

AN OVERSET GRID BASED ADAPTIVE MESH REFINEMENT ALGORITHM

Mohamad Elhadj Ali Barada* and Bayram Celik†
Istanbul Technical University
Astronautical Engineering Department
34469, Istanbul, Turkey

ABSTRACT

A robust overset grid based adaptive mesh refinement approach is developed and implemented in C++ for 2-D geometries. It is extendable to 3-D geometries. With the aid of the developed approach, "halo" the algorithm generates an optimal overset mesh system. In the proposed approach, a volume mesh overcasting a system of body conformal meshes is integrated with a properly reduced overlap layer. Additionally a background mesh with holes is simply generated using the "flooding" approach. The developed adaptive mesh refinement approach takes the geometry proximity restrictions into account and then generates and links refinement boxes based on the quadtree. The algorithm is also designed for dynamic problems. The algorithm is integrated with an in-house solver where preliminary results for a benchmark study of a simple 2-D laminar supersonic flow is presented.

INTRODUCTION

With the increasing level of competitiveness in the aerospace market, companies and research departments are required to provide more optimal solution in terms of increased efficiency and performance and reduced operational and analysis costs. This is only achievable using more advanced computational fluid dynamics, CFD, methods and optimization tools that are capable of handling more complex and detailed solutions in less time, computational and human intervention wise. A critical part of CFD is the mesh generation and maintenance. Meshing techniques are divided into two parts, structured and unstructured. The former has proven itself to be very computational cost and solution accuracy wise friendly especially in resolving the highly viscous regions. However, when it comes to complex geometries it very time consuming to generate a suitable structured mesh. Even if generated, the mesh is very tedious to be maintained during changes applied to the geometry. On the other hand, utilization of unstructured meshes tend to resolve the complex geometry problem at the cost of increased storage space, more complex and expensive mesh generation techniques, such as delaunay triangulation, and reduced numerical prediction accuracy at the viscous regions. An alternative method used to both approaches is the top-to-bottom non conformal mesh grid, where a block mesh is generated covering the domain and geometry, including its interior. Split cells are generated from the intersecting cells at the geometry walls. A boundary layer is then inflated from the geometry surface using a surface mesh

*Ph.D. Candidate, Department of Aerospace Engineering, Email: mbarada@itu.edu.tr

†Assist. Prof. Dr., Department of Aerospace Engineering, Email: celikbay@itu.edu.tr

generated over the geometry surface via the cut cells. The inflation pushes the cells of the block mesh. This is done by solving elliptic equations analogous to the Laplace equation governing isotropic elastic materials, hence artificially creating deformation. Not only that the process is expensive, the quality of the surface mesh is very crucial to the generation of the boundary layer. Experience has shown that generation of such surface mesh is very difficult. Also improper inflation may cause negative volume cells.

A robust mesh generation and maintenance algorithm that blends the advantages of structured and unstructured meshes is desirable [Nagaram and Liou, 2009]. In 1983, Benek et al. proposed a meshing method where the domain consists of multiple block and body conformal meshes that exchange field information using an interpolation scheme [Benek et al., 1983]. This approach has proven to be very successful in terms of handling complex problems [Henshaw, 2002] [Chan, 2009]. As a result, it has become one of the core aspects of many CFD codes such as ONERA's ElsA, NASA's Overflow and Pegasus [Rogers et al., 2003]. Later, it was also further extended as a basis for the implementation of Adaptive Mesh Refinement, AMR, by others such as Meakin [Meakin, 2000] and Berger and Oliger [Berger and Oliger, 1984]. In their AMR algorithm, a finer block is added at regions where high gradient fields and/or numerical error are reported. Meakin's approach was later improved by Peron et al. [Peron and Benoit, 2013] where the governing phantom "brick" is replaced with a quadtree whose nodes represent the refinement boxes. The refinement boxes at the same level are merged using Rigby's algorithm that is based on the weakest decent method [Rigby et al., 1997] to reduce the total number of grid points and thereby required memory space.

This study presents the development and implementation of an overset based adaptive mesh refinement algorithm and its integration to the recently developed in-house finite volume supersonic Navier Stokes solver [Turk and Celik, 2017]. The generated overset grid is then used as a foundation to develop an AMR algorithm where a quadtree based block meshes or refinement boxes are generated, linked, and maintained for unsteady flow problems.

DATA STRUCTURE

In any overset grid generation process, the generated meshes are classified into two types. A mesh that is body conformal; covers the geometry and conforms to its topographic features and a background mesh; a mesh covering the domain and the geometry. It is generatable using an elliptic generator. In the developed code however, the provided mesh is imported from a third party utility such as OpenFOAM's blockMesh tool. The developed code then loads the imported mesh information into a new datastructure. Each cell in the mesh is provided a computational status presented by *iBlank* similar to the one originally proposed by Benek et al.. The *iBlank* can be 3 (orphan/blank type; cell flow properties do not get updated during time march), 1 (Compute type; cell flow properties get updated using the Navier Stokes solver) and 2 (Receiver type; cell flow properties get updated using the chosen interpolation scheme). In addition to this, each cell is given a boolean property called *donorFlag*; a property stating whether the cell is a donor or not. Also, in case if the cell is a receiver, it is given a pointer to an *interpolationGroup* object which is a pointer to an *interpolationStencil* object and also includes the interpolation coefficients. The relations are shown in the UML in Fig. 1. An interpolation stencil is a collection of 4 adjacent cells (Type 1), 2 adjacent cells (Type 2) or only a single cell (Type 3) that are/is used as a donor cell to provide information to the receiver cells. The interpolation stencils create the so called "dual mesh" which is associated with each cell as shown in Fig. 2.

Cells and interpolation stencils inherit the characteristics of a *Searchable* item; An interface allowing the inheritors to be searched and stored based on their bounding boxes in an alternating digital tree or quadtree structure. The alternating digital tree is constructed where each node, pointing to the

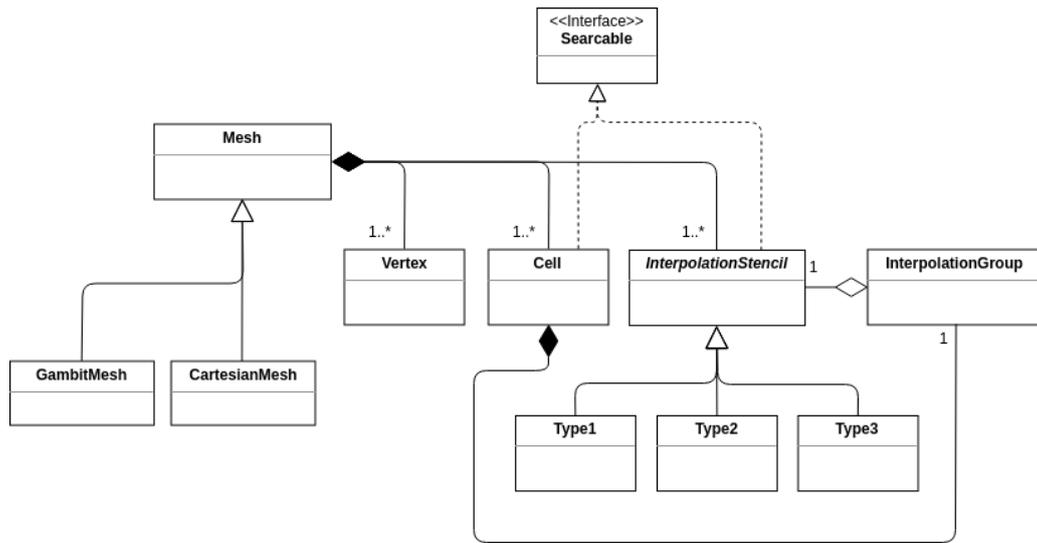


Figure 1: UML chart of the class relations integrating the overset grid developed datastructure.

searchable object, is allocated in the tree nodes according to the relative spatial position between the node and its parent node.

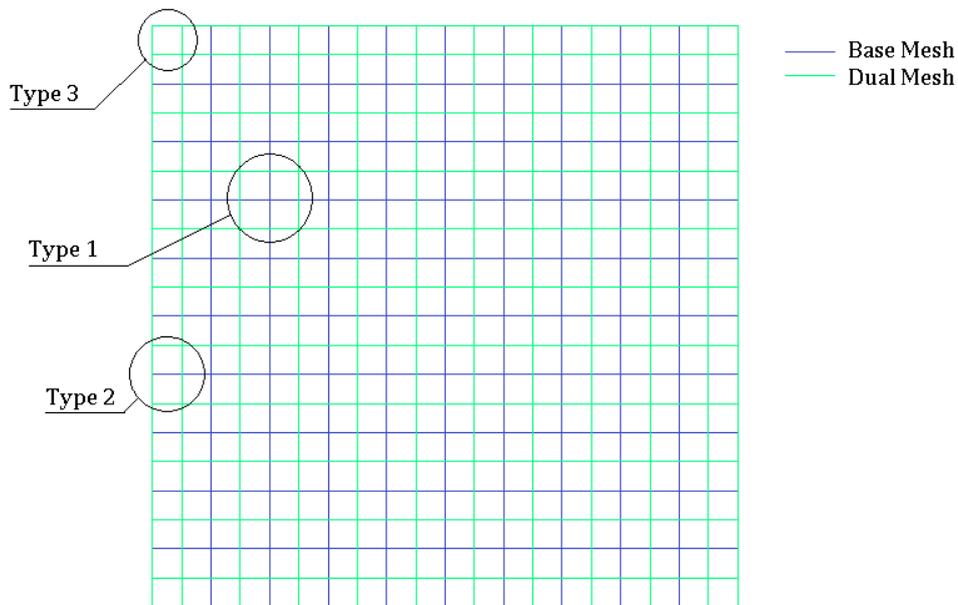


Figure 2: A blockMesh, blue grid, and its dual Mesh, green grid; The types of stencils are circled and labelled.

ALTERNATING DIGITAL TREE

In order speed up the spatial query process, an Alternating Digital Tree, ADT, is used to store each *Searchable* of a mesh. Referring to Fig. 3 the alternating digital tree is 4 hierachery leveled for the 2D case (3 in the figure). where each branching has a different spatial coordinate tested according to the hierachery level occupied as follows where the number presents the hierachery level (Criterion level in Fig. 3).

$$x_{b,min} \geq x_{a,min} \tag{1}$$

$$x_{b,max} \geq x_{a,max} \tag{2}$$

$$y_{b,min} \geq y_{a,min} \tag{3}$$

$$y_{b,max} \geq y_{a,max} \tag{4}$$

The ADT is also used to check for collision efficiently. The collision test used is the bounding box test. Referring to Fig. 4 and assuming cell B is being checked against a current mesh ADT node pointing to cell A, the inequalities (5) thru (8) are checked. If all the inequalities is satisfied, the node cell is marked as potential collider or donor, the search continues with both branches below. If however the failed criterion matches with the sorting criterion, the related branch is dropped while the other is carried over only with the spatial search. It is noted that a positive results of this test is not a sufficient condition in terms of occurrence of collision between two searchables. It is rather a "proximity alert" that shall show its advantages in the following section.

$$x_{a,min} \leq x_{b,max} \tag{5}$$

$$y_{a,min} \leq y_{b,max} \tag{6}$$

$$x_{a,max} \geq x_{b,min} \tag{7}$$

$$y_{a,max} \geq y_{b,min} \tag{8}$$

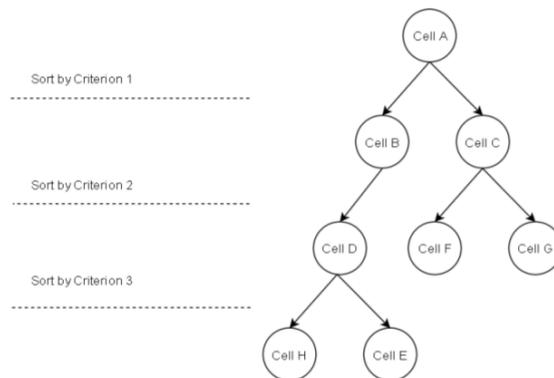


Figure 3: A generic alternating digital tree.

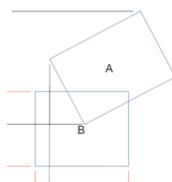


Figure 4: Two *Searchables* tested for collision using the bounding box test.

OVERSET GRID ALGORITHM

Referring to Figure 5, The overset Grid starts by marking background cells that potentially collide with the solid walls (a-b). By selecting cells on the interior of the geometry is selected where the "flooding" process is initiated (c). The flooding process is recursive where cells from the prior geometry interior selected neighbours are eliminated and then used to obtain the next "to be eliminated" cells group. The outer most cells, receiver cells, of the body conformal mesh are used to mark or imprint holes over the background mesh (d). The outline of the eliminated cells is inflated until hugging the halo thereby achieving a well optimized overlap layer between the background and body conformal meshes (e). In order to establish communication between the grids a non conservative binlinear interpolation is used.

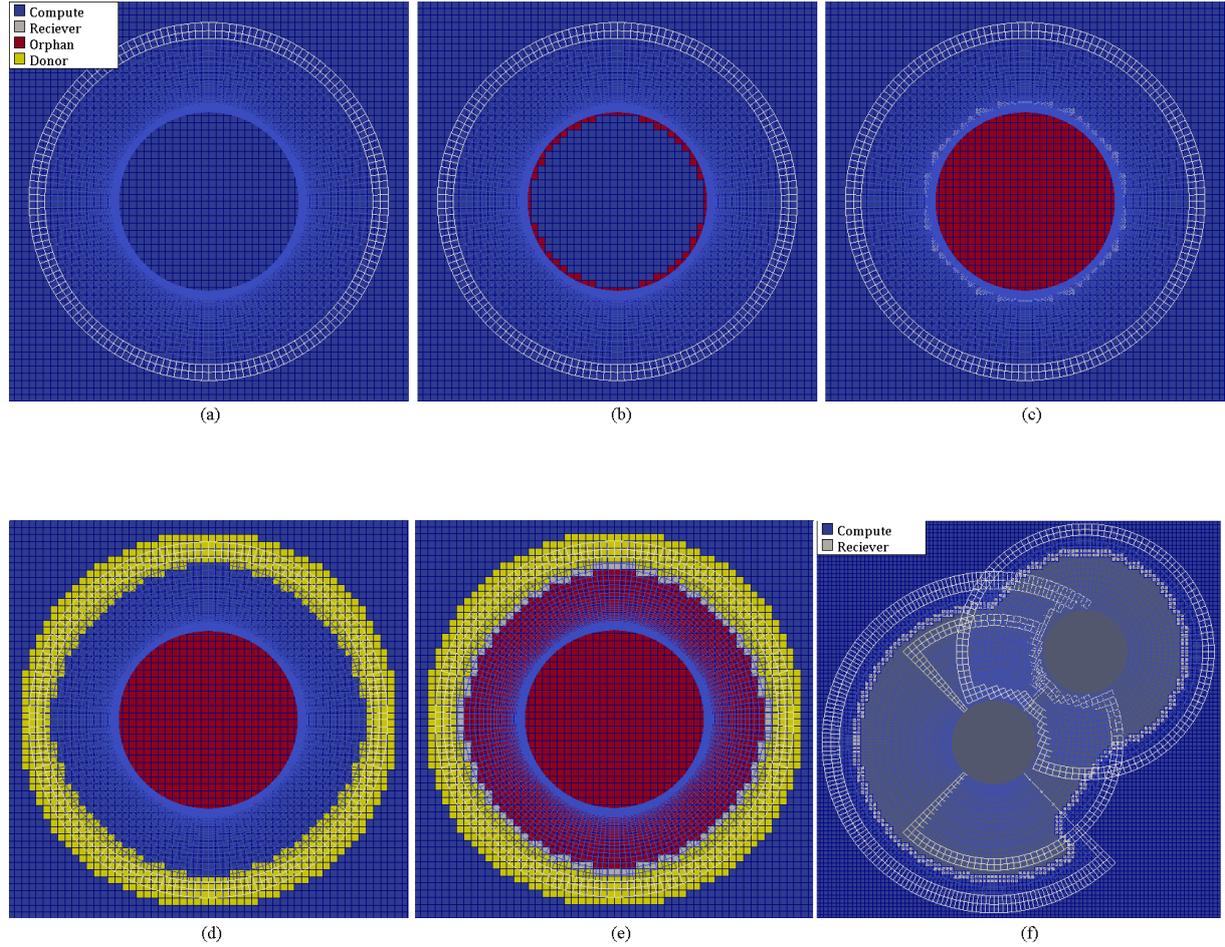


Figure 5: The overset Grid algorithm (a thru e) and its application on a representative complex geometry case (f).

The donor cells used and the interpolation coefficients are determined using Newton Raphson method where convergence is guaranteed within 5 iterations. These donor cells are determined from the yellow cells which are only a potential donor. Potential interpolation stencils are used to generate a shape function in terms of η and ξ ,

$$N_i = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) \quad (9)$$

The generated function for each interpolation stencil is tested where the correspondent η and ξ for the receiver cell center are obtained using Newton Raphson; This method converges within 5 steps.

$$f_x(\xi, \eta)^{(k)} = x_i - \sum_{j=1}^4 x_j N_j(\xi^{(k)}, \eta^{(k)}) \quad (10)$$

$$f_y(\xi, \eta)^{(k)} = y_i - \sum_{j=1}^4 y_j N_j(\xi^{(k)}, \eta^{(k)}) \quad (11)$$

If the obtained coefficients are within the following range then the interpolation stencil is assigned to the receiver cell by generating an *interpolationGroup* object. Otherwise, it is discarded.

$$-1 \leq \xi \leq 1 \quad (12)$$

$$-1 \leq \eta \leq 1 \quad (13)$$

The *interpolationGroup* object also stores the η and ξ coefficients values for the bilinear interpolation stage during the computational run.

$$B(\xi, \eta) = \sum_{i=1}^4 B_i N_i(\xi, \eta) \quad (14)$$

In order to speed up the process the alternating digital tree that sorts the cells and potential interpolation boxes based on their bounding box spatial coordinates is used.

The flooding and halo methodologies do not rely on the dimensionality of the problem. As a result, this approach can be extended to the 3D problems easily.

The approach is extended to complex systems of separate geometries fitted with intersecting body of conformal meshes. The hole generation process is applied, in addition to the background mesh, between the body conformal meshes. If any of the body conformal mesh cells is a wall cell, the hole blanking process is reverted. The receiver layer of each body conformal mesh is established after hole generation of body to body conformal meshes. During the background mesh potential donor flagging, the cells are flagged from a body conformal mesh only if there is no other suitable interpolation stencil from any other body conformal meshes. Figure (f) shows the implementation of the outline algorithm on an overset grid system that involves two interacting cylinders where one of which has two non enclosed body conformal meshes.

ADAPTIVE MESH REFINEMENT ALGORITHM

In comparisons to generic overset grid based computational run, the AMR run utilizes an algorithm that generates a background mesh accompanied by quadtree that is generated with a single node enveloping the domain. A quadtree data structure is generated over a geometry with refinement levels based on the spatial proximity of the nodes to the surface. Each node is used to generate a grid. The generated grids, also called refinement boxes are interconnected by an interpolation function. Figures 6 and 7 show such background mesh. It is noted that in the latter figure, the refinement boxes, or nodes, are smaller without comprising the refinement level. This allows for smaller transition region between different refinement levels during the AMR process.

During the computational run, the difference in the gradients of a flow property such as the pressure or temperature is measured using a sensor function. The current code uses the second order undivided

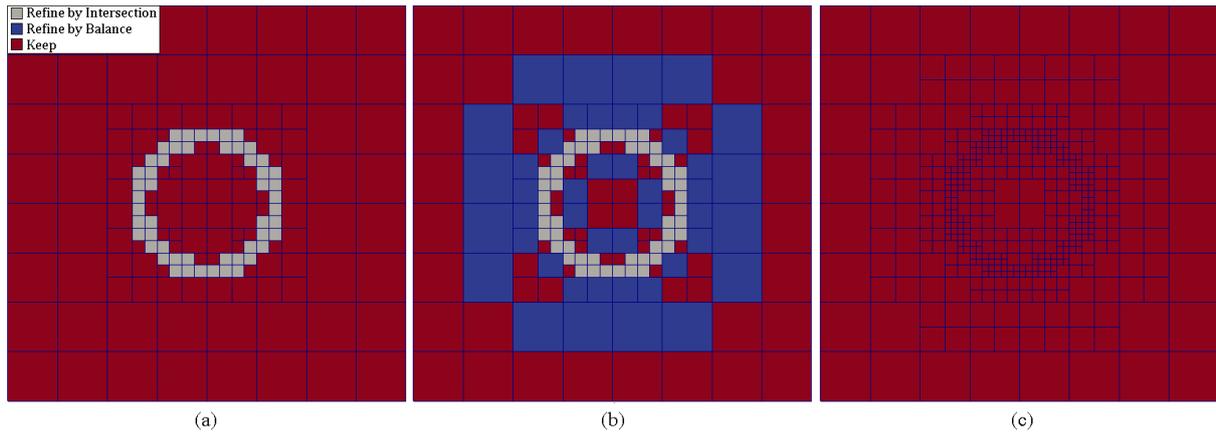


Figure 6: The marking process of the quadtree nodes based on (a) intersection and (b) balancing requirements and (c) resultant quadtree after the refinement process.

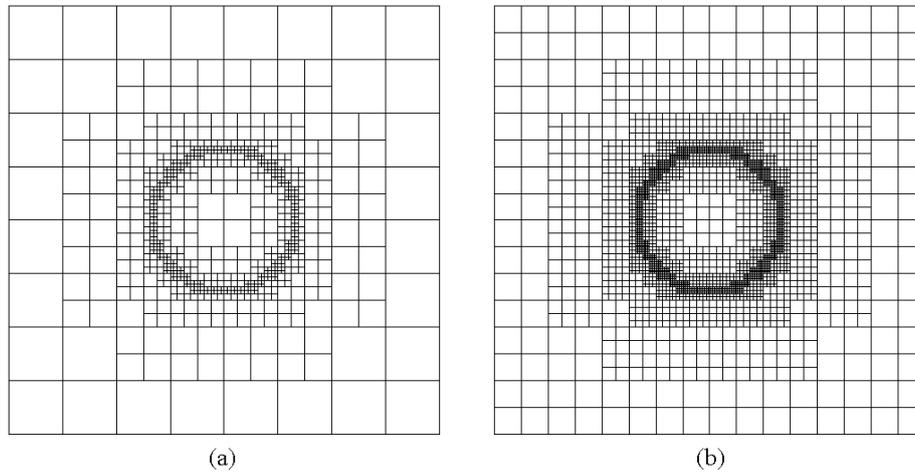


Figure 7: Generated quadtrees with different uniform refinement level but same cell count.

difference, Eq. 15. The obtained value is compared to a threshold range that is established using a generic bisection method proposed by Peron et al. which takes into account the maximum allowable cell count. If the obtained value exceeds the range, the governing node is marked for refinement, whereas if it undershoots the range, the governing node is marked for coarsening. If the value is within range, the node is marked as Keep; no refinement or coarsening is done.

$$q = \max \left\{ \frac{B_{k-1} - 2B_k + B_{k+1}}{2} \right\}^2 \Bigg|_{k=i,j} \quad (15)$$

Referring to Figure 8, the obtained flagged quadtree, during a 2D supersonic cylinder run, after the application of the sensor function requires some processing. Such processing takes into account the maximum refinement level constraint and the balancing requirements. The process starts by investigating the nodes neighbouring the geometry (Marked with the circle in (a)) These nodes can be of refinement level of R_t or $R_t - 1$. If as refine, the are changed to keep; this is the geomtric constraint where the result is (b). Level constraint is then applied where any node marked as refine and of refinement level of R_t is marked to keep; The result is the leve constraint applied (c).

Then any coarsen nodes beside the geometry that have inactive siblings or parents (Within the geomtry interior) are marked as keep otherwise left as refine (d). Finally the neighbours of the refine marked

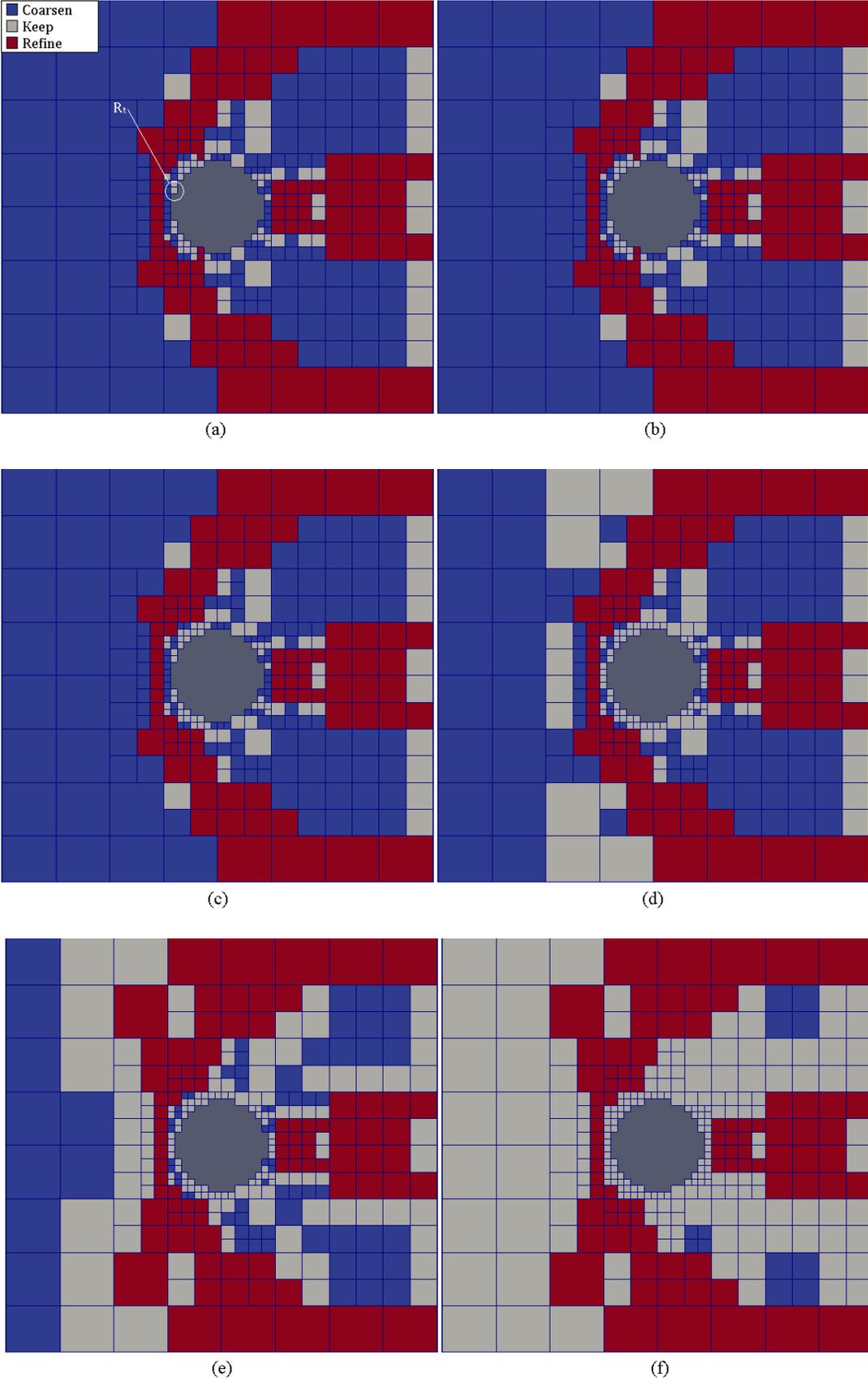


Figure 8: The AMR process.

nodes are checked, if they are of lower refinement level, they are marked for coarsening. This step is recursively repeated (e). Then, the neighbours of the keep nodes are checked where if they are of lower refinement level and are marked as coarsen they are marked as keep. This step is also repeated (f).

This approach does not only take into account the refinement process, but also allows for the coarsening of regions exhibiting low gradients. As a result, it is easily applicable to dynamic solutions.

With the quadtree processed, the nodes are refined and coarsened accordingly, a new multi-block structured mesh is generated and the solution is mapped onto it from the older background grid using bilinear interpolation, ADT and Newton Raphson.

SOLUTION METHODOLOGY

In order to quantify the performance of the developed Overset based Adaptive Mesh Refinement algorithm, it is integrated with an in-house cell centered finite volume Navier Stokes Solver [Turk and Celik, 2017]. The solver is coded with C++ in an object oriented fashion where the following set of equations in tensor notation are solved numerically,

$$\frac{\partial \rho u_i}{\partial x_i} = 0 \quad (16)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \quad (17)$$

$$\frac{\partial \rho e}{\partial t} + \frac{\partial \rho e u_i}{\partial x_i} = -\frac{\partial p u_i}{\partial x_i} + \Phi + \frac{\partial}{\partial x_i} \left(\kappa \frac{\partial T}{\partial x_i} \right) \quad (18)$$

$$p = \rho R T \quad (19)$$

where τ_{ij} , Φ and e are the stress tensor, dissipation by viscosity and the total energy. The viscosity is modeled using Sutherland's law. Van Leer flux splitting scheme is used for flux reconstruction on the faces. The solver uses a first order scheme for convection terms and a second order central scheme for the diffusion terms. Time integration is based on first order forward difference.

CASE STUDY: SUPERSONIC VISCOUS FLOW OVER A 2-D CYLINDER

The developed overset grid based AMR solver is used to solve a supersonic viscous flow over a 2-D cylinder where the flow properties are tabulated in Table 1.

Mach	2.9
Reynolds Number	7.2×10^5
γ	1.4
Temperature, K	300

Table 1: Supersonic flow freestream conditions

After conducting a mesh independency test, the run is performed on a mesh with a resolution of 8800 cells (initial) at background and body conformal mesh with 15448 cells as well as 500 iterations per AMR cycle. The initial AMR mesh is used where it is shown in Fig. 9 and has the geometric details presented in Table 2. Supersonic boundary conditions are applied where the left face is set to free stream conditions and the upper, lower and right faces to zero gradient. The wall is set to no-slip boundary condition. Note that the square domain has a height/width to cylinder diameter ratio of 6. The case is run on a machine with a single Intel Quad Core i5 2.67Ghz and 4 GigaBytes of RAM.

Table 2: Generated AMR mesh characteristics

	Value
Near Body Refinement Level	8
Uniform Refinement Level	0
Grid Size per Node	10×10
Cell Count	
Volume Mesh	8800
Body Conformal Mesh	15448

The threshold function used is based on the flow pressure, $B = p$ in Eq 15. Bi-sectioning method is used to obtain threshold range where the dictating factor is the mesh growth factor. The AMR settings are outlined in Table 3.

Table 3: AMR run settings.

	Value
Target Mesh Ratio	5.5
Growth Limiter	1.5
Ratio Tolerance	± 0.05
Iterations per AMR Cycle	500
Total Iterations	6000

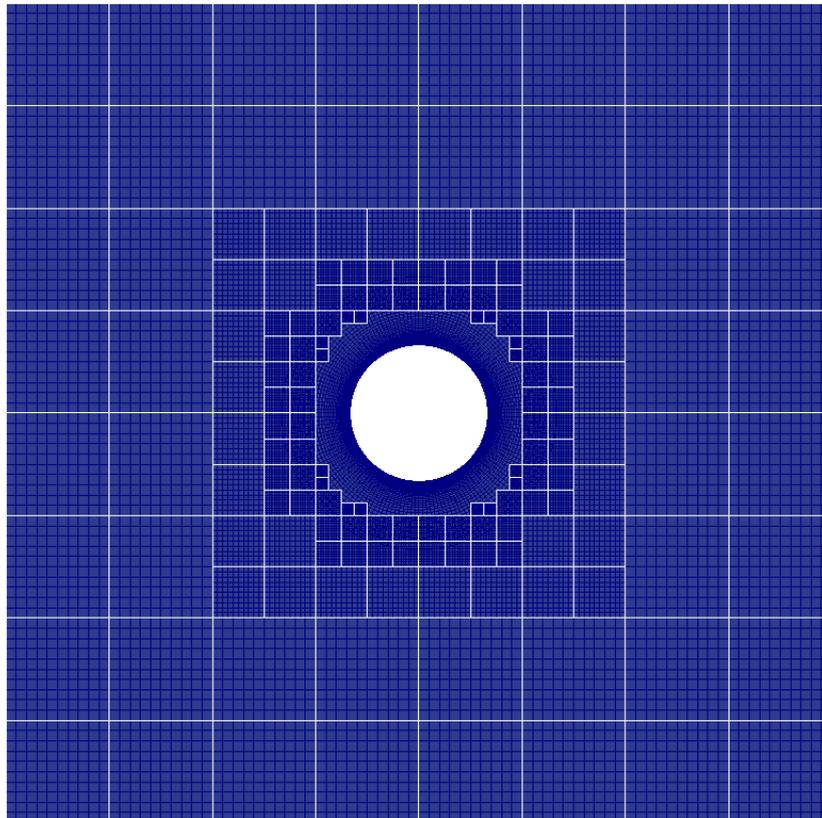


Figure 9: Initial overset grid mesh.

Figure 10 shows the obtained Mach contour and the predicted coefficient of pressure distribution over the cylinder surface in comparison with the experimental study of Gowen and Perkins [1953]. The

refinement criterion is based on the pressure gradient.

From the plot, The AMR has yielded a stagnation pressure of 1.8 which is 2.3% higher than the experimental result of 1.759. The computed C_p distribution over the cylinder is in good agreement with the reference study. The deviation between the results may result from the lack of a turbulence model. The developed AMR approach has dynamically changed the background mesh resolution such that the bow shock has been resolved properly.

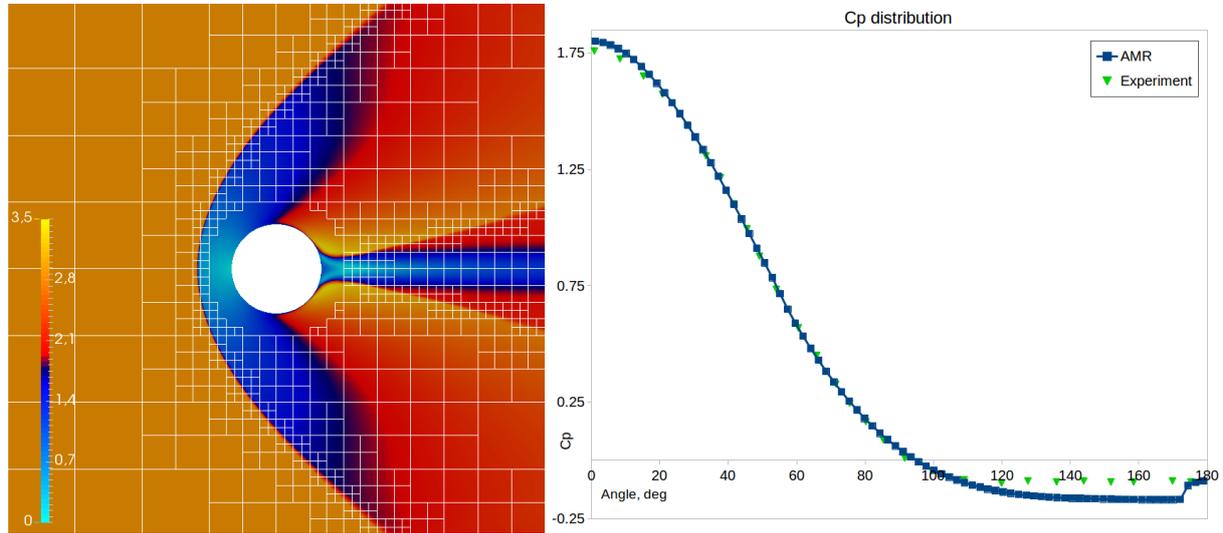


Figure 10: Achieved Mach contour (left) and C_p plots (right).

CONCLUSIONS

A robust overset grid based adaptive mesh refinement algorithm that is unrestricted to dimensionality is developed. The algorithm uses "halo" and "flooding" approaches to generate the holes over a background mesh and then optimize the overlapping layer with the body conformal grids. Alternating digital tree and Newton Raphson method are used for spatial search and in obtaining the interpolation coefficients, respectively. The developed code is shown to be able to handle meshes for complex geometies. The code is extended to apply AMR by using a quadtree based background mesh. A balancing and AMR operation manipulation algorithm is introduced and tested. The developed algorithm is implemented using C++ and integrated to an in-house solver where it is tested against a generic supersonic flow over a cylinder. Comparison of the results obtained from the developed algorithm with the results in the literature show that the developed algorithm is robust and efficient in terms of capturing the location and the strength of the shock as well as transferring the field information between meshes. It gives satisfactory stagnation pressure prediction (2.3% error) and properly captures the C_p distribution over the cylinder body in general.

References

- Benek, J., Steger, J., and Dougherty, F. C., "A flexible grid embedding technique with application to the Euler equations," *6th Computational Fluid Dynamics Conference Danvers*, (1983), p. 59-69.
- Berger, M. J., and Olinger, J., "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of computational Physics*, Vol. 58, No. 3, (1984), pp. 484–512.
- Chan, W. M., "Overset grid technology development at NASA Ames Research Center," *Computers & Fluids*, Vol. 38, No. 3, (2009), pp. 496–503.
- Gowen, F. E., and Perkins, E. W., "Drag of circular cylinders for a wide range of Reynolds numbers and Mach numbers," (1953), p. 19.
- Henshaw, W., "Adaptive Mesh and Overlapping Grid Methods," *Encyclopedia of Aerospace Engineering*, (2010).
- Meakin, R. L., "Adaptive spatial partitioning and refinement for overset structured grids," *Computer methods in applied mechanics and engineering*, Vol. 189, No. 4, (2000), pp. 1077–1117.
- Nagaram, M., and Liou, M.-S., "A Cartesian based body-fitted adaptive grid method for compressible viscous flows," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, (2009), p. 2.
- Peron, S., and Benoit, C., "Automatic off-body overset adaptive Cartesian mesh method based on an octree approach," *Journal of Computational Physics*, Vol. 232, No. 1, (2013), pp. 153–173.
- Rigby, D., Steinthorsson, E., Coirier, W., Rigby, D., Steinthorsson, E., and Coirier, W., "Automatic block merging methodology using the method of weakest descent," *35th Aerospace Sciences Meeting and Exhibit*, (1997), p. 97–197.
- Rogers, S. E., Suhs, N. E., and Dietz, W. E., "PEGASUS 5: an automated preprocessor for overset-grid computational fluid dynamics," *AIAA journal*, Vol. 41, No. 6, (2003), pp. 1037–1045.
- Turk, A. G., and Celik, B., "A 2-D finite volume Navier-Stokes solver for supersonic flows," *Anadolu Üniversitesi Bilim Ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik*, Vol. 18, No. 5, (2017), pp. 1042–1056.