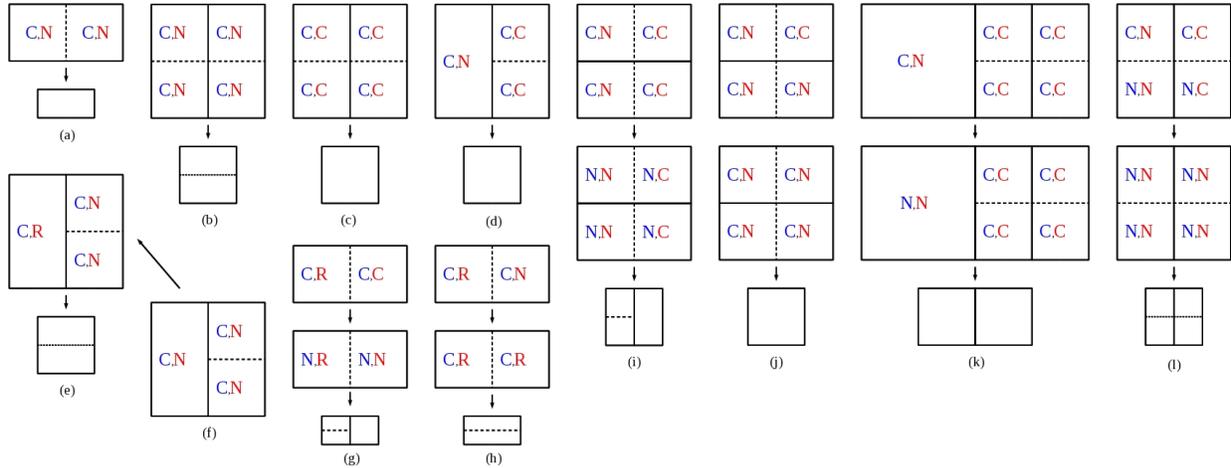


Figure 7: Examples of permitted and non-permitted coarsening scenarios. Flags for the ξ -direction are in blue, and flags for the ζ -direction are in red. Boundary between two sibling blocks are represented by dashed lines. (a-e) ξ -coarsening is permitted without modification of flags; (f, g) ξ -coarsening is permitted by forcing sibling to refine in ζ ; (h, i) ξ -coarsening is not permitted due to conflicts in ζ ; (j) ξ -coarsening is permitted after conflict in ζ is resolved; (k) ξ -coarsening is not permitted because siblings are too fine; (l) Coarsening is not permitted because direction for coarsening is ambiguous.

the flags to both blocks are changed to “N”. They are when the level difference resulting from AMR is 3 as in Figure 6a. This type of conflict is resolved first before any others. Otherwise, when a block flagged “C” conflicts with one of its neighbors as in Figure 6c and 6d, it would be flagged “N” instead; when a block flagged “N” conflicts with a neighbor flagged “R”, the block will be automatically forced to refine as in Figure 6b. This algorithm leads to a more general implementation which avoids counting all possible combinations of resolution changes and refinement flag values, and resolving each scenario individually.

Coarsening in ξ and ζ simultaneously does not only prevent the deadlock situation, it also permits many more coarsening situations as it removes the limitation associated with the sequence of AMR in different directions. Some of the representative scenarios are depicted in Figure 7. The deadlock situation is represented in Case (d), where a ξ -sibling of the coarse block emerges when its finer neighbors are coarsened in ζ , thereby allowing subsequent coarsening in ξ . Case (e) represents a situation that is somewhat opposite, where the east neighbor is first refined in ζ to create two siblings in ξ for the finer neighbors to the west. In order to permit coarsening, the block and its potential “sibling(s)” must both be flagged to coarsen; furthermore, ξ - and ζ -level of any block and its sibling must both be the same at the end of AMR. Checking of all blocks flagged “C” against both of these conditions is another major component of the conflict checking procedure, but it can also be implemented in a general manner. In some cases, both conditions could be satisfied by forcing blocks originally flagged “N” to refine, as in the cases of (h) and (i) in Figure 7.

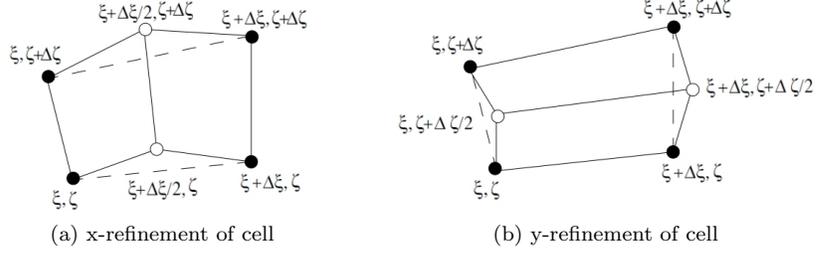


Figure 8: Addition of new nodes during anisotropic cell division using metrics.

III.C.3. Grid Generation and Solution Transfer

$$\begin{aligned}
 \vec{x}(\xi + \Delta\xi, \zeta + \Delta\zeta) &= \vec{x}(\xi, \zeta) + \left. \frac{\partial \vec{x}}{\partial \xi} \right|_{\xi, \zeta} \Delta\xi + \left. \frac{\partial \vec{x}}{\partial \zeta} \right|_{\xi, \zeta} \Delta\zeta \\
 &+ \frac{1}{2} \left(\left. \frac{\partial^2 \vec{x}}{\partial \xi^2} \right|_{\xi, \zeta} (\Delta\xi)^2 + 2 \left. \frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta} \right|_{\xi, \zeta} \Delta\xi \Delta\zeta + \left. \frac{\partial^2 \vec{x}}{\partial \zeta^2} \right|_{\xi, \zeta} (\Delta\zeta)^2 \right) \\
 &+ \mathcal{O}\left((\Delta\xi)^3, (\Delta\zeta)^3\right).
 \end{aligned} \tag{15}$$

Figure 8 illustrates the cell division process in both ξ and ζ , where new nodes are added during refinement. For cells lying on the physical boundaries of body-fitted meshes, locations of new nodes are determined by path-length splines representing the geometrical shapes of the boundaries. New interior nodes could be approximated by the mid-points between existing nodes on the coarse mesh, or using grid metrics. Taylor series expansion in Equation (15) is used to approximate the metrics which maps between physical domain $\vec{x} = (x, y)$ and a uniformly spaced Cartesian computational grid, (ξ, ζ) . Derivative values are computed using second-order central difference approximation, and third-order forward/backward difference at block boundaries. New node locations are then computed by combining the Taylor series expansion from existing nodes on the coarse mesh using a second-order averaging procedure. The mid-point approach is used by default in grid-generation, while the metrics is made available if more accurate grid representation is necessary during accuracy assessments.

Upon cell division, solution average from the coarse cell is transferred onto the two finer cells via direct injection. The cell division process is reversed during coarsening, and area-weighted sum of the finer cells solution average will be assigned to the coarse cell.

III.D. Message Passing

At interfaces between blocks of equal resolutions, solution values are directly assigned from cells to cells. Otherwise, restriction and prolongation procedures are required to evaluate the ghost cell solution values. The restriction procedure used during message passing is identical to what is done during interior cell solution transfer after coarsening. However, the prolongation procedure in message passing requires more accuracy, because the error introduced could not be reduced by solution processes. To preserve the high-order accuracy of the proposed scheme, reconstruction in the coarse interior cell is integrated over, and divided by, areas of the corresponding ghost cells to obtain the appropriate solution averages.

For anisotropic AMR, block resolutions in ξ and ζ are independent, hence a block could be finer than its neighbor in ξ , yet be of the same resolution or coarser in ζ , and vice versa. This results in 9 different types of resolution changes across block interfaces, which are summarized in Figure 9. 3 out of these 9 scenarios are isotropic. Furthermore, Cases (a), (b), (d) and (e) involve prolongation or restriction in one of the two directions, hence the procedures are almost identical to those in isotropic AMR with slight modifications. The two remaining cases involve message passing from blocks which are finer in one direction, but coarser in the other. Hence message passing for these two cases would consist of prolongation in one direction, followed by restriction in the other direction. Take Case (c) for example, the north subcell values of both cell (i, j)

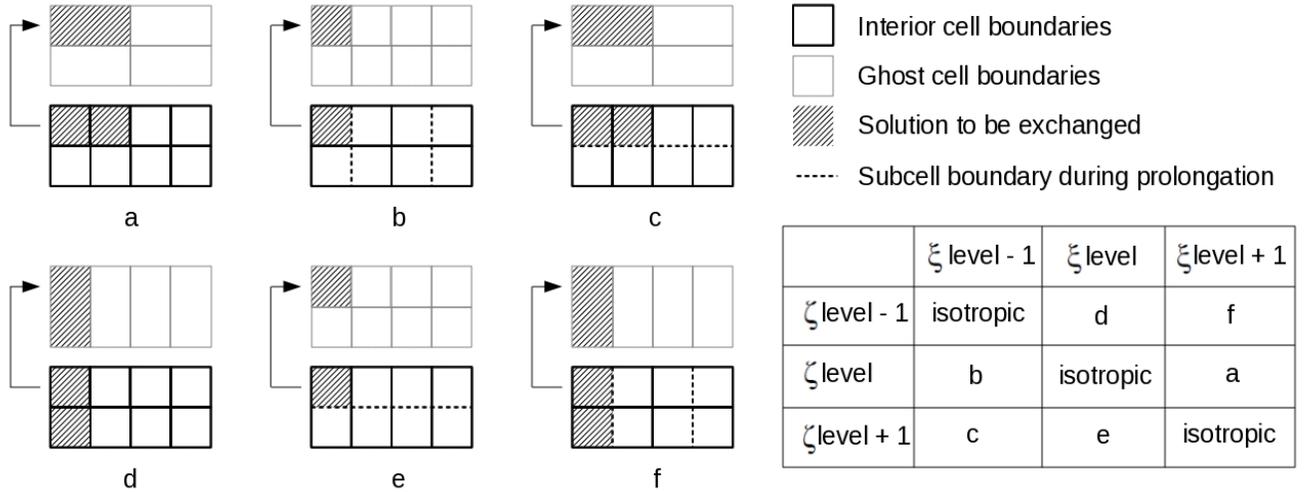


Figure 9: Summary of the different message passing scenarios in anisotropic AMR with graphical illustration of the prolongation/restriction required.

and cell $(i + 1, j)$ are first obtained through prolongation, following which solution in north sectors of both cells are recombined via restriction.

IV. Newton-Krylov-Schwarz Method

Steady-state problems can be very effectively solved by application of Newton’s method to the resulting nonlinear system of algebraic equations, and subsequently a Krylov subspace method, such as the generalized minimal residual (GMRES) algorithm proposed by Saad,⁵⁶ in combination with a domain-based additive Schwarz preconditioning technique⁵⁶ to solve the large, sparse, system of linear equations at each Newton step. In addition, this Newton-Krylov-Schwarz (NKS) approach is well suited to exploiting the potential of distributed-memory multi-processor machines because the Schwarz preconditioner breaks the problem into a set of sub-problems. Rather efficient parallel implementations of implicit algorithms via Schwarz preconditioning have been developed by Keyes and co-researchers and successfully applied to a range of flow problems.^{57–60} Following these ideas, Groth and co-researchers^{61–63} have developed a rather effective and scalable parallel implicit method based on a Jacobian-free inexact NKS approach with additive Schwarz preconditioning and domain partitioning following from the multi-block AMR mesh.

For all steady-state problems of interest, the described NKS method seeks numerical solution which satisfies

$$\mathbf{R}(\mathbf{U}) = 0. \quad (16)$$

This often leads to large coupled non-linear systems of algebraic equations, to which Newton’s method offers a robust and efficient iterative technique to compute the solution. Starting with an initial estimate, \mathbf{U}^0 , successively improved estimates of the solution are obtained by solving

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right)^n \Delta \mathbf{U}^n = \mathbf{J}^n \Delta \mathbf{U}^n = -\mathbf{R}(\mathbf{U}^n), \quad (17)$$

at each step, n , of the Newton method, and an improved approximation of the solution is given by

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta \mathbf{U}^n, \quad (18)$$

where $\mathbf{J} = \partial \mathbf{R} / \partial \mathbf{U}$ is the residual Jacobian. The iterative procedure is repeated until an appropriate norm of the solution residual is sufficiently small, i.e., $\|\mathbf{R}(\mathbf{U}^n)\|_2 < \epsilon \|\mathbf{R}(\mathbf{U}^0)\|_2$ where ϵ is some small tolerance value typically around approximately $10^{-12} - 10^{-10}$.

Each step of the Newton iterations requires the solution of a system of linear equations given by Equation (17). For most practical flow computations, this system is large, sparse, and non-symmetric. Therefore,

the implicit algorithm involved in this paper uses a class of Krylov subspace iterative methods known as GMRES methods, developed by Saad and co-workers,^{56,64–66} which are applied extensively in obtaining the solutions to such large sparse non-symmetric linear equations.^{60,67–71} The GMRES method is used in conjunction with Schwarz preconditioning as the Jacobian matrix is often ill-conditioned. A combination of global and local preconditioning techniques is used, where the global additive Schwarz preconditioner for N solution blocks is defined as follows:

$$\mathbf{M}^{-1} = \sum_{k=1}^{N_{blocks}} \mathbf{B}_k^T \mathbf{M}_k^{-1} \mathbf{B}_k. \quad (19)$$

The gather operator or matrix, \mathbf{B}_k , gathers the solution unknowns for the k^{th} domain from the global solution vector, and \mathbf{M}_k^{-1} is the local block preconditioner for the k^{th} domain. The described block-based anisotropic AMR fits well with the use of Schwarz preconditioning in NKS, and is therefore able to readily enable parallel implementation of the overall Newton method.

Application of GMRES leads to an overall solution algorithm with iterations within iterations: an “inner loop” of iterations to obtain the solution of the linear system, within an “outer loop” of iterations associated with the solution of the nonlinear problem. An inexact Newton method is used in this approach, in which the inner iterations are carried out only until $\|\mathbf{R}^n + \mathbf{J}^n \Delta \mathbf{U}^n\|_2 \leq \zeta \|\mathbf{R}^n\|_2$, where ζ is typically in the range 0.1-0.5.

V. Parallel Implementation

The multi-block quadrilateral mesh and tree data structure lends itself naturally to domain decomposition and enables efficient and scalable implementations of both explicit and implicit solution algorithms on distributed-memory multi-processor architectures.^{10,61} A parallel implementation of the block-based anisotropic AMR scheme has been developed and is used to generate numerical results for both the 2D advection-diffusion problem and the 2D Euler Equations in this paper. The test cases do not benefit most significantly from the use of distributed-memory architectures due to the small scales of the problems. However, they can demonstrate the feasibility of combining the high-order and anisotropic AMR framework with parallel implementation, which can be equally applied to larger scale problems. Existing parallel implementation uses the C++ programming language and the MPI (message passing interface) library.⁷² Domain decomposition is carried out by farming the solution blocks out to the separate processors, with more than one block permitted on each processor. For homogeneous architectures with multiple processors all of equal speed, an effective load balancing is achieved by exploiting the self-similar nature of the solution blocks and simply distributing the blocks equally among the processors. For heterogeneous parallel machines, such as a network of workstations and computational grids, a weighted distribution of the blocks can be adopted to preferentially place more blocks on the faster processors and less blocks on the slower processors. Inter-processor communication is mainly associated with block interfaces and involves the exchange of ghost-cell solution values and conservative flux corrections at every stage of the multi-stage time integration procedure. Message passing of the ghost-cell values and flux corrections is performed in an asynchronous fashion with gathered wait states and message consolidation.

VI. Numerical Results for 2D Advection-Diffusion Equation

VI.A. Advection-Diffusion in a Rectangular Channel

A problem involving both advection and diffusion in a rectangular channel is considered as the first validation case for the presented numerical scheme. The rectangular channel has an inflow at $x = 0$ described by $\sin(\pi y)$. A constant diffusion field with $\kappa(x, y) = 0.01$ smears the solution as a uniform velocity field $\vec{V}(x, y) = (0.1, 0)$ carries the inflow towards the end of the channel. A uniform initial condition of $u = 0$ is imposed at $t = 0$, which allows this case to be examined both as an unsteady and a steady-state problem. The unsteady evolution of the solution will be obtained using explicit time-marching; whereas the steady-state solution alone is separately obtained through the implicit Newton-Krylov-Schwarz method. Both simulations consist of 12×12 blocks, as well as a fourth-order accurate spatial reconstruction (i.e. $k = 4$).

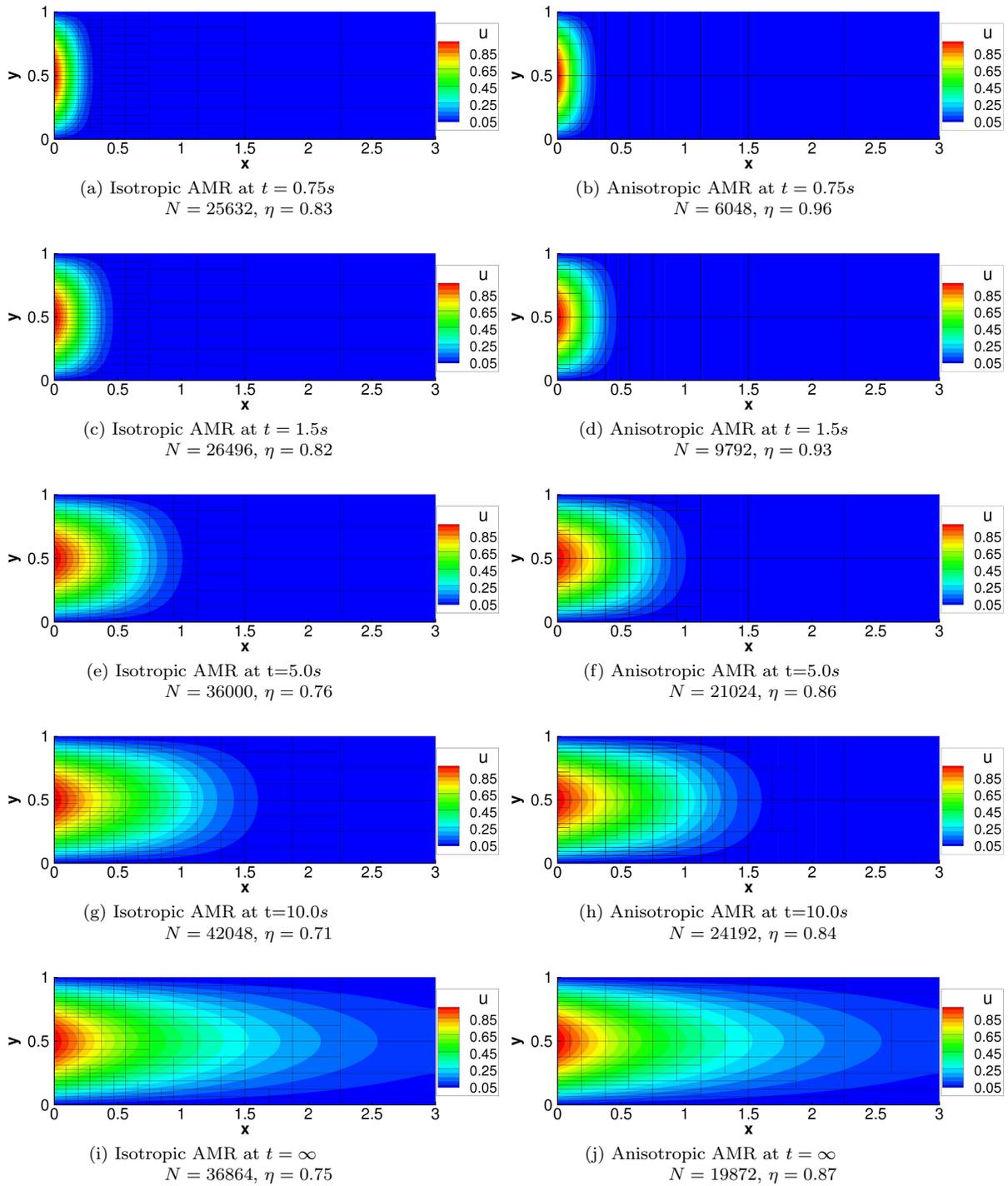


Figure 10: Comparison of isotropic and anisotropic AMR in simulating advection and diffusion in a rectangular channel as a time-variant problem, with CENO $k = 4$ reconstruction.

VI.A.1. Unsteady Results

The mesh for this simulation begins with four blocks, which undergo five levels of initial AMR before the simulation starts. Successive refinements are carried out every 200 time steps, although a higher refinement frequency of 50 is used near the beginning for anisotropic AMR. The CFL number is 0.5. The solution near $t = 0s$ is strongly anisotropic as the inflow creates a sharp gradient in the x -direction, but the variation becomes more gradual as t increases. This change of the solution distribution is well captured by anisotropic AMR as observed in snapshots of the solution at $t = 0.75s$, $t = 1.5s$, $t = 5.0s$, $t = 10.0$ and $t = \infty$ in Figure 10. The refinement and coarsening thresholds are 0.80 and 0.30 for anisotropic AMR, and 0.80 and 0.15 for isotropic AMR.

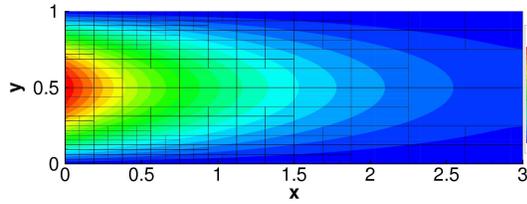
As expected, the advantage of anisotropic AMR is the most noticeable at $t = 0.75s$. Isotropic AMR constrains the aspect ratios of the blocks according to those on the initial mesh, hence the mesh resolution in y is forced to increase with that in x , resulting in 32 blocks near the west boundary of the channel at $x = 0$. In the mesh generated by anisotropic AMR, the same resolution in x is achieved with only 12 blocks in the y direction, whose dimensions in y are flexibly determined by the variation of the solution along the y -axis. Overall, a total of 42 blocks with 6048 cells were used in the anisotropic mesh with a refinement efficiency of 0.96, while a total of 178 blocks with 25632 cells were used in the isotropic mesh with a refinement efficiency of 0.83. Hence anisotropic AMR is able to provide a saving of 76% in terms of computational cost. Although overall variation of the solution becomes more uniform with time, local variations remain somewhat anisotropic, as seen particularly at the north and south boundaries of the rectangular channel near the inflow for $t = 1.5s$, $t = 5.0s$, $t = 10.0s$ and $t = \infty$. For regions in the domain where the gradient in one direction is weaker, the gradient-norm refinement criteria in isotropic AMR reduces the effect of the strong gradient in the other direction. Hence these local anisotropic variations are captured much better by anisotropic AMR.

VI.A.2. Implicit Results

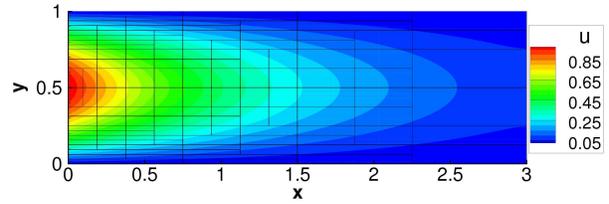
The implicit simulation uses the same initial mesh as the unsteady case, but without the initial AMR. Steady-state solution is first obtained on the coarse mesh using inexact Newton's method, which is then improved by increasing the mesh resolution through AMR and repeating the NKS-AMR cycles until the mesh ceases to change. This speeds up the computation process as Newton's method converges much faster when the solution approaches steady-state. The steady-state solution and history of convergence for both isotropic AMR and anisotropic AMR are shown in Figure 11. The resulting meshes for both AMR are slightly different from those in the explicit simulation, because the explicit and implicit solution procedures lead to very different refinement histories. The NKS method appears to work quite well for this problem, where the residual drops at least 12 orders of magnitude in mostly 8–10 Newton steps for both AMR methods. Furthermore, a maximum of 474 GMRES iterations are required per Newton iteration for isotropic AMR, whereas a maximum of 350 GMRES iterations are required for anisotropic AMR due to a smaller system size overall.

VI.A.3. Accuracy Analysis

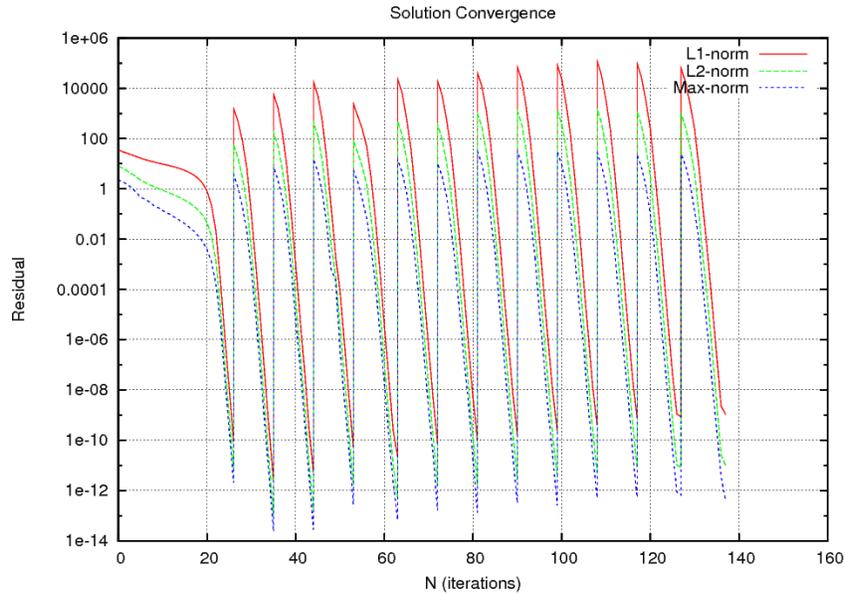
Convergence studies for the advection diffusion problem in a rectangular channel have been included to more accurately reflect the effectiveness of anisotropic AMR as compared to isotropic AMR and uniform refinements. Constrained reconstructions are enforced on all boundaries for this study to maintain high-order accuracy. Result of the convergence studies is shown in Figure 12. Each point in Figure 12 corresponds to the error assessed on meshes obtained either by 2, 3 or 4 uniform refinements, or by applying isotropic and anisotropic AMR with 2, 3, 4 or 5 maximum refinement levels until the mesh ceases to change. Convergence of the uniform mesh in Figure 12 is observed to have a slope of approximately 4 for fourth-order spatial accuracy as expected. However, both AMR methods as observed in this study do not introduce any improvements to the convergence of the uniform mesh. In the contrary, they appear to increase the number of cells required without improving the solution accuracy. This is largely related to the fact that the advection-diffusion in a rectangular channel is relatively smooth and lacking in scale disparity, hence the error distribution is likely uniform throughout the domain and uniform refinements are favored.



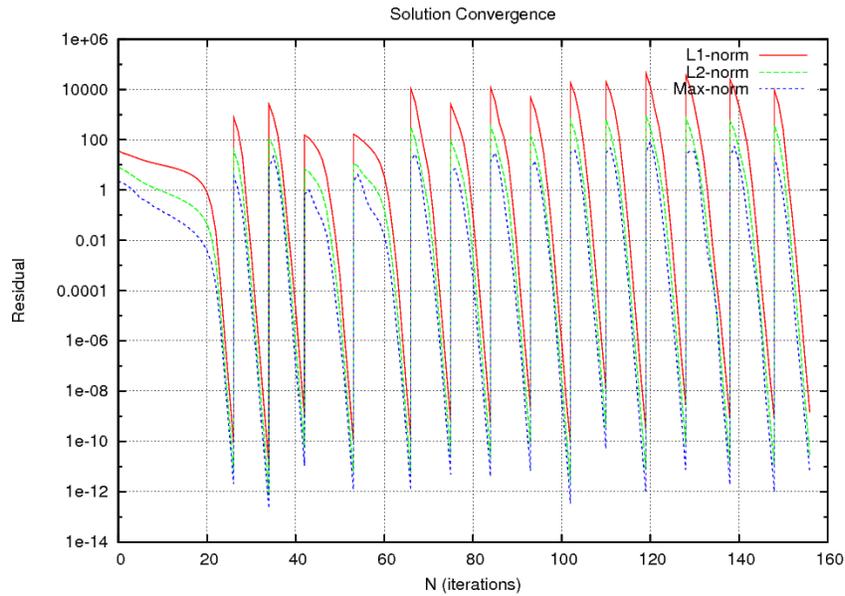
(a) Steady-State solution with isotropic AMR
 $N = 37728$, $\eta = 0.74$



(b) Steady-State solution with anisotropic AMR
 $N = 19584$, $\eta = 0.87$



(c) Convergence History for Isotropic AMR



(d) Convergence History for Anisotropic AMR

Figure 11: Comparison of isotropic and anisotropic AMR in simulating advection and diffusion in a rectangular channel as a steady-state problem, with CENO $k = 4$ reconstruction.

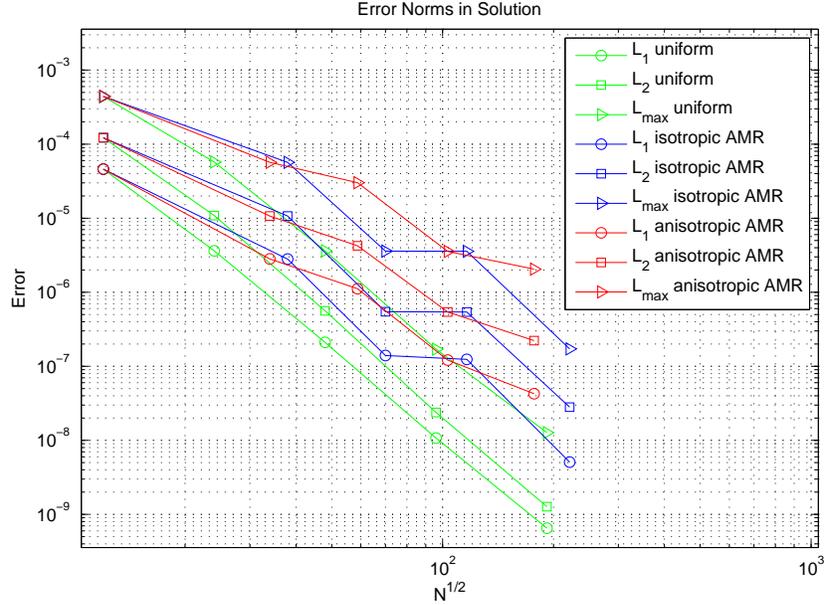


Figure 12: Convergence comparison for anisotropic AMR against isotropic AMR and uniform refinements in simulating of advection-diffusion in rectangular channel using $k = 4$ CENO reconstruction.

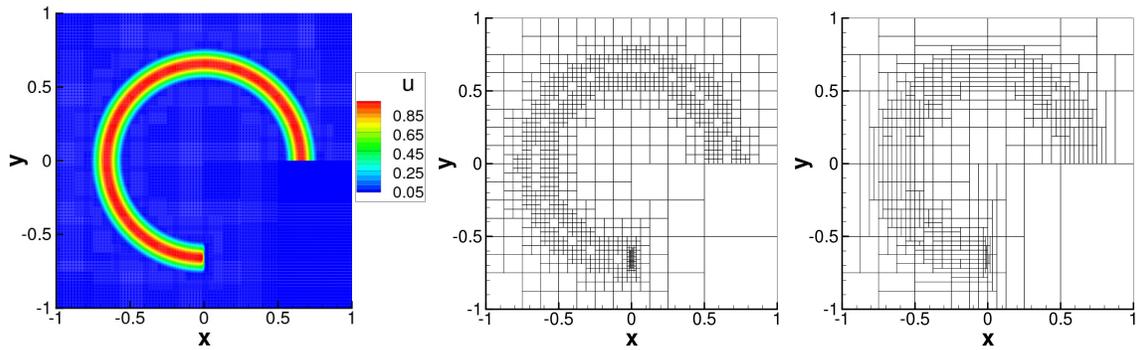
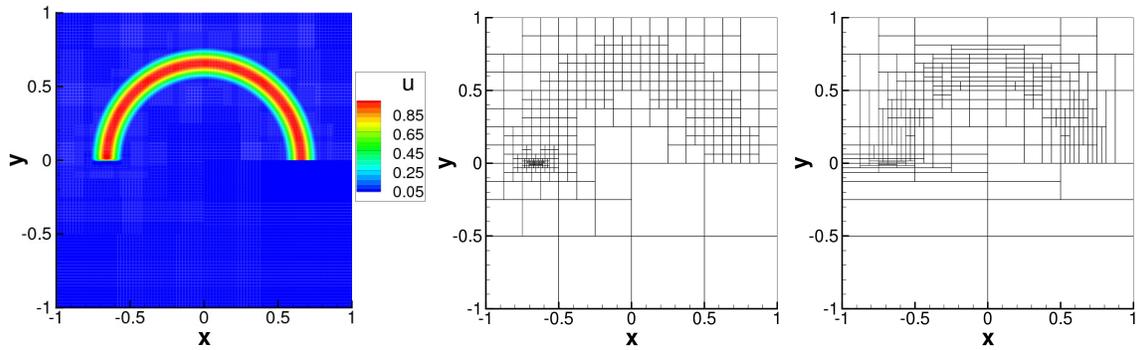
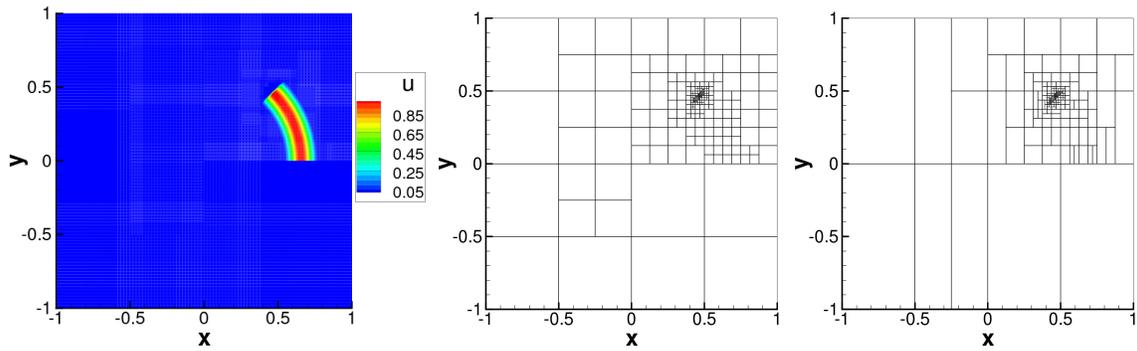
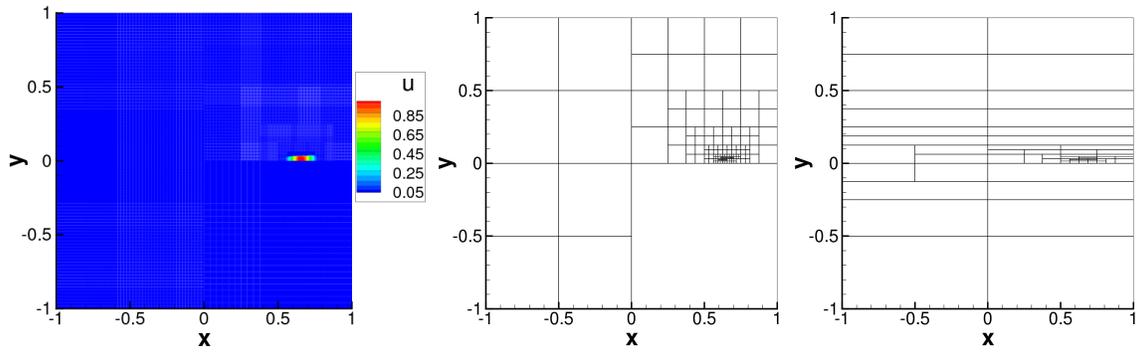
VI.B. Circular Advection Problem

The second problem considered for the 2D advection-diffusion equation involves pure advection by a constant angular velocity of 5 rad/s. The circular advection takes place within a square box with initially four 20×20 blocks, where the block interface along $y = 0$ and $0 \leq x \leq 1$ represents an inflow/outflow boundary. The inflow variation is defined by $\sin^2(4\pi(r - 0.375))$ for $r \in [0.375, 0.625]$ and 0 otherwise. A uniform initial condition of $u = 0$ is again imposed, which creates a discontinuous flow front starting at $\theta = 0$. The flow front makes an angular displacement of 2π before the solution connects at the inflow/outflow boundary to form a continuous ring of circular flow. Anisotropic AMR is expected to work best when the direction of solution variation is aligned with the coordinate directions. In the extreme case when the solution varies gradient is exactly 45 degrees from the main coordinate axis, anisotropic AMR simply reverts to isotropic AMR. Hence it would be interesting to observe how anisotropic AMR attempts to capture the solution for this problem as it goes through all possible angles of variation.

VI.B.1. Unsteady Results

Snapshots of the time-variant results have been plotted in Figure 13. The CFL number is again 0.5. To capture the discontinuous wave front, both isotropic and anisotropic AMR are allowed a maximum of 8 refinement levels through 2 levels of uniform refinements and 6 levels of initial AMR, followed by subsequent refinements every 10 time steps. Refinement threshold is 0.80 for isotropic AMR, and 0.60 for anisotropic AMR to encourage x -refinements near $t = 0$ of the simulation. Coarsening threshold is 0.20 for both AMR methods.

At $t = 0.01s$, the sharp gradient created by the discontinuous flow front is approximately aligned with the y -axis, again allowing anisotropic AMR to be very advantageous. Although both AMR methods have high efficiency of almost 1, anisotropic AMR is able to introduce an additional 65% saving when compared to isotropic AMR, as shown in Figure 13b and Figure 13c. The next snapshot occurs at $t = 0.157s$, which is when the wave front is 45 degrees from both axis. As anticipated, the mesh produced by anisotropic AMR near the flow front is very similar to that by isotropic AMR. Anisotropic AMR is still observed to offer some advantage in optimizing the number of blocks required in regions away from the wave front, which are refined to satisfy the constraint of maximum resolution change across block boundaries. In the next two snapshots,



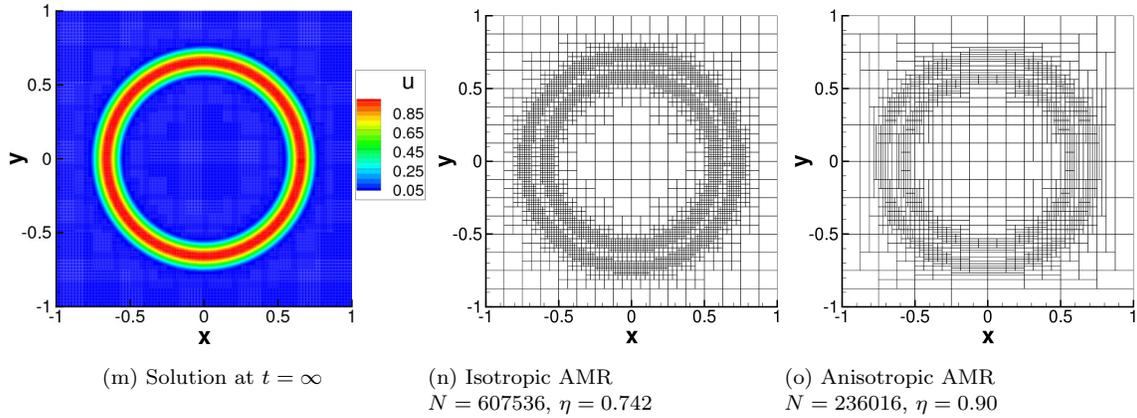


Figure 13: Comparison of isotropic and anisotropic AMR in simulating circular advection in a rectangular circular as a time-variant problem, with CENO $k = 3$ reconstruction.

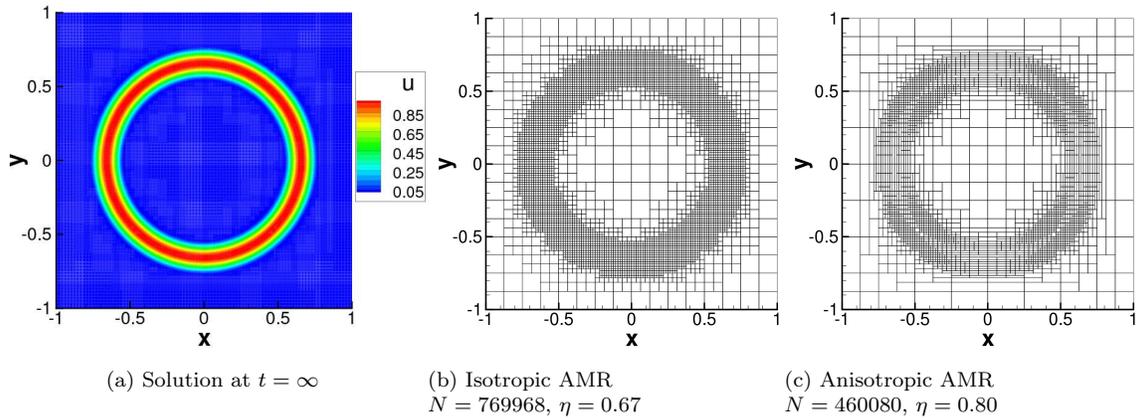


Figure 14: Comparison of isotropic and anisotropic AMR in simulating circular advection in a rectangular box as a steady-state problem, with CENO $k = 3$ reconstruction.

the wave front wave is again aligned with the coordinate axis. This transition is well captured by anisotropic AMR, which also reflects the effectiveness of mesh coarsening after each AMR.

After the wave front has reached the outflow boundary at $t = 1.26s$, the maximum level of refinement is reduced to 7 to avoid over refinement. The refinement threshold for both isotropic and anisotropic AMR is increased to 0.90. Coarsening threshold increased to 0.30 for anisotropic AMR and reduced to 0.05 for isotropic AMR. AMR frequency is reduced to 100 time steps as the solution is time-marched towards steady-state. In the final mesh depicted in Figure 13n and Figure 13o, anisotropic AMR places blocks with higher aspect ratios at locations where the solution variation is more aligned with the coordinate axis, and blocks with lower aspect ratios close to the 45 degree lines. The mesh resolution produced is for the most part similar to that of isotropic AMR in all directions of variations.

VI.B.2. Implicit Results

The steady-state solution for the circular advection problem is obtained following a similar approach as described in Section VI.A.2. The coarse-mesh steady-state solution is obtained on 64 blocks resulting from 2 uniform refinements on the initial mesh. Both isotropic and anisotropic AMR are allowed a maximum refinement level of 7, with a refinement threshold to 0.40 and a coarsening threshold to 0.10. The final steady-state solution is plotted in Figure 14. The mesh for both AMR methods are denser than in the

explicit results, again due to the very different refinement histories from the explicit and implicit procedures. However, distributions of the mesh blocks follow a similar trend.

VI.B.3. Accuracy Analysis

Accuracy assessments have been performed on the steady-state meshes for both the explicit and the implicit results, where the L_1 , L_2 and L_{max} norms of the solution error are summarized in Table 1. When examining the explicit results alone, anisotropic AMR appears less effective in reducing error in the solution, and the case is similar with the implicit results alone. However, if the anisotropic mesh from the implicit result is compared with the isotropic mesh from the explicit result, then it becomes evident that anisotropic AMR does in fact improve the solution accuracy while maintaining a significantly lower cell-count. It is important to note that this problem is not at all anisotropic, and that anisotropic AMR uses less than half of the computational cells in isotropic AMR with similar refinement and coarsening thresholds specified. Therefore, if the difference in cell count could be used towards increasing both the maximum refinement level and refinement thresholds for anisotropic AMR, it is believed that anisotropic AMR will likely produce more accurate results than isotropic AMR.

VII. Numerical Results for 2D Euler Equations

VII.A. Shock-Box Simulation

The shockbox problem is an important application of the 2D Euler Equations for inviscid compressible gases. It is also strongly anisotropic, making it an excellent practical application of the proposed high-order finite-volume scheme with anisotropic AMR. The shockbox problem considered herein consists of air initially rest. The left state, initially occupying $x < 0$ and $y < 0$, is at standard atmospheric condition (i.e. $p = 101.325kPa$, $\rho = 1.225kg/m^3$); and the right state, initially occupying $x \geq 0$ and $y \geq 0$, is at four times the standard atmospheric condition (i.e. $p = 405.300kPa$, $\rho = 4.9kg/m^3$). Reflection boundary conditions are imposed on all four sides of the domain. Time-accurate solution for the unsteady simulation is obtained via the fourth-order Runge-Kutta time-marching method, with a CFL number of 0.25. Flux evaluation at cell boundaries makes use of the Roe approximate Riemann solver, combined with a fourth-order space-accurate reconstruction (i.e. $k = 3$). To fully capture the discontinuity in the initial states, the simulation begins by performing 1 uniform refinements and 7 initial AMR on a single block consisting of 10×10 cells. The mesh is updated once every 20 time steps using both isotropic and anisotropic AMR.

The required mesh resolution for the shockbox problem, especially near the start of the simulation at $t = 0.045s$, is driven by the rapid solution change at the discontinuities, which could be satisfied by refining the mesh only in the direction of wave propagation along either x or y . As a result, the performance of isotropic AMR becomes relatively poor by unnecessarily increasing the mesh resolution along the discontinuities, where the solution changes are minimal. In comparison, anisotropic AMR effectively achieves the same mesh resolution as isotropic AMR across all discontinuities, while keeping the mesh coarse in the other direction. In doing so, anisotropic AMR is able to introduce a significant 87% reduction in the number of computational cells.

For the later snapshots at $t = 0.145s$, $t = 0.3s$ and $t = 0.745s$, the interference pattern produced by intersecting waves become more dominant as time increases. Discontinuities in the interference region are not aligned with the x or y axis, in a similar manner as in the circular advection problem, where anisotropic AMR is observed to revert to isotropic AMR. Therefore, the computational saving from anisotropic AMR at $t = 0.745s$ is 47%, much lower than that at $t = 0.045s$, but still quite significant. A detailed accuracy analysis could not be performed due to the lack of an exact solution. However, by the observation that anisotropic

Table 1: Accuracy Comparison for Circular Advection Simulation

AMR	Explicit Results in Figure 13				Implicit Results in Figure 14			
	N	L_1	L_2	L_{max}	N	L_1	L_2	L_{max}
Isotropic	607536	1.5×10^{-4}	7.26×10^{-4}	7.81×10^{-3}	769968	8.14×10^{-6}	4.62×10^{-5}	6.67×10^{-4}
Anisotropic	236016	2.66×10^{-4}	1.21×10^{-3}	1.17×10^{-2}	460080	5.01×10^{-5}	2.6×10^{-4}	3.77×10^{-3}

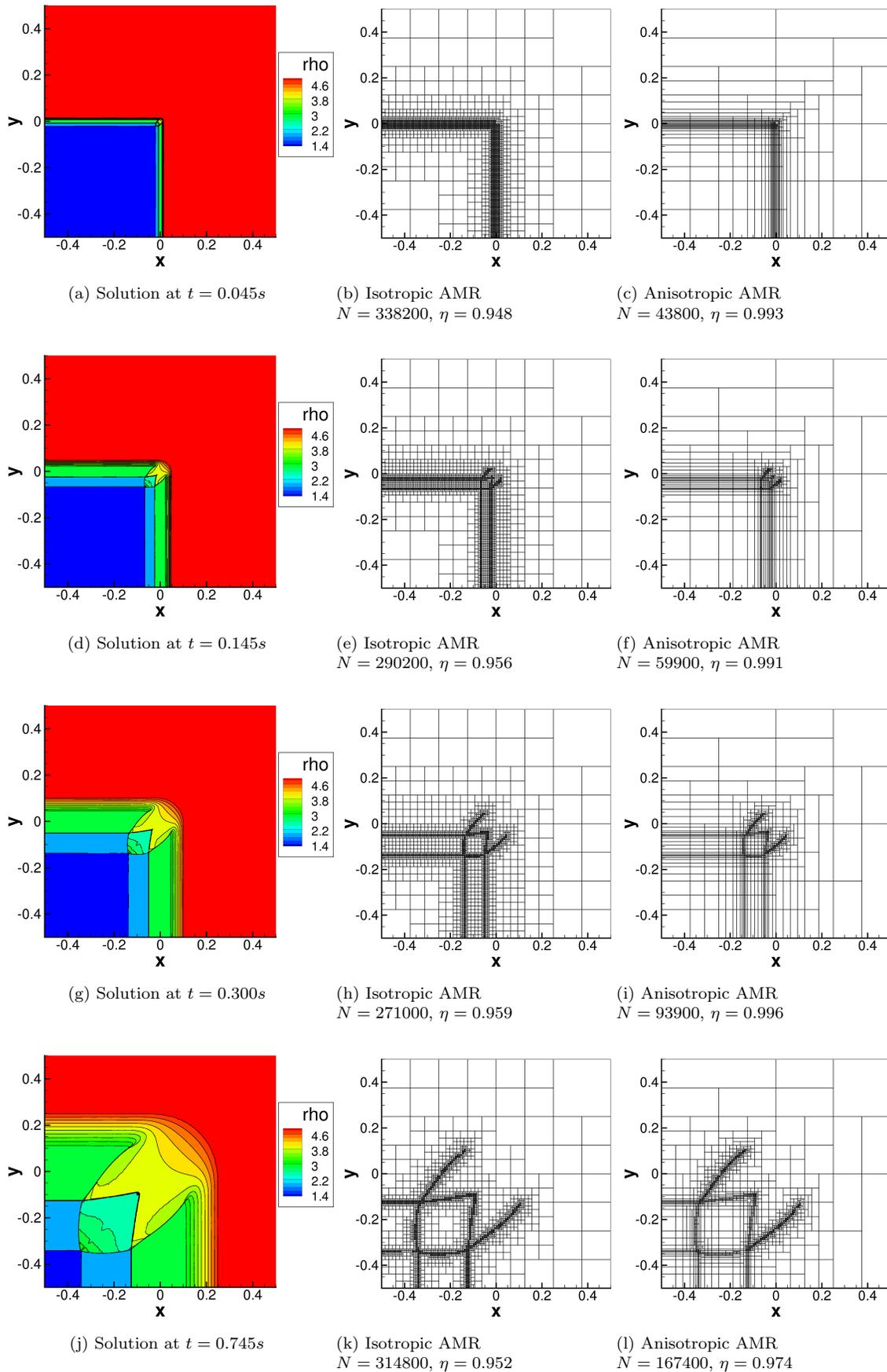


Figure 15: Comparison of isotropic and anisotropic AMR in simulating the unsteady shock-box problem, with $k = 3$ CENO reconstruction.

AMR achieves similar spatial resolution as isotropic AMR in critical regions with most continuities and rapid solution changes, it could be concluded that the accuracy of anisotropic AMR and isotropic AMR are likely on par.

VII.B. Supersonic Flow Past a Cylinder

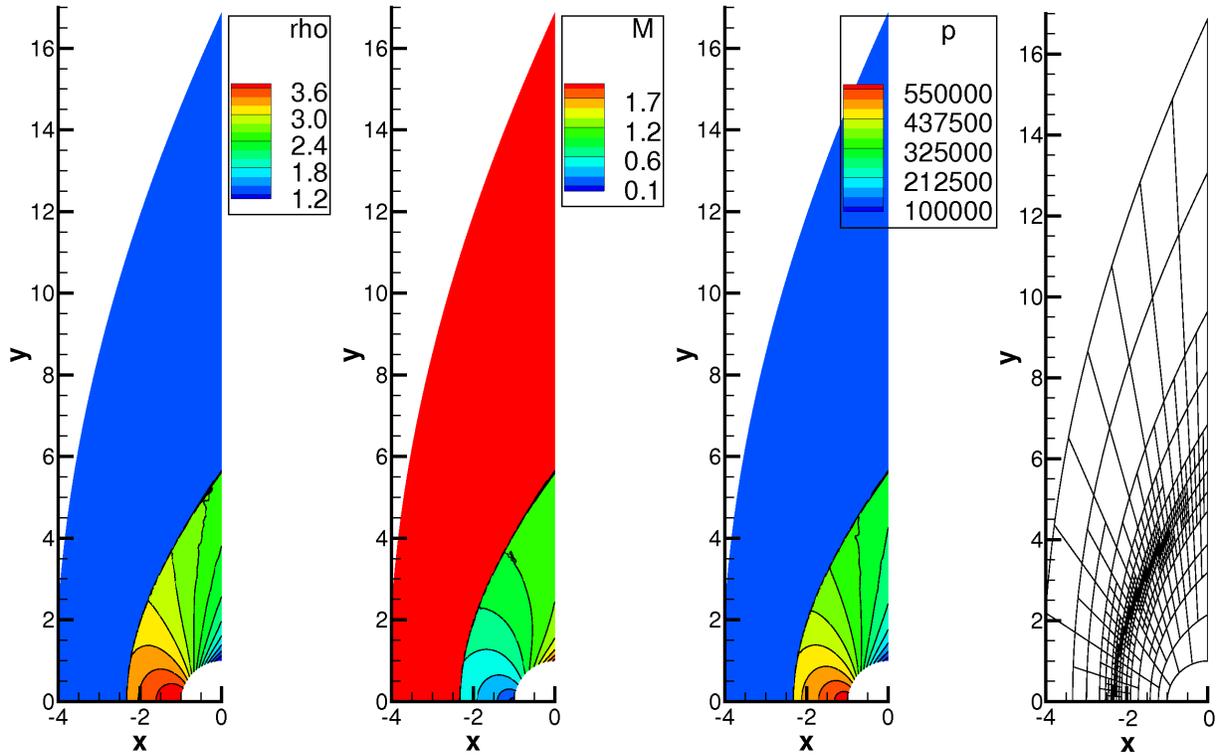
The second simulation considered for the Euler equations involves supersonic flow past a bluff body, which causes a bow shock to form at a distance away from the bluff body surface. Directly in front of the bluff body, the supersonic freestream flow becomes subsonic behind the shock and the flow comes to a full stop at the stagnation point. The flow behind the shock elsewhere could be supersonic depending on the angle at which the freestream flow crosses the bow shock. Due to the absence of viscous effects in the Euler equations, the flow velocity along the bluff body surface away from the stagnation point is non-zero. The bluff body simulation examined here involves freestream flow at standard atmospheric pressure and density (i.e $p = 101.325kPa$ and $\rho = 1.225kg/m^3$), with a Mach number of 2. The bluff body is a 2D cylinder with radius $r = 1$, and with its axis of symmetry perpendicular to the freestream flow. The simulation is carried out on a body-fitted mesh with the south boundary representing the surface of the bluff body and a curved north boundary shaped according to an estimated bluff body Mach number of 1.5. A fixed boundary condition was imposed at the north boundary for the freestream inflow; a reflection boundary condition at the south and west boundaries for the subsonic regions; and a boundary condition with constant extrapolation at the east boundary for the outflow. A $k = 3$ CENO reconstruction is used for this simulation. High-order boundary treatment is reinforced by representing curved boundaries using splines, as well as by applying constraint reconstruction at the south boundary. The Rusanov flux evaluation is used for added dissipation.

The initial mesh consists of a single block with 20×20 cells, which undergoes 1 uniform refinement before the simulation begins. Grid generation for this problem uses the mid-point approach. An implicit method similar to that used in Section VI.A.2 is applied to obtain the steady-state results shown in Figure 16. Convergence history of this study was plotted in Figure 17. Although accuracy assessment is again not available due to the lack of an analytical solution, the predicted shape and location of the bow shock, as well as the distribution of the density, Mach number and pressure in the subsonic region as shown in Figure 16 all appear very similar for isotropic and anisotropic AMR. Due to the setup of the initial mesh, the bow shock is not perfectly aligned with the grid lines and anisotropic AMR did not refine solely in one coordinate direction. Nevertheless, by optimizing the number of blocks along the bow shock, the application of anisotropic AMR has prevented coarser blocks in the more uniform regions of the domain from refining due to resolution conflicts. Hence anisotropic AMR still introduces a significant 70% reduction in the number of cells.

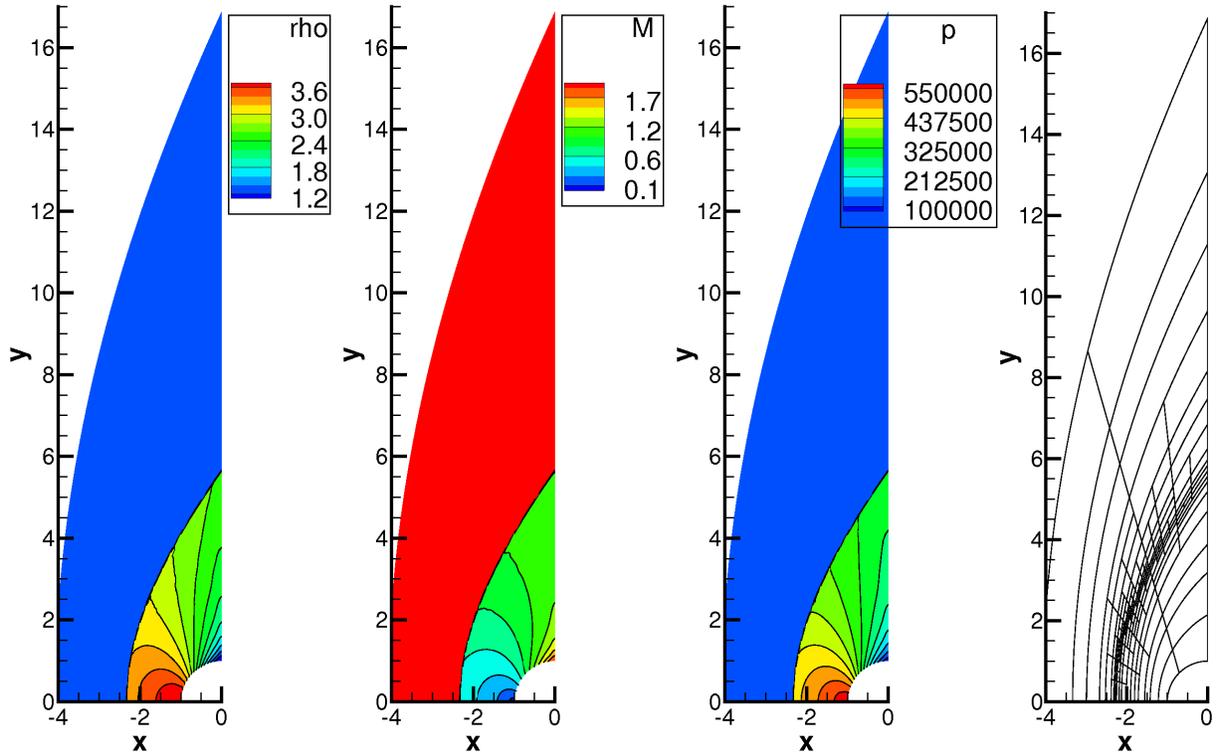
VII.C. Ringleb Flow

The last numerical example considered for this paper involves the simulation of Ringleb's flow. Ringleb's flow consists of isentropic and irrotational flow expanding between two streamlines, where analytical solutions are available. Therefore, this serves as an excellent test case where accuracy of the proposed high-order block-based anisotropic AMR scheme could be assessed and compared. Unlike the other two cases considered for the Euler equations, Ringleb's flow is much smoother in the solution distribution. Therefore, a transonic variant of Ringleb's flow has been chosen for this study, where the supersonic region near the southwest corner of the domain is expected to require higher mesh resolution. The domain for this problem is defined by streamlines corresponding to $k_{min} = 0.5$ and $k_{max} = 1.2$, and subsonic inflow at the north boundary corresponding to $q = 0.3$. Fourth-order accurate space reconstruction is used, with exact solution imposed as boundary conditions on all boundaries. Grid generation makes use of the grid metrics to ensure that the quality of the mesh does not affect accuracy of the solution. Lastly, refinement criteria for this problem involve gradients of the pressure field, such that variations in the supersonic region could be emphasized.

Result of the convergence study has been plotted in Figure 18. All accuracy assessment studies begin with an initial mesh consisting of one 10×10 block. Solution accuracy with 2, 3, 4 and 5 levels of uniform refinements are again included as a reference, which show a fourth-order convergence for both density and pressure as anticipated. Accuracy assessments with the use of isotropic and anisotropic AMR are performed on the converged meshes with maximum refinement levels set to 2, 3, 4, 5 and 6 respectively. Steady-state solutions are obtained using a multi-stage method with optimal smoothing. Both AMR methods are observed to improve the convergence quite significantly up to 5 levels of refinement, with anisotropic AMR being more superior. Upon reaching 5 levels of refinements, AMR ceases to provide any advantage as the

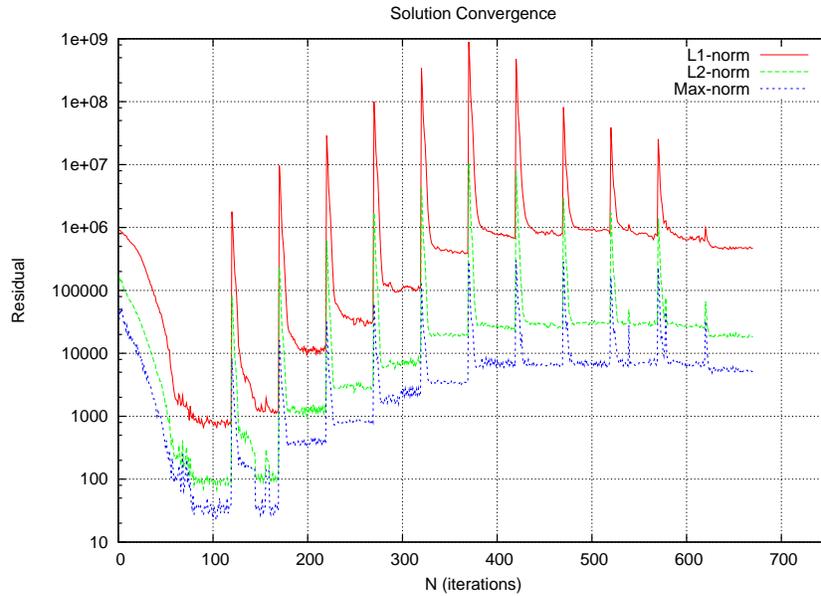


(a) Density, Mach number and pressure distribution with mesh using isotropic AMR: $N = 242800$, $\eta = 0.963$.

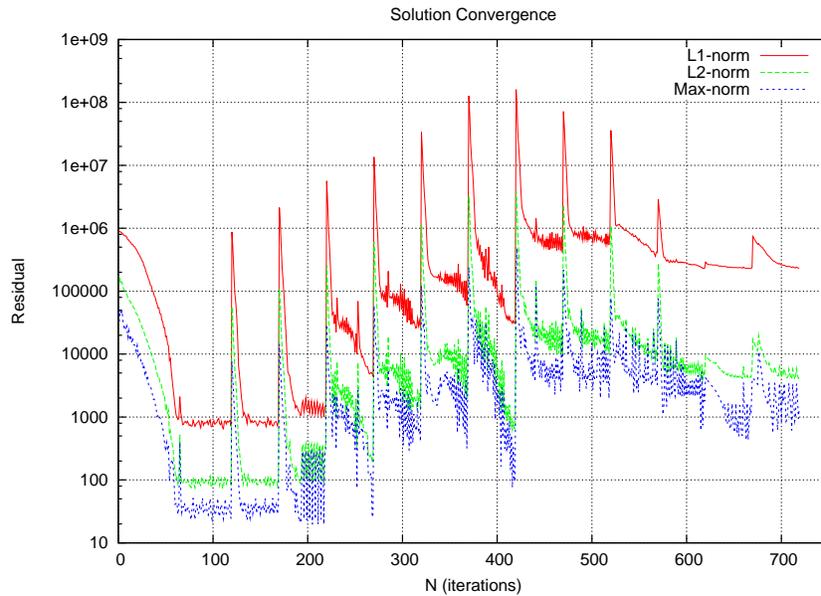


(b) Density, Mach number and pressure distribution with mesh using anisotropic AMR: $N = 74000$, $\eta = 0.989$.

Figure 16: Steady-state simulation of Mach 2 flow past circular cylinder with radius $r = 1$, with CENO $k = 3$ reconstruction.



(a) Convergence history with isotropic AMR.



(b) Convergence history with anisotropic AMR.

Figure 17: Steady-state simulation of Mach 2 flow past circular cylinder with radius $r = 1$, with CENO $k = 3$ reconstruction.

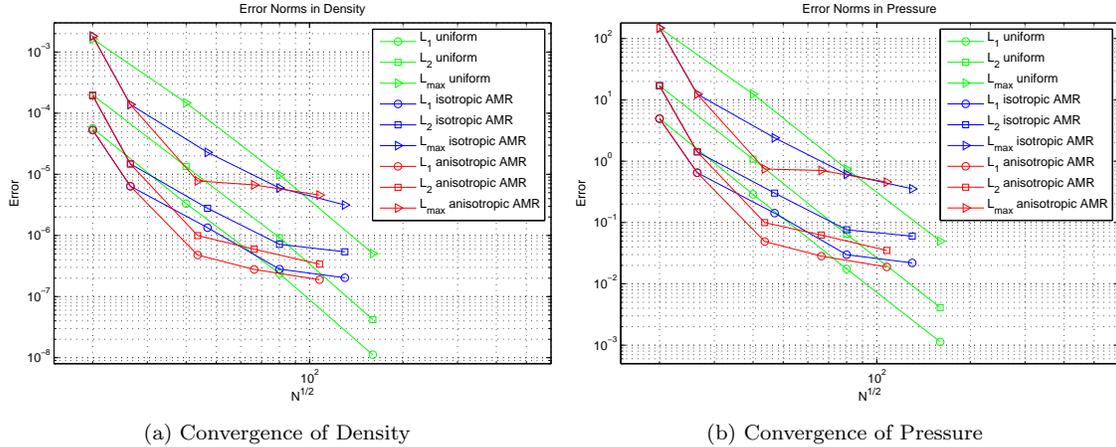


Figure 18: Convergence comparison for anisotropic AMR against isotropic AMR and a uniformly refined mesh in simulating of Ringleb's flow using $k = 3$ CENO reconstruction.

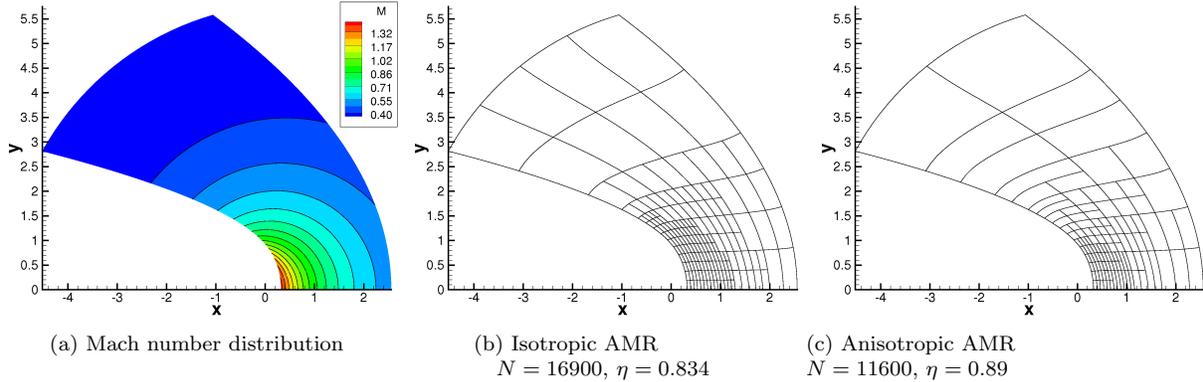


Figure 19: Mach number distribution for a transonic Ringleb's flow, shown with the finest grids from isotropic and anisotropic AMR as used in the convergence studies.

error becomes more distributed throughout the domain. Meshes resulting from both AMR methods with a maximum refinement level of 6 are shown in Figure 19.

VIII. Conclusion

In summary, this study describes in some details a novel parallel, high-order, anisotropic, block-based, AMR, finite-volume scheme. Its application to both steady and unsteady problems described by the advection-diffusion equation as well as the Euler equations have also been examined. It has been shown that the proposed scheme has great potential in reducing the computational costs associated with accurately resolving numerical solution of physically complex flow problems having disparate spatial and temporal scales, especially ones which exhibit strong anisotropic features. Meanwhile, it preserves many advantages of the previous block-based isotropic AMR technique, in particular its parallel performance. Future research will involve the development of more sophisticated directional refinement criteria based on error estimation techniques, the extension of the proposed algorithm to multi-dimensional problems, and finally applications of this technique in more challenging flow problems described in the introduction of this paper.

Acknowledgments

Computational resources for performing the calculations reported herein were provided by the SciNet High Performance Computing Consortium at the University of Toronto and Compute/Calcul Canada through funding from the Canada Foundation for Innovation (CFI) and the Province of Ontario, Canada.

References

- ¹Keyes, D. E., “A Science-Based Case for Large-Scale Simulation Volume 2,” Report prepared for the Office of Science, U. S. Department of Energy, September 2004.
- ²Berger, M. J. and Olinger, J., “Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,” *Journal of Computational Physics*, Vol. 53, 1984, pp. 484–512.
- ³Berger, M. J. and Colella, P., “Local Adaptive Mesh Refinement for Shock Hydrodynamics,” *Journal of Computational Physics*, Vol. 82, 1989, pp. 67–84.
- ⁴De Zeeuw, D. and Powell, K. G., “An Adaptively Refined Cartesian Mesh Solver for the Euler Equations,” *Journal of Computational Physics*, Vol. 104, 1993, pp. 56–68.
- ⁵Aftomis, M. J., Berger, M. J., and Melton, J. E., “Robust and Efficient Cartesian Mesh Generation for Component-Base Geometry,” *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952–960.
- ⁶Groth, C. P. T., Zeeuw, D. L. D., Powell, K. G., Gombosi, T. I., and Stout, Q. F., “A Parallel Solution-Adaptive Scheme for Ideal Magnetohydrodynamics,” Paper 99-3273, AIAA, June 1999.
- ⁷Aftomis, M. J., Berger, M. J., and Adomavicius, G., “A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries,” Paper 2000-0808, AIAA, January 2000.
- ⁸Berger, M. J. and Saltzman, J. S., “AMR on the CM-2,” *Applied Numerical Mathematics*, Vol. 14, 1994, pp. 239–253.
- ⁹Groth, C. P. T., De Zeeuw, D. L., Gombosi, T. I., and Powell, K. G., “Global Three-Dimensional MHD Simulation of a Space Weather Event: CME Formation, Interplanetary Propagation, and Interaction with the Magnetosphere,” *Journal of Geophysical Research*, Vol. 105, No. A11, 2000, pp. 25,053–25,078.
- ¹⁰Sachdev, J., Groth, C., and Gottlieb, J., “A Parallel Solution-Adaptive Scheme for Predicting Multi-Phase Core Flows in Solid Propellant Rocket Motors,” *International Journal of Computational Fluid Dynamics*, Vol. 19, No. 2, 2005, pp. 159–177.
- ¹¹Northrup, S. A. and Groth, C. P. T., “Solution of Laminar Diffusion Flames Using a Parallel Adaptive Mesh Refinement Algorithm,” Paper 2005-0547, AIAA, January 2005.
- ¹²Gao, X. and Groth, C. P. T., “Parallel Adaptive Mesh Refinement Scheme for Turbulent Non-Premixed Combusting Flow Prediction,” Paper 2006-1448, AIAA, January 2006.
- ¹³Gao, X. and Groth, C. P. T., “A Parallel Adaptive Mesh Refinement Algorithm for Predicting Turbulent Non-Premixed Combusting Flows,” *International Journal of Computational Fluid Dynamics*, Vol. 20, No. 5, 2006, pp. 349–357.
- ¹⁴Gao, X. and Groth, C. P. T., “Parallel Adaptive Mesh Refinement Scheme for Three-Dimensional Turbulent Non-Premixed Combustion,” Paper 2008-1017, AIAA, January 2008.
- ¹⁵Gao, X., *A Parallel Solution-Adaptive Method for Turbulent Non-Premixed Combusting Flows*, Ph.D. thesis, University of Toronto, August 2008.
- ¹⁶Gao, X. and Groth, C. P. T., “A Parallel Solution-Adaptive Method for Three-Dimensional Turbulent Non-Premixed Combusting Flows,” *Journal of Computational Physics*, Vol. 229, No. 5, 2010, pp. 3250–3275.
- ¹⁷Harten, A., Enquist, B., Osher, S., and Chakravarthy, S. R., “Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III,” *Journal of Computational Physics*, Vol. 71, 1987, pp. 231–303.
- ¹⁸Barth, T. J., “Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes,” Paper 93-0668, AIAA, January 1993.
- ¹⁹Abgrall, R., “On Essentially Non-Oscillatory Schemes on Unstructured Meshes: Analysis and Implementation,” *Journal of Computational Physics*, Vol. 114, 1994, pp. 45–58.
- ²⁰Sonar, T., “On the Construction of Essentially Non-Oscillatory Finite-Volume Approximations to Hyperbolic Conservation Laws on General Triangulations: Polynomial Recovery, Accuracy and Stencil Selection,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 140, 1997, pp. 157–181.
- ²¹Ollivier-Gooch, C. F., “Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction,” *Journal of Computational Physics*, Vol. 133, 1997, pp. 6–17.
- ²²Jiang, G.-S. and Shu, C.-W., “Efficient Implementation of Weighted ENO Schemes,” *Journal of Computational Physics*, Vol. 126, 1996, pp. 202–228.
- ²³Stanescu, D. and Habashi, W. G., “Essentially Nonoscillatory Euler Solutions on Unstructured Meshes Using Extrapolation,” *AIAA Journal*, Vol. 36, 1998, pp. 1413–1416.
- ²⁴Friedrich, O., “Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids,” *Journal of Computational Physics*, Vol. 144, 1998, pp. 194–212.
- ²⁵Hu, C. and Shu, C.-W., “Weighted Essentially Non-Oscillatory Schemes on Triangular Meshes,” *Journal of Computational Physics*, Vol. 150, 1999, pp. 97–127.
- ²⁶Ollivier-Gooch, C. F. and Altena, M. V., “A High-Order Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation,” *Journal of Computational Physics*, Vol. 181, No. 2, 2002, pp. 729–752.
- ²⁷Cockburn, B. and Shu, C.-W., “TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite-Element Method for Conservation Laws II: General Framework,” *Mathematics of Computation*, Vol. 52, 1989, pp. 411.
- ²⁸Cockburn, B., Hou, S., and Shu, C.-W., “TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite-Element Method for Conservation Laws IV: The Multidimensional Case,” *Journal of Computational Physics*, Vol. 54, 1990, pp. 545.

- ²⁹Hartmann, R. and Houston, P., “Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations,” *Journal of Computational Physics*, Vol. 183, 2002, pp. 508–532.
- ³⁰Luo, H., Baum, J. D., and Löhner, R., “A Hermite WENO-Based Limiter for Discontinuous Galerkin Method on Unstructured Grids,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 686–713.
- ³¹Gassner, G., Lörcher, F., and Munz, C.-D., “A Contribution to the Construction of Diffusion Fluxes for Finite Volume and Discontinuous Galerkin Schemes,” *Journal of Computational Physics*, Vol. 224, 2007, pp. 1049–1063.
- ³²Wang, Z. J., “Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids – Basic Formulation,” *Journal of Computational Physics*, Vol. 178, 2002, pp. 210–251.
- ³³Wang, Z. J. and Liu, Y., “Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids – II. Extension to Two-Dimensional Scalar Equation,” *Journal of Computational Physics*, Vol. 179, 2002, pp. 665–697.
- ³⁴Wang, Z. J., Zhang, L., and Liu, Y., “High-Order Spectral Volume Method for 2D Euler Equations,” Paper 2003–3534, AIAA, June 2003.
- ³⁵Wang, Z. J. and Liu, Y., “Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids – III. One Dimensional Systems and Partition Optimization,” *Journal of Scientific Computing*, Vol. 20, No. 1, 2004, pp. 137–157.
- ³⁶Sun, Y., Wang, Z. J., and Liu, Y., “High-Order Multidomain Spectral Difference Methods for the Navier-Stokes Equations,” Paper 2006-0301, AIAA, January 2006.
- ³⁷Haselbacher, A., “A WENO Reconstruction Algorithm for Unstructured Grids Based on Explicit Stencil Reconstruction,” Paper 2005–0879, AIAA, January 2005.
- ³⁸Ivan, L. and Groth, C. P. T., “High-Order Central ENO Finite-Volume Scheme with Adaptive Mesh Refinement,” Paper 2007-4324, AIAA, June 2007.
- ³⁹Ivan, L., *Development of High-Order CENO Finite-Volume Schemes with Block-Based Adaptive Mesh Refinement*, Ph.D. thesis, University of Toronto, October 2010.
- ⁴⁰Ham, F., Lien, F.-S., and Strong, A. B., “A Cartesian Grid Method with Transient Anisotropic Adaption,” *Journal of Computational Physics*, Vol. 179, 2002, pp. 469–494.
- ⁴¹Jameson, L., “AMR vs. High Order Schemes,” *SIAM Journal on Scientific Computing*, Vol. 18, 2003, pp. 1–24.
- ⁴²Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.
- ⁴³Einfeldt, B., “On Godunov-Type Methods for Gas Dynamics,” *SIAM Journal on Numerical Analysis*, Vol. 25, 1988, pp. 294–318.
- ⁴⁴Linde, T. J., *A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere*, Ph.D. thesis, University of Michigan, May 1998.
- ⁴⁵Linde, T., “A Practical, General-Purpose, Two-State HLL Riemann Solver for Hyperbolic Conservation Laws,” *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 391–402.
- ⁴⁶Gottlieb, J. J. and Groth, C. P. T., “Assessment of Riemann Solvers for Unsteady One-Dimensional Inviscid Flows of Perfect Gases,” *Journal of Computational Physics*, Vol. 78, 1988, pp. 437–458.
- ⁴⁷Lawson, C. and Hanson, R., “Solving Least Squares Problems,” Tech. rep., Prentice-Hall, INC, 1974.
- ⁴⁸Venkatakrisnan, V., “On the Accuracy of Limiters and Convergence to Steady State Solutions,” Paper 93-0880, AIAA, January 1993.
- ⁴⁹Quirk, J. J. and Hanebutte, U. R., “A Parallel Adaptive Mesh Refinement Algorithm,” Report 93-63, ICASE, August 1993.
- ⁵⁰Quirk, J. J., *An Adaptive Grid Algorithm for Computational Shock Hydrodynamics*, Ph.D. thesis, Cranfield Institute of Technology, January 1991.
- ⁵¹Davis, R. L. and Dannenhoffer, J. F., “Decomposition and Parallelization Strategies for Adaptive Grid-Embedding Techniques,” *International Journal of Computational Fluid Dynamics*, Vol. 1, 1993, pp. 79–93.
- ⁵²Sun, M. and Takayama, K., “Conservative Smoothing on an Adaptive Quadrilateral Grid,” *Journal of Computational Physics*, Vol. 150, 1999, pp. 143–180.
- ⁵³Coirier, W. J., *An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations*, Ph.D. thesis, University of Michigan, 1994.
- ⁵⁴De Zeeuw, D. L., *A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations*, Ph.D. thesis, University of Michigan, September 1993.
- ⁵⁵Gao, X., *A Parallel Solution-Adaptive Method for Turbulent Non-Premixed Combusting Flows*, Ph.D. thesis, University of Toronto, September 2008.
- ⁵⁶Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
- ⁵⁷Knoll, D. A., McHugh, P. R., and Keyes, D. E., “Newton-Krylov Methods for Low-Mach-Number Compressible Combustion,” *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 961–967.
- ⁵⁸Cai, X.-C., Gropp, W. D., Keyes, D. E., Melvin, R. G., and Young, D. P., “Parallel Newton-Krylov-Schwarz Algorithms for the Transonic Full Potential Equations,” *SIAM Journal on Scientific Computing*, Vol. 19, No. 1, 1998, pp. 246–265.
- ⁵⁹Gropp, W. D., Kaushik, D. K., Keyes, D. E., and Smith, B. F., “High-Performance Parallel Implicit CFD,” *Parallel Computing*, Vol. 27, 2001, pp. 337–362.
- ⁶⁰Knoll, D. A. and Keyes, D. E., “Jacobian-Free Newton-Krylov Methods: A Survey of Approaches and Applications,” *Journal of Computational Physics*, Vol. 193, 2004, pp. 357–397.
- ⁶¹Groth, C. P. T. and Northrup, S. A., “Parallel Implicit Adaptive Mesh Refinement Scheme for Body-Fitted Multi-Block Mesh,” Paper 2005-5333, AIAA, June 2005.
- ⁶²Northrup, S. A. and Groth, C. P. T., “Solution of Laminar Combusting Flows Using a Parallel Implicit Adaptive Mesh Refinement Algorithm,” Fourth International Conference on Computational Fluid Dynamics, Ghent, Belgium, July 10–14, 2006, July 2006.

⁶³Northrup, S. A. and Groth, C. P. T., “Prediction of Unsteady Laminar Flames Using a Parallel Implicit Adaptive Mesh Refinement Algorithm,” *Proceedings of the U. S. National Combustion Meeting, Ann Arbor, Michigan, U.S.A., May 17–20, 2009*, 2009, p. paper 23H3.

⁶⁴Saad, Y. and Schultz, M. H., “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Equations,” *SIAM Journal for Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

⁶⁵Saad, Y., “Krylov Subspace Methods on Supercomputers,” *SIAM Journal for Scientific and Statistical Computing*, Vol. 10, No. 6, 1989, pp. 1200–1232.

⁶⁶Brown, P. N. and Saad, Y., “Hybrid Krylov Methods for Nonlinear Systems of Equations,” *SIAM Journal for Scientific and Statistical Computing*, Vol. 11, No. 3, 1990, pp. 450–481.

⁶⁷Wigton, L. B., Yu, N. J., and Young, D. P., “GMRES Acceleration of Computational Fluid Dynamics Codes,” Paper 85-1494, AIAA, July 1985.

⁶⁸Venkatakrishnan, V. and Mavriplis, D. J., “Implicit Solvers for Unstructured Meshes,” *Journal of Computational Physics*, Vol. 105, 1993, pp. 83–91.

⁶⁹Nielsen, E. J., Anderson, W. K., Walters, R. W., and Keyes, D. E., “Application of Newton-Krylov Methodology to a Three-Dimensional Unstructured Euler Code,” Paper 95-1733-CP, AIAA, June 1995.

⁷⁰Barth, T. J., “An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation,” Paper 95-0221, AIAA, January 1995.

⁷¹Pueyo, A. and Zingg, D. W., “An Efficient Newton-GMRES Solver for Aerodynamic Computations,” Paper 97-1955, AIAA, June 1997.

⁷²Gropp, W., Lusk, E., and Skjellum, A., *Using MPI*, MIT Press, Cambridge, Massachusetts, 1999.