

# Block-Based Adaptive Mesh Refinement Finite-Volume Scheme for Hybrid Multi-Block Meshes

Jason Z. X. Zheng and Clinton P. T. Groth  
Corresponding author: jzheng@utias.utoronto.ca  
University of Toronto Institute for Aerospace Studies  
4925 Dufferin Street, Toronto, Ontario, Canada

**Abstract:** A block-based adaptive mesh refinement (AMR) finite-volume scheme is proposed and developed for solution of hyperbolic conservation laws on two-dimensional hybrid multi-block meshes. A Godunov-type upwind finite-volume spatial-discretization scheme, with piecewise limited linear reconstruction and Riemann-solver based flux functions, is applied to the quadrilateral and triangular cells of the hybrid multi-block mesh and these computational cells are embedded in either body-fitted structured or general unstructured grid partitions or subdomains of the hybrid grid. A hierarchical quadtree data structure is used to allow local refinement of the individual subdomains based on heuristic physics-based refinement criteria. An efficient and scalable parallel implementation of the proposed algorithm is achieved via domain decomposition. The hybrid mesh approach readily allows for the use of body-fitted structural mesh blocks in the vicinity of bodies and solid surfaces, where the structured nature and orthogonality of the grid to the boundary can provide added accuracy and solution efficiency and the use of general unstructured partitions to fill the remaining computational domain and connecting the structured mesh. The use of unstructured grid topology to connect the body-fitted blocks near solid boundaries greatly simplifies the initial grid generation process. The performance of the proposed parallel hybrid AMR scheme is demonstrated through application to the solution of the Euler equations of compressible gas dynamics for a number of flow problems in two space dimensions. The efficiency of the AMR procedure and accuracy, robustness, and scalability of the hybrid mesh scheme are assessed.

*Keywords:* adaptive mesh refinement (AMR), hybrid mesh, domain decomposition, upwind finite-volume methods

## 1 Introduction and Motivation

Although there have been significant advances in computational fluid dynamics (CFD) in the past 20-30 year and CFD has proven to be an important enabling technology in many areas of science and engineering, a recent assessment of the needs for large-scale and high-performance scientific computing [1, 2] has identified the need for greater automation of mesh generation via adaptive mesh refinement (AMR) to reduce the time to generate high-quality meshes and for the treatment of problems having complex geometries. At the present time, a general rule of thumb is that the very least approximately 50% of the time to obtain a CFD flow solution is associated with initial mesh generation and further human intervention is required if mesh adjustment is required to improve solution quality. Computational grids that automatically adapt to the solution would therefore be extremely beneficial. Given an initially coarse mesh which can be generated in a relatively shorter period of time, an effective would then arrive at a refined high quality mesh while significantly lowering the manpower requirements and computer costs usually associated with mesh generation and the subsequent solution computation. Combining the AMR strategy with an efficient parallel solution strategy to produce a parallel AMR method that both reduces the overall problem size and the corresponding time to calculate a solution would obviously be particularly beneficial.

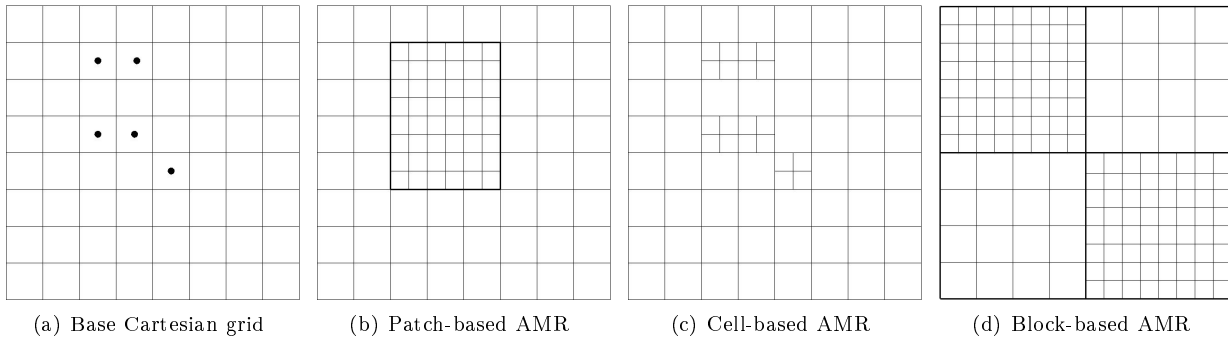


Figure 1: Illustration of (b) patch-based, (c) cell-based, and (d) block-based AMR techniques applied to a base Cartesian mesh (a) with cells flagged for refinement indicated by black dots.

## 1.1 Adaptive Mesh Refinement on Structured Mesh

Adaptive mesh refinement has proven to be very effective for treating problems with disparate length scales, providing the required spatial resolution while minimizing memory and storage requirements. When combined with Godunov-type finite-volume schemes [3], they have proved to be particularly effective for the solution of hyperbolic systems of conservation laws on structured Cartesian and body-fitted mesh and have been developed for a wide variety of engineering problems [4–33]. To date, several distinct AMR strategies have emerged which can be generally classified into four broad categories depending on the partitioning algorithm and/or data structure used to track mesh connectivity:

- i) patch-based AMR methods;
- ii) cell-based AMR methods;
- iii) block-based AMR methods;
- iv) hybrid block-based AMR techniques.

Figure 1(a) depicts a base Cartesian mesh with cells flagged for refinement. Figures 1(b)–1(d) demonstrate the subsequent refinement of this base mesh resulting from the patch-based, cell-based, and block-based AMR schemes.

Berger and Olinger, along with Colella, originally proposed a dynamic gridding technique for computing time-dependent solutions to hyperbolic partial differential equations (PDEs) in multiple space dimensions on regular Cartesian mesh [4–6]. This approach is now more generally referred to as patch-based AMR. The algorithm begins with a coarse base-level Cartesian grid and, as the calculation progresses, individual grid cells are flagged for refinement as illustrated in Figure 1(b). The patch-based AMR strategy relies on a fairly sophisticated algorithm to organize collections of individual computational cells into rectangular patches. The mesh within these newly formed patches can then be further refined, creating additional nested patches.

In cell-based AMR, as proposed and developed for example by Powell and co-workers [9–11, 16, 18, 19], Berger and Leveque [7], and Aftomis and co-workers [20, 28, 29], each cell may be refined individually as shown in Figure 1(c) and is stored using a tree data structure (quadtree in two dimensions, and octree for three dimensions). This cell-based tree structure is flexible and readily allows for the local refinement of the mesh by keeping track of the computational cell connectivity as new grid points are generated by the refinement process (4 new cells in two dimensions and 8 in three dimensions). Most cell-based approaches have been applied to Cartesian meshes and, in many cases, cut cells are used to treat complex geometry. Very efficient AMR schemes have been devised using the latter; fully three-dimensional meshes around extremely complex objects can be generated automatically and routinely in a matter of hours or less using this technique [20, 28, 29]. Nevertheless, discretization of elliptic operators on Cartesian cut cells can be challenging [16] and applications are generally restricted to hyperbolic systems.

In a block-based AMR strategy, mesh adaptation is accomplished by the dividing and coarsening of entire solution pre-defined blocks or groupings of cells. Although not required, each of the groupings or

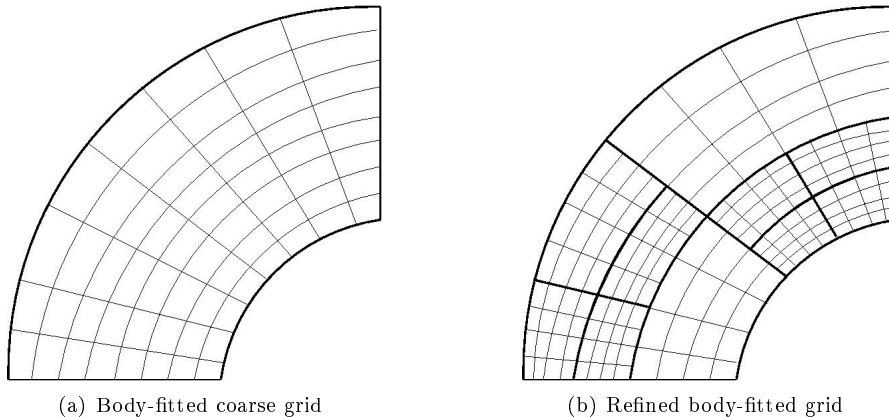


Figure 2: Illustration of block-based adaptive mesh refinement on body-fitted grid topology showing original coarse grid (a) and refined grid (b).

blocks generally has an equal number of cells as shown in Figure 1(d). Tree data structures are again used for tracking block connectivity and mesh refinement; however, the block-based AMR strategy results in a much lighter tree structure as compared to that of cell-based methods. While typically larger numbers of mesh cells are created during the refinement process (i.e., typically more than the corresponding number of cells introduced in cell-based AMR approaches), block-based methods may more readily lend themselves to efficient and scalable parallel implementations via domain decomposition [13, 26, 27, 30–33].

Applications of the efficient and scalable parallel block-based approaches on Cartesian mesh are described by Quirk [12], Berger [13], and by Groth *et al.* [26, 27, 34]. More recently, Groth and co-researchers [30–33, 35] have developed a rather flexible block-based AMR scheme allowing automatic solution-directed mesh adaptation on multi-block body-fitted (curvilinear) meshes consisting of quadrilateral (two-dimensional, 2D, case) and hexahedral computational cells (three-dimensional, 3D, case). This block-based approach has been shown to enable efficient and scalable parallel implementations for a variety of flow problems, as well as allow local refinement of body-fitted mesh with anisotropic stretching. The latter aids in the treatment of complex flow geometry and flows with thin boundary, shear, and mixing layers and/or discontinuities and shocks. Extensions of the block-based body-fitted AMR approach for embedded boundaries not aligned with the mesh [36] and with an anisotropic refinement strategy [37] are also possible and have been developed. Figure 2 illustrates the application of the block-based AMR technique to a body-fitted mesh.

Another AMR approach for treating more complex geometries with curved boundaries is based on composite overlapping or overset grids used together with AMR. In essence, a Chimera overlapping grid technique is combined with AMR and curvilinear grids that conform to the curved boundaries are used together in an overlapping fashion with one or more Cartesian grids which fill the computational domain. Figure 3 illustrates an overlapping grid consisting of two structured body-fitted grids with one annular grid and a background Cartesian grid. Boden and Toro [38] and Henshaw *et al.* [39–41] have shown that AMR on overlapping grids can provide an efficient approach for solving problems with multiple space and time scales for complex geometry. Challenges for this AMR approach are associated with determining and/or re-evaluating grid block connectivity as well as grid blocks hidden by refined grids following mesh refinement (this information must be re-computed and stored) and interpolation of solution quantities between different base grids and/or between grids with different levels to ensure that accurate results. Global conservation properties of the solution method are also difficult or impossible to enforce discretely with the overlapping grid approach.

Finally, hybrid block-based AMR approaches have also been considered. Holst and Keppens [42] applied a hybrid approach to general curvilinear coordinate systems, modifying the full tree data structure to allow for incomplete block refinement and incorporate ideas from patch-based strategies. The proposed hybrid AMR strategy requires two means to traverse the grid hierarchy, e.g., there is a doubly linked list of grid pointers per level in addition to the tree data structure. Thus, the mixed data structure further complicates the neighbour search algorithm in three-dimensions. Holst and Keppens [42] compared the three AMR

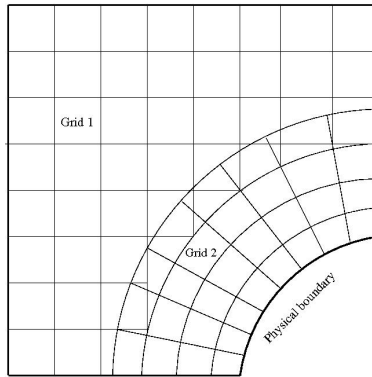


Figure 3: Illustration of overlapping grid topology consisting of two structured body-fitted component grids.

strategies, i.e., a patch-based, a tree block-based, and a hybrid block-based, for a smooth two-dimensional advection test problem on a doubly periodic domain with a second order numerical scheme, and found that the block-based AMR approach is the most efficient in terms of the execution speed for the same accuracy.

## 1.2 Adaptive Mesh Refinement on Unstructured Mesh

The use of unstructured mesh with finite-volume discretization strategies has received much attention in the past and a review of this literature is beyond the scope of the present paper. However, while AMR has been widely adapted and implemented in a variety of applications with structured grids, the majority of which are Cartesian mesh approaches, AMR is still an active area of research and under development for unstructured grids. Due to the inherent unstructured nature of such grid, current AMR approaches for unstructured meshes are mostly cell-based methods [43–45]. Refer to Figure 4 for illustration of cell-based AMR applied to a two-dimensional unstructured mesh. In this case, cell information, including neighbour connectivity, is stored in linked lists and there can be significant computational overhead due to indirect addressing of solution data. While formal block-based AMR methods are rather uncommon, parallel implementation of solution methods for unstructured grids via domain decomposition has been achieved and rather efficient dynamic mesh repartitioning algorithms have been devised based on graph partitioning and space-filling curve techniques. The multi-level graph partitioning algorithm called Metis, developed by Karypis and Kumar [46, 47], is currently used quite extensively in many CFD applications for partitioning unstructured mesh into multi-block elements containing a specified number of sub-blocks. However, recent studies (see for example, Harlacher *et al.* [48] have shown that memory requirements and message passing associated with the creation and use of all-to-all communicators can severely limit the scalability of Metis to large numbers of processors, providing additional impetus for the consideration of block-based AMR strategies for unstructured mesh. The latter would require only local re-partitioning of the mesh rather than a complete global re-partitioning of the mesh following mesh refinement.

## 1.3 Hybrid Mesh and Adaptive Mesh Refinement

Hybrid meshing techniques have received considerable interest in recent years for providing greater flexibility in meshing complex geometries in applications ranging from aerodynamic to reservoir flow simulations [49–58]. They permit the use of body-fitted structured mesh blocks in the vicinity of boundaries and solid body surfaces, where the structured nature and orthogonality of the grid to the boundary can provide added accuracy and solution efficiency. Conversely, general unstructured grid topology can be used to fill the remaining computational domain and connecting the structured regions of the mesh, thereby reducing the level of human intervention required to generate the mesh, allowing for greater automation of the mesh generation process, and potentially reducing the overall time for mesh generation. As with unstructured meshes, AMR strategies for hybrid mesh approaches have for the most part been limited to cell-based refinement techniques [49–51] although, as noted above for structured mesh, block-based approaches may

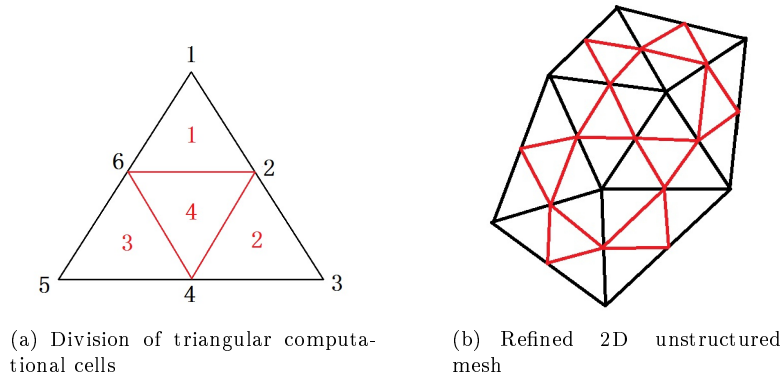


Figure 4: Illustration of cell-based AMR applied to a two-dimensional unstructured mesh showing (a) division of triangular computational cell into four sub-elements and (b) a grouping of triangular cells following uniform refinement of the triangular elements showing newly added elements in red.

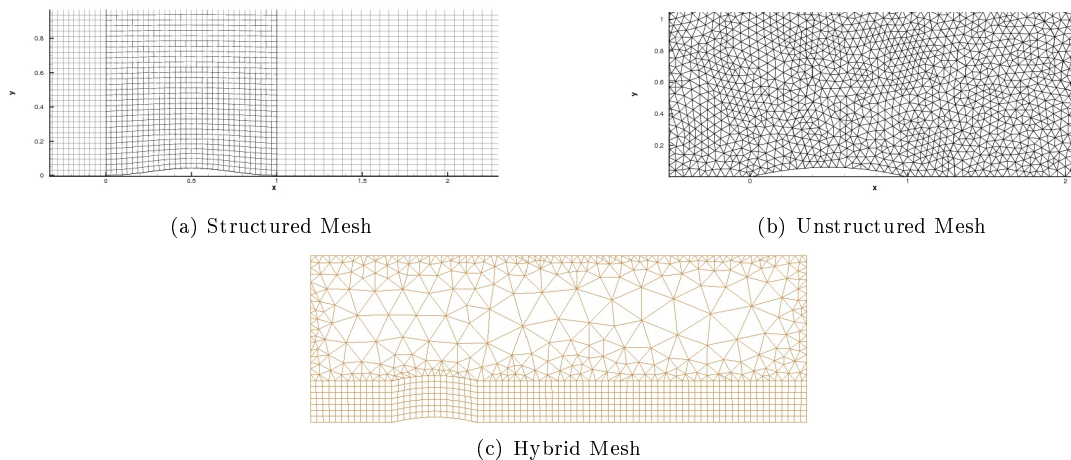


Figure 5: Comparison of the different mesh topologies of (a) structured, (b) unstructured, and (c) hybrid meshing techniques applied to two-dimensional channel flow with a bump.

offer advantages in the context of parallel implementation of solution methods on massively parallel computer architectures. As a comparison, examples of structured, unstructured, and hybrid meshes generated for the same two-dimensional channel flow geometry with a bump are depicted in Figure 5.

## 1.4 Scope of Present Study

The development of an AMR strategy for hybrid meshes is considered in the present work. A block-based AMR finite-volume scheme is proposed for the solution of hyperbolic conservation laws on two-dimensional hybrid multi-block meshes. A Godunov-type upwind finite-volume spatial-discretization scheme, with piecewise limited linear reconstruction and Riemann-solver based flux functions, is applied to the quadrilateral and triangular cells of the hybrid multi-block mesh and these computational cells are embedded in either body-fitted structured or general unstructured grid partitions or subdomains of the hybrid grid. A hierarchical quadtree data structure is used to allow local refinement of the individual subdomains based on heuristic physics-based refinement criteria. The data structure permits an efficient and scalable parallel implementation of the proposed algorithm via domain decomposition. In addition, the nature of the structured blocks is exploited to reduce computational overhead and storage. The performance of the proposed parallel hybrid

AMR scheme is demonstrated through application to the solution of the Euler equations of compressible gas dynamics for a number of flow configurations and regimes in two space dimensions. The efficiency of the AMR procedure and accuracy, robustness, and scalability of the hybrid mesh scheme are assessed.

## 2 Equations of Compressible Gas Dynamics

For the development of the proposed AMR algorithm for hybrid mesh, solutions of the Euler equations governing compressible inviscid flows of polytropic gases in two space dimensions are considered. For 2D planar flows, the conservative form of the Euler equations reflecting the conservation of mass, momentum, and energy can be summarized as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \quad (1)$$

where  $\mathbf{U}$  is the conserved variable solution vector given by

$$\mathbf{U} = [ \rho, \rho u, \rho v, \rho e ]^T, \quad (2)$$

$x$  and  $y$  are the spatial coordinates,  $t$  is time,  $\rho$  is the gas density,  $u$  and  $v$  are the velocity components in the  $x$ - and  $y$ -coordinate directions,  $e = p/(\rho(\gamma - 1)) + (u^2 + v^2)/2$  is the specific total energy,  $p = \rho RT$  is the pressure,  $T$  is the gas temperature,  $R$  is the gas constant,  $\gamma$  is the specific heat ratio, and  $\mathbf{F}$  and  $\mathbf{G}$  are  $x$ - and  $y$ -direction solution flux vectors given by

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho e + p) \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho e + p) \end{bmatrix}. \quad (3)$$

For a polytropic gas (thermally and calorically perfect gas), the ratio of specific heats,  $\gamma$ , is a constant and the specific heats are given by  $C_v = R/(\gamma - 1)$  and  $C_p = \gamma R/(\gamma - 1)$ .

## 3 Godunov-Type Finite-Volume Scheme

The preceding Euler equations have a hyperbolic nature. The proposed AMR algorithm therefore makes use of a cell-centred Godunov-type upwind finite-volume spatial discretization procedure [3] in conjunction with limited linear solution reconstruction and Riemann-solver based flux functions to solve the conservation form of these PDEs on multi-block mesh composed of either quadrilateral or triangular computational cells. The semi-discrete form of this finite-volume formulation applied to any cell,  $i$ , is given by

$$\frac{d\mathbf{U}_i}{dt} = -\frac{1}{A_i} \sum_k \left( \vec{\mathbf{F}} \cdot \vec{\mathbf{n}} \Delta \ell \right)_{i,k} = \mathbf{R}_i(\mathbf{U}), \quad (4)$$

where  $\mathbf{U}_i$  is the area-averaged conserved solution for cell  $i$ ,  $\vec{\mathbf{F}} = (\mathbf{F}, \mathbf{G})$  is the flux dyad,  $A_i$  is the area of the cell, and  $\Delta \ell$  and  $\vec{\mathbf{n}}$  are the length of the cell face and unit vector normal to the cell face or edge, respectively. Refer to Figure 6 for illustrations of quadrilateral and triangular computational cell configurations, cell faces, normals. The vector,  $\mathbf{R}$ , is referred to as the residual vector. The numerical fluxes at the faces,  $k$ , of each cell,  $\vec{\mathbf{F}} \cdot \vec{\mathbf{n}}$ , are determined from the solution of a Riemann problem. Given the left and right solution states,  $\mathbf{U}_l$  and  $\mathbf{U}_r$ , at the cell interfaces, the numerical flux is given by

$$\vec{\mathbf{F}} \cdot \vec{\mathbf{n}} = \mathcal{F}(\mathbf{U}_l, \mathbf{U}_r, \vec{\mathbf{n}}), \quad (5)$$

where the numerical flux  $\mathcal{F}$  is evaluated by solving a Riemann problem in a direction defined by the normal to the face with initial data  $\mathbf{U}_l$  and  $\mathbf{U}_r$ . The left and right solution states are determined via a least-squares piece-wise limited linear solution reconstruction procedure in conjunction with either the Barth-Jespersen

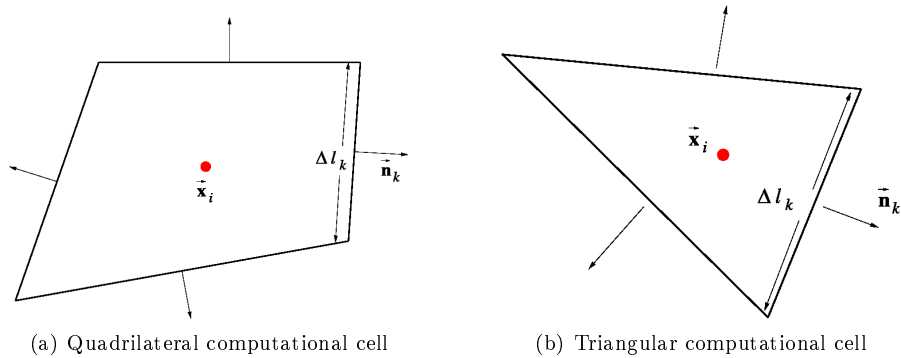


Figure 6: Schematic diagram of (a) quadrilateral and (b) triangular computational cells making up computational domain of 2D multi-block computational grid.

or Venkatakrishnan limiters [59,60]. In the present algorithm, both exact and approximate Riemann solvers can be used to solve the Riemann problem and evaluate the numerical flux. The Roe linearized Riemann solver [61], HLLE and modified HLLE flux function due to Linde [62–64], the HLLC flux function [65], and the exact Riemann solver of Gottlieb and Groth [66] have all been implemented and may be used.

Solutions of the semi-discrete form of the governing equations given in Eq. (4), represented by the area-averaged solution quantities within each computational cell,  $\mathbf{U}_i$ , is obtained herein by applying a standard second-order accurate, Runge-Kutta, explicit time-marching scheme to the resulting coupled non-linear ordinary differential equations (ODEs). Steady-state solutions are obtained by advancing the solution in time until a converged time-invariant solution is achieved. While the latter is certainly non-optimal, it is sufficient for the purposes of the present work.

## 4 Multi-Block Hybrid Mesh

The computational triangular and quadrilateral computational cells described above are embedded in either fully structured or fully unstructured grid blocks, respectively. The structured body-fitted grid blocks are taken to consist of  $N_{\text{cells}} = N_i \times N_j$  cells, where  $N_i$  and  $N_j$  are even, but not necessarily equal integers, representing the number of cells in each logical coordinate direction of the body-fitted mesh block. Refer to Figure 7(a). The unstructured grid blocks are also each taken to consist of  $N_{\text{cells}}$  computational cells, but in this case of triangular topology. An example of a multi-block unstructured mesh is depicted in Figure 8(a).

Solution data associated with each structured grid block are stored in indexed two-dimensional array data structures and it is therefore straightforward to obtain solution information from neighbouring cells within the blocks. Solution data within each of the corresponding unstructured grid blocks are stored in an edge-based, link-list, data structure [59]. The data structure contains both cell-centred solution data and the cell vertices. The connectivity of the cell faces and vertices is stored in a quadruple for each edge that consists of pointers to the two vertices defining the interface as well as pointers to two cells that share that the edge. One obvious advantage of this type of data structure is that it allows for straightforward traversal of each cell face during integration of the solution fluxes. It also affords a means for retrieval of cell connectivity.

Various techniques are used to generate the initial (unrefined) body-fitted structured mesh blocks for the hybrid mesh considered in this study. The initial unstructured mesh is generated here using the Gmsh software developed by Geuzaine and Remacle [67]. Domain decomposition or partitioning of the initial unstructured grid into multiple grid blocks is obtained using the Metis mesh partitioning software [46, 47]. Metis creates mesh partitions that are approximately of the same size (i.e., number of cells) while minimizing the number of faces residing on boundaries of the partitions, two desirable features when parallel implementation of the solution method is considered. The connectivity of the resulting hybrid grid blocks is stored directly in a hierarchical quadtree data structure that is described below. As an example, the block topology for a two-dimensional multi-block hybrid mesh consisting of both structured and unstructured grid

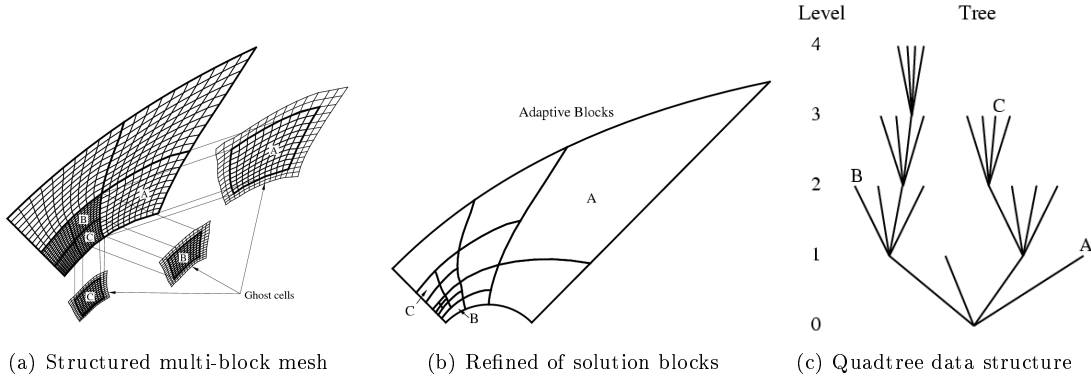


Figure 7: Multi-block body-fitted structured quadrilateral mesh of block-based hybrid AMR algorithm illustrating (a) the structured mesh blocks and layers of overlapping ghost cells, (b) refined solution blocks arising from four levels of refinement applied to a single initial block, and (c) the associated hierarchical quadtree data structure.

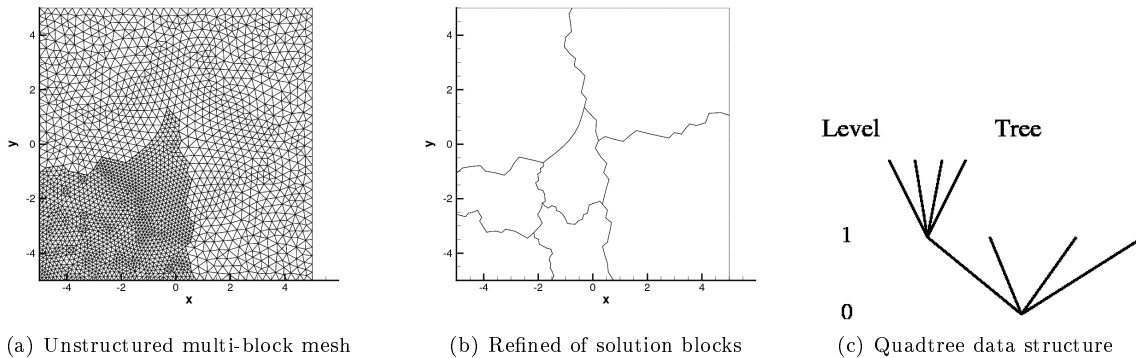


Figure 8: Multi-block unstructured triangular mesh of block-based hybrid AMR algorithm illustrating (a) the structured mesh blocks, (b) refined solution blocks arising from two levels of refinement applied to a single initial block, and (c) the associated hierarchical quadtree data structure.

blocks for flow past two circular cylinders in a channel is given in Figure 9.

## 5 Block-Based Adaptive Mesh Refinement for Hybrid Mesh

As noted above, rather flexible block-based AMR schemes allowing automatic solution-directed mesh adaptation on multi-block body-fitted meshes consisting of quadrilateral and hexahedral computational cells have been developed Groth and co-researchers [30–33, 35]. These block-based approaches have been shown to enable efficient and scalable parallel implementations for a variety of flow problems, as well as allow local refinement of body-fitted mesh with anisotropic stretching according to physics-based refinement criteria. The anisotropic stretching permits the use of anisotropic mesh for resolving thin solution layers, such as boundary and free shear layers. Note that although the proposed block-based AMR approach is somewhat less flexible and incurs some inefficiencies in solution resolution as compared to a cell-based approaches (i.e., for the same solution accuracy, generally more computational cells are introduced in the adapted grid), the block-based method can offer many advantages over cell-based techniques when computational performance and parallel implementation of the solution algorithm is considered.

In the proposed AMR algorithm for multi-block hybrid mesh, the block-based AMR algorithm of Groth



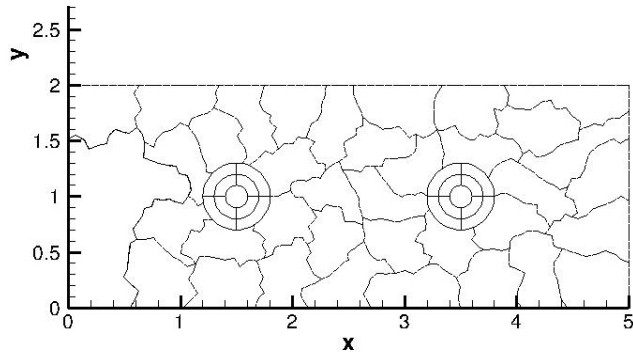


Figure 9: Illustration of block topology for a two-dimensional multi-block hybrid mesh consisting of both structured and unstructured grid blocks for flow past two circular cylinders in a channel.

and co-researchers [30–33, 35] is applied directly to the structured grid blocks and further extended to allow local refinement of the unstructured grid blocks. In what follows, the proposed AMR scheme for hybrid mesh is described, with particular emphasis on the required extensions for the treatment of the unstructured mesh blocks.

## 5.1 Refinement and Coarsening of Multi-Block Mesh

Mesh adaptation of the multi-block hybrid mesh is accomplished by the dividing and coarsening of appropriate solution blocks. In regions requiring increased cell resolution, a “parent” block is refined by dividing itself into four “children” or “offspring”. Each of the four quadrants or sectors of a parent block becomes a new block having the same number of cells as the parent and thereby doubling the cell resolution in the region of interest. This process can be reversed in regions that are deemed over-resolved and four children are coarsened into a single parent block. Note however that no regions of the mesh can be made coarser than it was originally and the mesh refinement is constrained such that resolution changes of only a factor of two is permitted between adjacent or neighbouring blocks.

For the structured body-fitted solution blocks, the generation of the mesh points in the refined grid blocks is obtained from the initially coarser mesh block by making use of the grid metrics of the body-fitted (curvilinear) parent mesh [32]. Use of the grid metrics in determining the grid points of the refined grid blocks is very effective in preserving the original mesh point clustering in the body-fitted mesh and maintaining the smoothness and locations of the grid lines in the mesh. Standard restriction and prolongation operators are used to evaluate the solution on all blocks created by the coarsening and division processes, respectively. Figure 7(b) illustrates the refinement and coarsening of the structured blocks of the hybrid mesh.

For the unstructured grid blocks, an isotropic refinement procedure is applied to each triangular cell of the parent block in which the new nodes or vertices of the refined blocks are defined simply by using the original vertices and mid-points of the faces of the parent-block triangular cells connecting the adjacent newly defined nodes as shown in Figure 4(b). Partitioning of the refined grid into the four children blocks is then accomplished by using Metis applied now only to the refined cells of the original parent block. In this way local refinement procedure for grid blocks is straightforwardly extended to the unstructured blocks. Note also that the parallel scalability limitations of Metis [48] for large numbers of processors and hence partitions is avoided in this block-based approach. Metis is only used to partition each refined parent block individually into a small number of partitions (four).

## 5.2 Block Connectivity and Quadtree Data Structure

A hierarchical quadtree data structure with multiple “roots”, multiple “trees”, and additional interconnects between the “leaves” of the trees is used to keep track of mesh refinement and the connectivity between grid blocks in the hybrid mesh. For the structured grid blocks, this quadtree data structure is depicted in Figure 7(c). Each grid block corresponds to a node of the quadtree structure. The blocks of the initial mesh are the

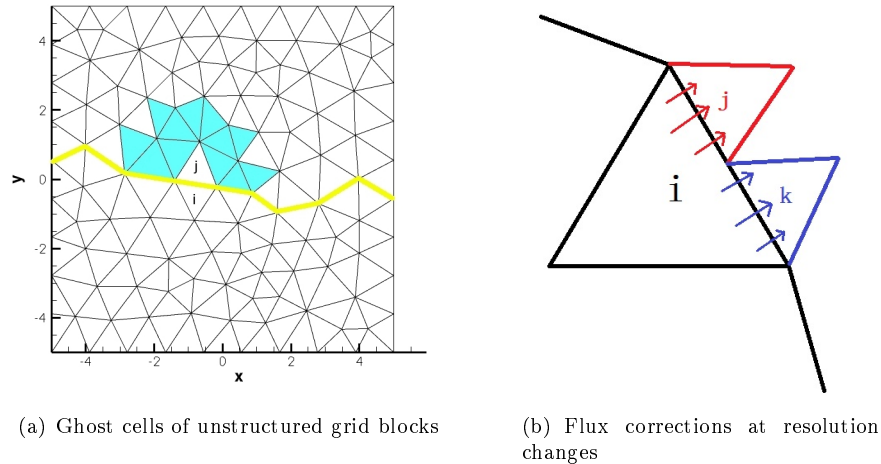


Figure 10: (a) Ghost cells of multi-block unstructured triangular mesh used to facilitate the exchange of solution information between blocks and (b) schematic illustrating the application of conservative flux corrections.

roots of the tree structure. Associated with each root is a separate quadtree data structure that contains all of the blocks making up the leaves of the tree created during mesh refinement. The proposed data structure allows for fully unstructured connectivity of the root blocks of the multi-block mesh. For the structured blocks, the connectivity and orientation of the root blocks are computed directly from the grid geometry information [32].

The proposed quadtree data structure is sufficiently general to handle mesh refinement and coarsening of the unstructured grid blocks and store information associated with grid block connectivity as shown in Figure 8(c). Of particular importance is the modification of the unstructured root block connectivity to handle the unstructured grid blocks.

Traversal of the quadtree structure by recursively visiting the parents and children of solution blocks can be used to determine block connectivity. However, in order to reduce overhead associated with accessing solution information from adjacent blocks, the neighbours of each block are computed and stored, providing direct interconnects between blocks of the hierarchical data structure that are neighbours in physical space. One of the advantages of the hierarchical quadtree data structure is that it readily permits local mesh refinement. Local modifications to the multi-block mesh can be performed without re-gridding the entire mesh and re-calculating all grid block connectivities.

### 5.3 Exchange of Solution Data and Ghostcells

In order that the finite-volume scheme may be applied to all blocks in a more independent manner, some solution information is shared between adjacent blocks having common interfaces. For the structured grid blocks, this information is stored in an additional layers of overlapping “ghost” cells associated with each block as shown in Figure 7(a). Transformation matrices and offsets are used to facilitate the communication between neighboring blocks and deal with the general unstructured block connectivity introduced at the roots [32]. At interfaces between blocks of equal resolution, these ghost cells are simply assigned the solution values associated with the appropriate interior cells of the adjacent blocks. At resolution changes, restriction and prolongation operators, similar to those used in block coarsening and division, are employed to evaluate ghost cell solution values. Additional inter-block communication is also required at interfaces with resolution changes to strictly enforce the flux conservation properties of the finite-volume scheme [4, 6]. In particular, the interface fluxes computed on more refined blocks are used to correct the interface fluxes computed on coarser neighbouring blocks and ensure the solution fluxes are conserved across block interfaces.

For the unstructured grid blocks, ghost cells also introduced at unstructured block boundaries to facilitate

exchange of inter-block solution information as depicted in Figure 10(a) and interface flux corrections applied as in structured block case. See Figure 10(b).

## 5.4 Mesh Refinement Criteria

Although several approaches are possible, for this study, the coarsening and division of blocks are directed using multiple physics-based refinement criteria [9]. Refinement criteria,  $\epsilon$ , based on a combination of the density gradient, and divergence and curl of the velocity provide reliable detection of flow features such as shocks, contact surfaces, stagnation points, and shear layers. Mathematically, these criteria can be written as

$$\epsilon_1 \propto \vec{\nabla} \rho, \quad \epsilon_2 \propto \vec{\nabla} \cdot \vec{u}, \quad \epsilon_3 \propto \vec{\nabla} \times \vec{u}. \quad (6)$$

These refinement criteria are used for both the structured and unstructured grid blocks of the hybrid mesh.

## 5.5 Parallel Implementation

By design, the proposed multi-block AMR scheme for hybrid mesh is well suited to parallel implementation on distributed-memory multi-processor architectures. A parallel implementation of the block-based AMR scheme has been developed using the C++ programming language and the MPI (message passing interface) library. For homogeneous architectures with multiple processors all of equal speed, the self-similar nature of the solution blocks is exploited and parallel implementation is carried out via domain decomposition where the solution blocks are simply distributed equally among the available processors, with more than one block permitted on each processor. A simple stack is used to keep track of available (open or unused) processors. For heterogeneous machines, such as a 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.

In order to carry out mesh refinement and inter-block communication, a complete copy of the hierarchical quadtree data structure is stored on each processor. This is possible because, unlike cell-based meshing techniques, the block-based tree data structure is not overly large. The structure need only retain the connectivity between the solution blocks as opposed to a complete map of the cell connectivity, as required by general unstructured mesh solution methods. Inter-processor communication is mainly associated with block interfaces and involves the exchange of ghost-cell solution values and conservative flux corrections. Message passing of the this information is performed in an asynchronous fashion with gathered wait states and message consolidation.

## 5.6 AMR for Multi-Block Unstructured Mesh

To illustrate the proposed AMR scheme applied to a general unstructured multi-block mesh, consider the results of Figure 11. The figure shows both the grid blocks and triangular computational cells of the initial and final refined mesh after application of three levels of refinement. Note the increased resolution afforded by the local refinement of the unstructured blocks.

# 6 Numerical Results

The numerical results are now described for several flow problems to illustrate the capabilities of the proposed solution method. Supersonic flow past a circular cylinder and supersonic flow past two circular cylinders in a channel are considered. All computations were performed on a high performance parallel cluster consisting of 3,780 Intel Xeon E5540 (2.53GHz) nodes with 16GB RAM per node. The cluster is connected with a high speed InfiniBand switched fabric communications link.

## 6.1 Supersonic Flow Over a Circular Cylinder, $M_\infty = 2$

The first test case considered is that of steady state (time-invariant) supersonic flow past a two-dimensional circular cylinder with its axis of symmetry perpendicular to the free-stream flow direction such that a stationary bow shock forms about the body. The free-stream Mach number is  $M_\infty = 2$  for the case of

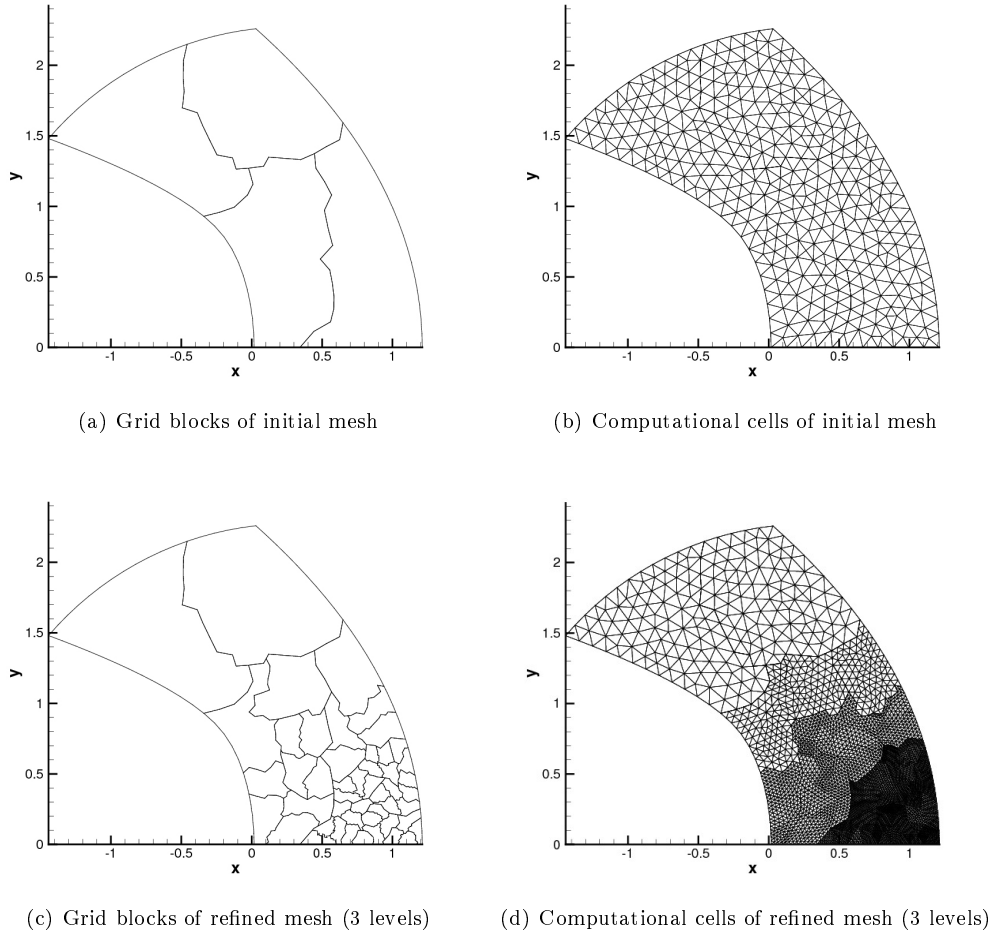


Figure 11: Illustration of AMR applied to multi-block unstructured mesh with three level of refinement showing both the grid blocks and triangular computational cells of the initial and final refined mesh.

interest. The cylinder flow problem with solved using 128 grid blocks without applying the AMR procedure. However, the relative number of structured and unstructured blocks in the hybrid mesh was varied in order to investigate the effects of solution accuracy and parallel performance of the proposed solution algorithm. In each case, the total number of computational cells in the mesh was maintained at at about 32,500. Table 1 shows the numbers of structured and unstructured blocks used in each of the simulations.

Predicted solutions for the five different outlined in Table 1 are shown in Figure 12. It is noted that the predicted density distributions for each of the hybrid grids have the general features: a strong bow shock is present ahead of the cylinder with a region of high density flow following directly behind the shock. Expansion waves result in the rear of the cylinder and lead to low density regions downstream of the cylindrical body. Moreover, the flow past the cylinder is symmetrical with respect to the horizontal axis of the cylinder.

Although Figure 12 indicates that the predicted solutions on each of the hybrid mesh have many similarities, there are some subtle differences in the predicted solutions. In particular, it is observed that the structured mesh blocks tend to exhibit greater solution quality due to the orthogonality nature of the mesh cells.

The parallel performance of the proposed AMR scheme for hybrid mesh has also been assessed for this steady-state supersonic flow problem. Strong scaling of the algorithm for this fixed size problem with 128 partitions was examined. The parallel performance of the proposed solution method is shown in Figure 13 for this case. The figure shows the parallel run time associated with the computation of the supersonic circular

Table 1: Combinations of structured and unstructured grid blocks used in hybrid mesh for supersonic flow over a circular cylinder,  $M_\infty = 2$ .

	Number of Structured Blocks	Number of Unstructured Blocks
<b>Test Case 1 (TC1)</b>	0	128
<b>Test Case 2 (TC2)</b>	32	96
<b>Test Case 3 (TC3)</b>	64	64
<b>Test Case 4 (TC4)</b>	96	32
<b>Test Case 5 (TC5)</b>	128	0

cylinder flow problem for the hybrid computational mesh outlined in Table 1 as a function of the number of processor cores, from 8 up to 128 cores. It can be seen that, for all cases, as the number of processors used to compute the problem increases the run time is correspondingly reduced with a parallel efficiency that is very close to unity, indicating that there is minimum parasitic performance loss due to communication between the processors. Furthermore, the parallel is virtually unaffected by the topology of the hybrid mesh, indicating that any additional overhead associated with the hybrid mesh capabilities is small.

## 6.2 Supersonic Flow Past Two Circular Cylinders in a Channel, $M_\infty = 2$

Further validation of the proposed parallel AMR finite-volume scheme for hybrid mesh is sought by considering a second test cases involving steady-state supersonic flow past two circular cylinders located within a channel. The upstream Mach number in this case is again  $M_\infty = 2$  and the two cylinder are of different sizes (i.e., have different radii) and are placed asymmetrically with respect to the symmetry axis of the channel.

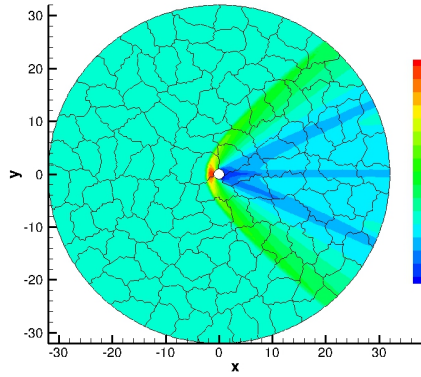
Figure 14 depicts the predicted distributions of the flow density for the supersonic flow past the two circular cylinders asymmetrically located within a channel with  $M_\infty = 2$ . Results are shown for five successively refined multi-block hybrid mesh. The original/initial unrefined hybrid mesh is not shown but contained 14 solution blocks and 2988 cells. The finest mesh after five levels of refinement has 1460 blocks and 311,724 cells. It is evident the the proposed AMR scheme for hybrid mesh is able to resolve the bow shocks associated with each cylinder, the reflected shocks off the channel walls, and the complex non-linear wave and shock interactions that occur in the flow as the mesh is refined. A measure of the efficiency of the block-based AMR scheme for this problem can be defined by a refinement efficiency parameter,  $\alpha$ , given by

$$\alpha = 1 - N_{\text{cells}}/N_{\text{uniform}} \quad (7)$$

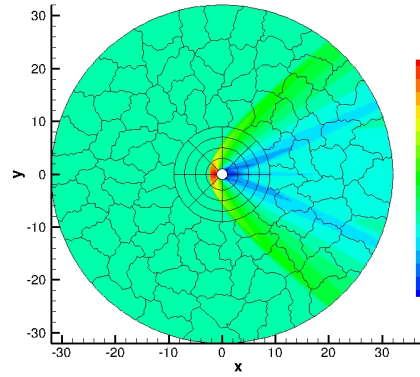
where  $N_{\text{cells}}$  is the actual number of cells in the mesh and  $N_{\text{uniform}}$  is the total number of cells that would have been created on a uniformly refined mesh composed of solution blocks all at the finest level. The efficiency of the AMR scheme is  $\alpha = 0$  for the initial mesh, where all solution blocks at the same level of refinement, but rapidly improves as the number of refinement levels increases. A refinement efficiency of  $\alpha = 0.89812$  is achieved on the finest mesh, indicating the ability of the block-based AMR approach to deal with flows having disparate spatial scales by reducing the number of computational cells required to solve a problem while maintaining solution resolution in areas of interest.

## 7 Concluding Remarks and Future Research

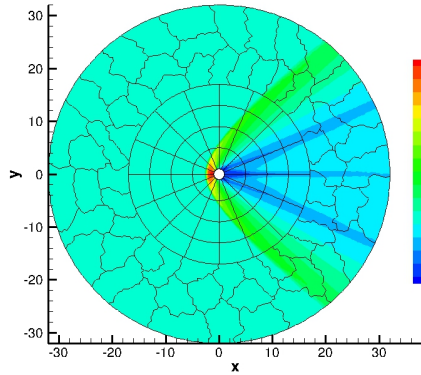
A block-based adaptive mesh refinement finite-volume scheme has been described for the solution of hyperbolic conservation laws on two-dimensional multi-block hybrid meshes. A Godunov-type upwind finite-volume spatial-discretization scheme, with piecewise limited linear reconstruction and Riemann-solver based flux functions, is applied to the solution of the hyperbolic PDEs on quadrilateral and triangular cells of the hybrid multi-block mesh and these computational cells are embedded in either body-fitted structured or general unstructured grid partitions or subdomains of the hybrid grid. A hierarchical quadtree data structure



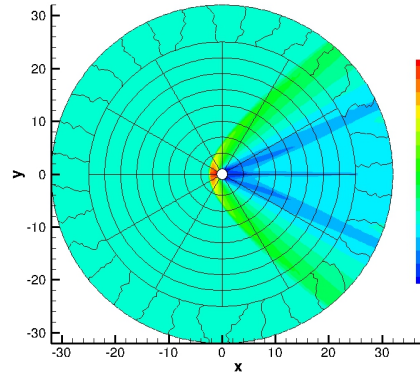
(a) TC1: 128 unstructured blocks



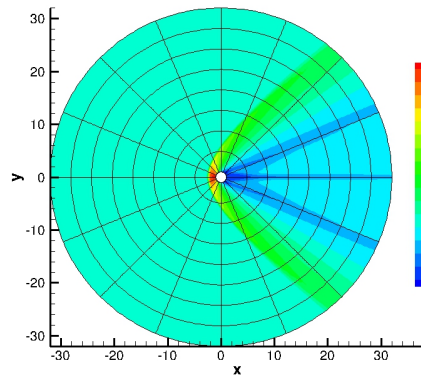
(b) TC2: 96 unstructured blocks and 32 structured blocks



(c) TC3: 64 unstructured blocks and 64 structured blocks



(d) TC4: 32 unstructured blocks and 96 structured blocks



(e) TC5: 128 structured blocks

Figure 12: Predicted distributions of the flow density for supersonic flow over a circular cylinder with  $M_\infty = 2$  for the five different hybrid mesh described in Table 1

has been developed to allow local refinement of the individual subdomains based on heuristic physics-based

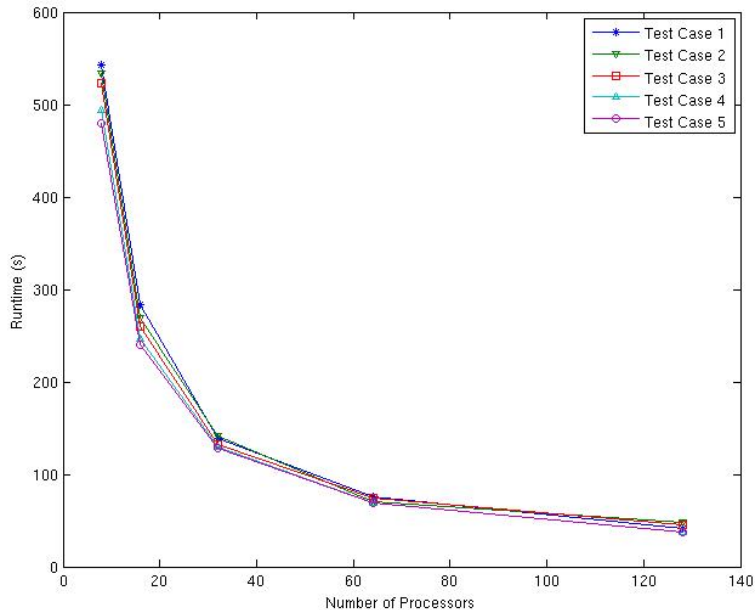


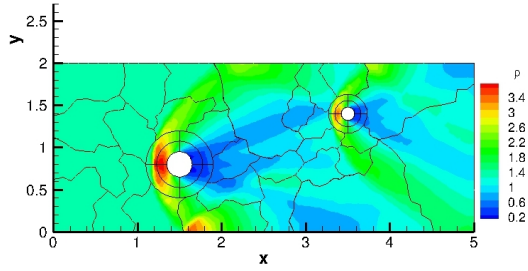
Figure 13: Parallel run time associated with the computation of supersonic flow over a circular cylinder with  $M_\infty = 2$  for the hybrid computational mesh outlined in Table 1 as a function of the number of processor cores.

refinement criteria. An efficient and scalable parallel implementation of the proposed algorithm is achieved via domain decomposition. The hybrid mesh approach readily allows for the use of body-fitted structural mesh blocks in the vicinity of bodies and solid surfaces, where the structured nature and orthogonality of the grid to the boundary can provide added accuracy and solution efficiency and the use of general unstructured partitions to fill the remaining computational domain and connecting the structured mesh. The performance of the proposed parallel hybrid AMR scheme has been demonstrated through application of the proposed algorithm to the solution of the Euler equations. Several steady-state supersonic flow problems in two space dimensions with strong non-linear wave interactions were investigated. The AMR procedure for hybrid mesh was shown to be effective in accurately capturing solution discontinuities while reducing the overall size of the computational mesh. The parallel scalability of the proposed hybrid-mesh scheme was also investigated and high parallel performance was observed for up to 128 processor cores.

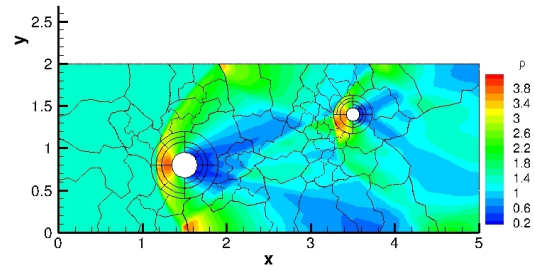
The present study represents the first steps towards an effective AMR strategy for hybrid mesh. Future follow-on research will involve further evaluation of proposed scheme, the application of error-based refinement criteria for directing the mesh adaptation, application to the hybrid mesh algorithm to the solution of viscous flows, and possible extensions of the method to three-dimensional flows.

## Acknowledgments

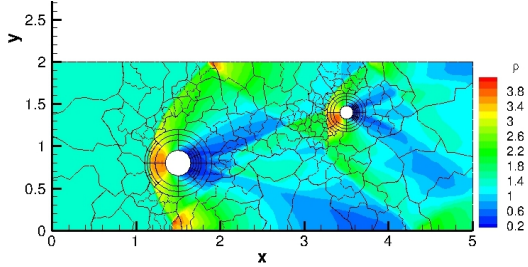
Financial support for the research described herein was provided by MITACS (Mathematics of Information Technology and Complex Systems) Network, part of the Networks of Centres of Excellence (NCE) program funded by the Canadian government. This funding is gratefully acknowledged with many thanks. Computational resources for performing all of 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.



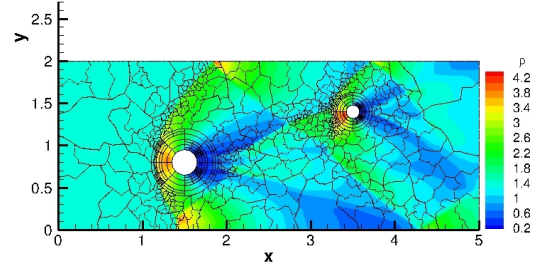
(a) Refinement Level 1: 44 blocks, 8826 cells,  $\alpha = 0.261546$



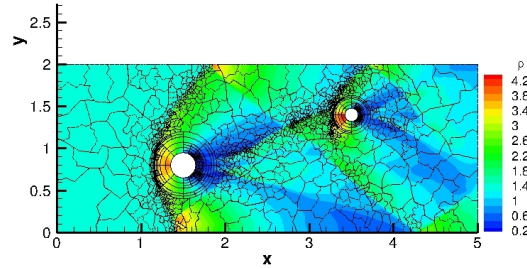
(b) Refinement Level 2: 110 blocks, 22512 cells,  $\alpha = 0.529116$



(c) Refinement Level 3: 272 blocks, 57087 cells,  $\alpha = 0.701478$



(d) Refinement Level 4: 656 blocks, 138513 cells,  $\alpha = 0.81892$



(e) Refinement Level 5: 1460 blocks, 311724 cells,  $\alpha = 0.89812$

Figure 14: Predicted distributions of the flow density for supersonic flow past two circular cylinder located within a channel with  $M_\infty = 2$  five successively refined multi-block hybrid mesh (original/initial unrefined hybrid mesh is not shown).

## References

- [1] D. E. Keyes. A science-based case for large-scale simulation volume 1. Report prepared for the Office of Science, U. S. Department of Energy, July 2003.
- [2] D. E. Keyes. A science-based case for large-scale simulation volume 2. Report prepared for the Office of Science, U. S. Department of Energy, September 2004.
- [3] S. K. Godunov. Finite-difference method for numerical computations of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271–306, 1959.
- [4] M. J. Berger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.
- [5] M. J. Berger. Data structures for adaptive grid generation. *SIAM J. Sci. Stat. Comput.*, 7(3):904–916, 1986.
- [6] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:67–84, 1989.



- [7] M. J. Berger and R. J. LeVeque. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. Paper 89-1930, AIAA, June 1989.
- [8] J. J. Quirk. *An Adaptive Grid Algorithm for Computational Shock Hydrodynamics*. PhD thesis, Cranfield Institute of Technology, January 1991.
- [9] K. G. Powell, P. L. Roe, and J. Quirk. Adaptive-mesh algorithms for computational fluid dynamics. In M. Y. Hussaini, A. Kumar, and M. D. Salas, editors, *Algorithmic Trends in Computational Fluid Dynamics*, pages 303–337. Springer-Verlag, New York, 1993.
- [10] D. De Zeeuw and K. G. Powell. An adaptively refined Cartesian mesh solver for the Euler equations. *J. Comput. Phys.*, 104:56–68, 1993.
- [11] D. L. De Zeeuw. *A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations*. PhD thesis, University of Michigan, September 1993.
- [12] J. J. Quirk and U. R. Hanebutte. A parallel adaptive mesh refinement algorithm. Report 93-63, ICASE, August 1993.
- [13] M. J. Berger and J. S. Saltzman. AMR on the CM-2. *Appl. Numer. Math.*, 14:239–253, 1994.
- [14] A. S. Almgren, T. Buttke, and P. Colella. A fast adaptive vortex method in three dimensions. *J. Comput. Phys.*, 113:177–200, 1994.
- [15] J.B. Bell, M.J. Berger, J.S. Saltzman, and M. Welcome. A three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 15:127–138, 1994.
- [16] W. J. Coirier. *An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations*. PhD thesis, University of Michigan, 1994.
- [17] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome. An adaptive cartesian grid method for unsteady compressible flow in irregular regions. *J. Comput. Phys.*, 120:278–304, 1995.
- [18] W. J. Coirier and K. G. Powell. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. *J. Comput. Phys.*, 117:121–131, 1995.
- [19] W. J. Coirier and K. G. Powell. Solution-adaptive Cartesian cell approach for viscous and inviscid flows. *AIAA J.*, 34(5):938–945, May 1996.
- [20] M. J. Aftosmis, M. J. Berger, and J. E. Melton. Robust and efficient Cartesian mesh generation for component-based geometry. *AIAA J.*, 36(6):952–960, 1998.
- [21] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. J. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, 142:1–46, 1998.
- [22] R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, M. A. Fiveland, and J. P. Jessee. An adaptive projection method for unsteady, low mach number combustion. *Combust. Sci. Tech.*, 140:123–168, 1998.
- [23] J. P. Jessee, W. A. Fiveland, L. H. Howell, P. Colella, and R. B. Pember. An adaptive mesh refinement algorithm for the radiative transport equation. *J. Comput. Phys.*, 139(2):380–398, 1998.
- [24] P. Colella, M. Dorr, and D. Wake. Numerical solution of plasma-fluid equations using locally refined grids. *J. Comput. Phys.*, 152:550–583, 1999.
- [25] L.H. Howell, R.B. Pember, P. Colella, J.P. Jessee, and W. A. Fiveland. A conservative adaptive-mesh algorithm for unsteady, combined-mode heat transfer using the discrete ordinates method. *Numer. Heat Trans.*, 35:407–430, 1999.
- [26] C. P. T. Groth, D. L. De Zeeuw, K. G. Powell, T. I. Gombosi, and Q. F. Stout. A parallel solution-adaptive scheme for ideal magnetohydrodynamics. Paper 99-3273, AIAA, June 1999.
- [27] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, and K. G. Powell. Global three-dimensional MHD simulation of a space weather event: CME formation, interplanetary propagation, and interaction with the magnetosphere. *J. Geophys. Res.*, 105(A11):25,053–25,078, 2000.
- [28] M. J. Aftosmis, M. J. Berger, and G. Adomavicius. A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries. Paper 2000-0808, AIAA, January 2000.
- [29] M. J. Aftosmis, M. J. Berger, and S. M. Murman. Applications of space-filling curves to cartesian methods for cfd. Paper 2004-1232, AIAA, January 2004.
- [30] J. S. Sachdev, C. P. T. Groth, and J. J. Gottlieb. A parallel solution-adaptive scheme for predicting multi-phase core flows in solid propellant rocket motors. *Int. J. Comput. Fluid Dyn.*, 19(2):157–175, 2005.
- [31] X. Gao and C. P. T. Groth. A parallel adaptive mesh refinement algorithm for predicting turbulent

- non-premixed combustng flows. *Int. J. Comput. Fluid Dyn.*, 20(5):349–357, 2006.
- [32] X. Gao and C. P. T. Groth. A parallel solution-adaptive method for three-dimensional turbulent non-premixed combustng flows. *J. Comput. Phys.*, 229(5):3250–3275, 2010.
- [33] X. Gao, S. A. Northrup, and C. P. T. Groth. Parallel solution-adaptive method for two-dimensional non-premixed combustng flows. *Prog. Comput. Fluid Dyn.*, 11(2):76–95, 2011.
- [34] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, and K. G. Powell. Three-dimensional MHD simulation of coronal mass ejections. *Adv. Space Res.*, 26(5):793–800, 2000.
- [35] C. P. T. Groth and S. A. Northrup. Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. Paper 2005-5333, AIAA, June 2005.
- [36] J. S. Sachdev and C. P. T. Groth. A mesh adjustment scheme for embedded boundaries. *Communications in Computational Physics*, 2(6):1095–1124, 2007.
- [37] Z. J. Zhang and C. P. T. Groth. Parallel high-order anisotropic block-based adaptive mesh refinement finite-volume scheme. Paper 2011-3695, AIAA, June 2011.
- [38] E.P.Boden and E. F. Toro. A combined chimera-amr technique for computing hyperbolic pdes. In *Proceedings of the Fifth Annual Conference of the CFD Society of Canada, Toronto, Ontario, Canada*, pages 5.13–5.18. CFD Society of Canada, 1997.
- [39] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *J. Comput. Phys.*, 90:1–64, 1990.
- [40] W. D. Henshaw. A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. *J. Comput. Phys.*, 113:13–25, 1994.
- [41] W. D. Henshaw and D. W. Schwendeman. An adaptive numerical scheme for high-speed reacting flow on overlapping grids. *J. Comput. Phys.*, 191:420–447, 2003.
- [42] B. van der Holst and R. Keppens. Hybrid block-AMR in cartesian and curvilinear coordinates: Mhd applications. *J. Comput. Phys.*, 226:925–946, 2007.
- [43] C. D. Norton and T. A. Ćwik. Parallel unstructured AMR and gigabit networking for beowulf-class clusters. In *Parallel Processing Applied Mathematics*, volume 2328 of *Lecture Notes in Computer Science*, pages 552–563, Berlin, 2002. Springer-Verlag.
- [44] R. C. Ripley, F. S. Lien, and M. M. Yovanovich. Adaptive mesh refinement of supersonic channel flows on unstructured meshes. *Int. J. Comput. Fluid Dyn.*, 18(2):189–198, 2004.
- [45] P. A. Cavallo, N. Sinha, and G. M. Feldman. Parallel unstructured mesh adaptation method for moving body applications. *AIAA J.*, 43(9):1937–1945, 2005.
- [46] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 65(1):359–392, 1998.
- [47] G. Karypis and V. Kumar. Metis - a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. University of Minnesota, Department of Computer Science / Army HPC Research Center, 1998.
- [48] D. F. Harlacher, H. Klimach, S. Roller, C. Siebert, and F. Wolf. Dynamic load balancing for unstructured meshes on space-filling curves. In *IEEE 26th International Parallel and Distributed Processing Symposium, Workshops, and PhD Forum*, pages 1655–1663, Washington, D. C., 2012. IEEE Computer Society.
- [49] K. W. Schulz and Y. Kallinderis. Unsteady flow structure interaction for incompressible flows using deformable hybrid grids. *J. Comput. Phys.*, 143:569, 1998.
- [50] M. H. Kobayashi, J. M. C. Pereira, and J.C.F. Pereira. A conservative finite-volume second-order-accurate projection method on hybrid unstructured grids. *J. Comput. Phys.*, 150:40, 1999.
- [51] D. Kim and H. Choi. A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids. *J. Comput. Phys.*, 162:411–428, 2000.
- [52] J. A. Shaw and A. J. Peace. Simulating three-dimensional aeronautical flows on mixed block-structured/semi-structured/unstructured meshes. *Int. J. Numer. Meth. Fluids*, 39:213–246, 2002.
- [53] Y. Ito and K. Nakahashi. Improvements in the reliability and quality of unstructured hybrid mesh. *Int. J. Numer. Meth. Fluids*, 45:79–108, 2004.
- [54] Y. Kallinderis and H. T. Ahn. Incompressible navier-stokes method with general hybrid meshes. *J. Comput. Phys.*, 210:75–108, 2005.
- [55] M. Darbandi and A. Naderi. Multiblock hybrid grid finite volume method to solve flow in irregular geometries. *Computational Methods in Applied Mechanics and Engineering*, 196:321–336, 2006.

- [56] N. Flandrin, H. Borouchaki, and C. Bennis. 3D hybrid mesh generation for reservoir simulation. *Int. J. Numer. Meth. Engin.*, 65:1639–1672, 2006.
- [57] Y. Ito, A. M. Shih, B. K. Soni, and N. Kazuhiro. Multiple marching direction approach to generate high quality hybrid meshes. *AIAA J.*, 45(1):162–167, 2007.
- [58] C. Kavouklis and Y. Kallinderis. Parallel adaptation of general three-dimensional hybrid meshes. *J. Comput. Phys.*, 229:3454–3473, 2010.
- [59] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. Paper 93-0668, AIAA, January 1993.
- [60] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. Paper 93-0880, AIAA, January 1993.
- [61] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.
- [62] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 25:294–318, 1988.
- [63] T. J. Linde. *A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere*. PhD thesis, University of Michigan, May 1998.
- [64] T. Linde. A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws. *Int. J. Numer. Meth. Fluids*, 40:391–402, 2002.
- [65] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1):25–34, 1994.
- [66] J. J. Gottlieb and C. P. T. Groth. Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. *J. Comput. Phys.*, 78:437–458, 1988.
- [67] C. Geuzaine and J.-F. Remacle. Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processor facilities. *Int. J. Numer. Meth. Engin.*, 79(11):1309–1331, 2009.