AIAA AVIATION Forum
5-9 June 2017, Denver, Colorado
23rd AIAA Computational Fluid Dynamics Conference

AIAA 2017-4113

Block-Based Anisotropic AMR with *A Posteriori* Adjoint-Based Error Estimation for Three-Dimensional Inviscid and Viscous Flows

N. Narechania* , L. Freret† and C. P. T. Groth‡

University of Toronto Institute for Aerospace Studies

4925 Dufferin Street, Toronto, Ontario, M3H 5T6, Canada

Downloaded by Clinton Groth on June 27, 2017 | http://arc.aiaa.org | DOI: 10.2514/6.2017-4113

An anisotropic output-based adaptive mesh refinement scheme is proposed for the numerical prediction of inviscid and viscous flows on three-dimensional multi-block meshes using parallel distributed-memory computer architecture. A finite-volume discretization procedure with limited piecewise linear reconstruction is used in combination with a second-order implicit time-marching algorithm to solve the governing partial differential equations on body-fitted hexahedral meshes. The anisotropic block-based refinement provides significant reductions in the size of the computational mesh by locally refining the grid only in selected directions as dictated by the flow physics and the predicted solution. The adjoint-based error estimation enables formal evaluation of *a posteriori* estimates of the errors in solution-dependent engineering functionals in terms of local estimates of the truncation error as measured by the solution residual error. These errors are calculated by solving an adjoint problem related to the functional of interest and using the solution of the adjoint problem to appropriately weigh primal flow quantity residual errors evaluated on a finer mesh using an *h*-refinement strategy. The resulting dual-weighted error estimate is used to direct the local mesh adaptation and the error-driven refinement strategy creates meshes which are customized for the accurate calculation of the functionals of interest. The performance of the output-based mesh refinement scheme is demonstrated for several representative steady time-invariant inviscid flows governed by the Euler equations and viscous flows governed by the Navier-Stokes equations.

I. Introduction

I.A. Motivation

As the use of computational fluid dynamics (CFD) has become more routine with a greater breadth of applications, the need for more efficient and accurate numerical techniques has also grown. In particular, flows with a wide range of length and time scales present an inherent difficulty to numerical methods due to the resulting stiffness of the governing partial differential equations. Common examples of such flows are: (i) chemically reactive flows; (ii) turbulent flows; (iii) aerodynamic flows with detailed three-dimensional (3D) wing-fuselage configurations; (iv) conducting flows involving electromagnetic phenomena; and (v) micro-scale non-equilibrium flows.

There are several ways to decrease the computational cost of performing a simulation of multi-scale flows without simplifying the mathematical models or compromising the accuracy of the computed solution. One approach is to make use of adaptive mesh refinement (AMR),[1–6] wherein the number of computational cells is increased only in regions involving smaller length scales and correspondingly reduced in regions with solution content having much larger length scales. In this way, the AMR is able to reduce both the overall size of the computational mesh for a desired resolution as well as the computational cost of performing a simulation, while also taking into account the possible disparate length scales present in the flow. It also automates the

[*] Ph.D. Candidate, Email: nishant.narechania@mail.utoronto.ca and AIAA Student Member.
[†] Post-doctoral fellow, Email: lfreret@utias.utoronto.ca.
[‡] Professor, Email: groth@utias.utoronto.ca and AIAA Senior Member.

adaption of the computational meshes which greatly reduces the need for human intervention. Additionally, AMR is potentially highly advantageous for unsteady problems where different regions of the flow domain may display complex phenomena at different instances in time.

Solution-dependent AMR strategies can be broadly classified into four categories:

1. patch-based AMR (e.g, Berger and Colella,[1] Quirk and Hanebutte,[7] Pantano *et al.*[8]) in which individual cells of an initially coarse mesh are dynamically flagged and grouped together to form rectangular patches;

2. cell-based AMR (e.g., De Zeeuw and Powell,[9] Aftosmis *et al.*,[6] Berger and LeVeque,[10] Ripley *et al.*[11]) where each cell is refined individually and the cell connectivity is typically stored in a tree data structure;

3. block-based AMR (e.g., Berger and Saltzman,[2] Groth *et al.*[12, 13]) in which the computational domain is represented by a multi-block mesh with subdomains or "blocks" containing a pre-defined number of cells and during refinement, the blocks are subdivided into a number of blocks, each block containing the same pre-defined number of cells; and

4. hybrid AMR (e.g., Holst and Keppens[14]) which involves a combination of patch-based and block-based AMR methods.

The parallel block-based AMR techniques proposed by Groth and co-workers[15–21] are of particular interest here. They provide an effective treatment for the issues associated with disparate scales and have been shown to lead to highly scalable on distributed memory parallel computers. Isotropic block-based AMR, as originally proposed by Gao and Groth,[15, 18, 19] is based on an octree data structure wherein each block flagged for refinement is refined equally in all directions. The isotropic block-based AMR methods have been applied quite extensively to combustion problems by Groth and co-workers. For example, Northrup and Groth[16, 22, 23] used the 3D isotropic block-based AMR for simulation of laminar steady and unsteady premixed and non-premixed flames. Gao and Groth[15, 18, 19, 24–26] and Jha[27, 28] applied isotropic block-based AMR to turbulent diffusion flames. Ivan *et al.*[29–31] and Susanto *et al.*[32] have also used block-based AMR for magnetohydrodynamic simulations. An anisotropic version of the block-based AMR based on a binary-tree data structure was more recently proposed by Zhang and Groth[33] so as to increase further the computational savings. When applied to convection-diffusion and the inviscid flow equations, significant mesh savings were found. Williamschen and Groth[34] extended this anisotropic block-based AMR to 3D simulations of inviscid flows governed by the Euler equations. They used a uniform block approach where the ghost cells for a block had the same refinement level as the block itself. More recently, an anisosotropic block-based approach making use of heterogeneous non-uniform blocks was proposed by Freret and Groth.[21] The latter has several advantages making it more efficient and better suited for high-order spatial discretizations. Note also that the extension of the anisotropic AMR approach for use with the high-order central essentially non-oscillatory (CENO) finite-volume scheme is considered in the recent study by Freret *et al.*[35]

In most of the aforementioned studies of block-based AMR studies,[15–21] heuristic physics- and/or gradient-based criteria are used to direct the mesh refinement. These approaches are easily implementable and work well for many of the problems studied. However, as discussed by Williamschen and Groth[34] and Freret and Groth,[21] gradient-based methods do not necessarily directly respond to errors in the numerical solution, hampering grid convergence and error control as the mesh refinement proceeds. Mesh refinement criteria based on solution error would make the AMR significantly more effective.

In particular, adjoint-based error estimation techniques would enable the use of a formal error estimate for directing mesh adaptation and providing sensitivities of engineering functionals to this error. Adjoint methods for AMR have been considered by previous researchers.[36–43] For example, Heuveline and Rannacher[36] applied adjoint-based error estimation in combination with AMR to the solution of second-order elliptic equations. Becker *et al.*[37] applied similar methods for problems involving computation of aerodynamic forces. Venditti and Darmofal[38–40] used error-estimation strategies to adapt unstructured triangular meshes for computing two-dimensional (2D) flows over airfoils. Nemec and Aftosmis[41, 42] used the solution of the discrete adjoint for performing mesh refinement on 3D polyhedral meshes for compressible inviscid flows using cell-based AMR. Ceze and Fidkowski[43] used adjoint methods for mesh and polynomial-order adaptation for Navier-Stokes and RANS flows using cell-based AMR on body-fitted meshes(coarsening was not permitted in their study). Finally, Yano and Darmofal[44] have also used adjoint methods for performing anisotropic mesh adaptation on simplex meshes for advection-diffusion problems. A key advantage of these error-based refinement methods is that they, unlike gradient-based methods, can respond directly to estimates of the solution error and they can be used for isolating only those regions of the flow which are sensitive to the functional or output of interest. This results in specially tuned AMR meshes which are customized for minimizing errors and accurately evaluating engineering functionals of particular interest.

I.B. Scope of Research

To overcome the shortcomings shown by previous gradient-based strategies for directing block-based AMR methods, a new output-based AMR strategy is proposed which employs the *a posteriori* error-estimation technique similar to that originally developed by Venditti and Darmofal[38–40] combined with the recent anisotropic AMR scheme of Freret and Groth.[21] In the proposed combined approach, the solution of the adjoint for a functional of interest is used to weigh estimated local errors in the solution residual thereby selecting regions of the computational domain that most directly influence the functional. The proposed method generates locally refined meshes which are customized for the accurate evaluation of the engineering functional of interest. Examples of engineering functionals considered here include aerodynamic forces such as lift and drag.

In what follows, the solution methodology comprising the governing equations, the second-order finite-volume scheme, inexact Newton method, and the parallel block-based AMR approach are all first described.[21,22] The formulation and solution of the adjoint problem, calculation of error indicators and error-driven mesh adaption are described next. Finally, numerical results are discussed for several steady inviscid compressible flow problems governed by the Euler equations and steady viscous compressible flow problems governed by the Navier-Stokes equations, obtained using the proposed AMR scheme, allowing the relative performance of the proposed output-based anisotropic AMR method to be assessed. Comparisons with uniformly refined mesh and physics-based AMR are also shown.

II. Solution Methodology

II.A. Governing Equations

The conservative form for the governing equations describing the compressible gaseous flows of interest can be written in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \vec{\mathbf{F}} = \mathbf{0}, \tag{1}$$

where $\mathbf{U}$ is the vector of conserved variables and $\vec{\mathbf{F}}$ is the flux vector. For a three-dimensional Cartesian geometry, Eq. (1) can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{0}, \tag{2}$$

where $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$ are fluxes in the x, y, z directions, respectively. For viscous flows governed by the Navier-Stokes equations, these fluxes can be further classified into inviscid and viscous components. Consequently, the Eq. (2) can be rewritten as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial (\mathbf{F}_{\mathrm{I}} - \mathbf{F}_{\mathrm{V}})}{\partial x} + \frac{\partial (\mathbf{G}_{\mathrm{I}} - \mathbf{G}_{\mathrm{V}})}{\partial y} + \frac{\partial (\mathbf{H}_{\mathrm{I}} - \mathbf{H}_{\mathrm{V}})}{\partial z} = \mathbf{0}, \tag{3}$$

where the solution and inviscid flux vectors are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \mathbf{F}_{\mathrm{I}} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho u h \end{bmatrix}, \mathbf{G}_{\mathrm{I}} = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ \rho v h \end{bmatrix}, \mathbf{H}_{\mathrm{I}} = \begin{bmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ \rho w h \end{bmatrix}. \tag{4}$$

and where ρ is the density, u, v, w are x, y, z velocity components, respectively, $e = p/(\rho(\gamma - 1)) + u^2/2$ is the specific total energy and $h = e + p$ is the specific enthalpy. The ideal gas equation $p = \rho RT$ is used for closure of the system where T is the gas temperature and R is the ideal gas constant. The ratio of specific heats $\gamma = C_p/C_v$ is assumed to be constant. The viscous fluxes are given by

$$\mathbf{F}_{\mathrm{V}} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ -q_x + u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \end{bmatrix}, \mathbf{G}_{\mathrm{V}} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ -q_y + u\tau_{yx} + v\tau_{yy} + w\tau_{yz} \end{bmatrix}, \mathbf{H}_{\mathrm{V}} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ -q_z + u\tau_{zx} + v\tau_{zy} + w\tau_{zz} \end{bmatrix}. \tag{5}$$

where q_x, q_y, q_z are the heat flux components and τ_{xx},τ_{xy},τ_{xz},τ_{yy},τ_{yz},τ_{zz} are the components of the stress tensor. For inviscid flows, the components of the viscous vectors $\mathbf{F}_{\mathrm{V}}$, $\mathbf{G}_{\mathrm{V}}$ and $\mathbf{H}_{\mathrm{V}}$ are identically zero and the governing equations reduce to the usual Euler equations.

II.B. Finite-Volume Spatial Discretization and Semi-Discrete Form

The spatial discretization is accomplished herein by using a second-order cell-centered finite-volume scheme. Following application of this finite-volume method to a hexahedral computational cell or element, (i, j, k), of a structured three-dimensional grid, Eq. (1) can be re-expressed in semi-discrete form as

$$\frac{\mathrm{d}\mathbf{U}_{i,j,k}}{\mathrm{d}t} = -\mathbf{R}_{i,j,k}(\mathbf{U}) = -\frac{1}{V_{i,j,k}} \sum_{f=1}^{N_{f_{i,j,k}}} \left(\vec{\mathbf{F}}_f.\vec{\mathbf{n}}_f \Delta A_f\right)_{i,j,k}, \tag{6}$$

where $\mathbf{U}_{i,j,k}$ is the averaged conserved solution for cell (i, j, k), and $\mathbf{R}_{i,j,k}$ is the discrete residual representing the summation of the face fluxes for cell (i, j, k). The variables $V_{i,j,k}$, $\vec{\mathbf{F}}_f$, $\vec{\mathbf{n}}_f$ and ΔA_f denote the cell volume, flux vector, outward pointing unit normal vector and the area of the cell face, f, respectively, and $N_{f_{i,j,k}}$ is an integer value representing the number of faces for cell (i, j, k). A schematic diagram illustrating a hexahedral cell is provided in Figure 1(a). For the evaluation of the inviscid fluxes, limited piecewise-linear least-squares reconstruction is used for calculating primitive flow variables at the cell faces. A Godunov-type flux function,[45] namely the so-called HLLE approximate Riemann solver based flux function proposed by Einfeldt,[46] is then used for evaluation of the inviscid fluxes at the cell faces. Viscous fluxes involving both the solution and its gradients are calculated using a Green-Gauss integration procedure[16] at the cell faces.

II.C. Inexact Newton's Method

An inexact Newton's method is used to solve the coupled system of non-linear algebraic equations describing steady solutions of the semi-discrete form of the governing equations above for all computational cells in the mesh given by

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = -\mathbf{R}(\mathbf{U}) = 0. \tag{7}$$

These non-linear equations are linearized through the application of Newton's method to arrive at the following system of linear equations

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right] \Delta \mathbf{U}^{(n)} = \mathbf{J}\Delta \mathbf{U}^{(n)} = -\mathbf{R}(\mathbf{U}^{(n)}), \tag{8}$$

where $\mathbf{J} = \partial \mathbf{R}/\partial \mathbf{U}$ is the Jacobian of the residual and $\mathbf{U}^{(n+1)} = \mathbf{U}^{(n)} + \Delta \mathbf{U}^{(n)}$. Thus, for a given initial estimate of the solution, $\mathbf{U}^{(n=0)}$, an improved approximation, $\mathbf{U}^{(n+1)}$, is obtained by solving the system of linear equations at each step, n, of Newton's method. The linear system of equations is solved iteratively until the solution residual is sufficiently small below a user-defined tolerance level.

Equation (8) is of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{9}$$

and can be solved using an iterative linear solver. Such systems are typically very large, non-symmetric, and associated with sparse banded matrices. For large sparse systems, it is economical to use Krylov subspace methods. The Generalized Minimum Residual (GMRES) method, a class of Krylov subspace methods, initially developed by Saad and Schultz[47] is used here. GMRES is implemented here in a parallel fashion allowing use of parallel computer architectures having multiple processors using the Message Passing Interface (MPI) library of subroutines.[48,49] Refer to the paper by Northrup and Groth[22] for further details of the parallel Newton method used here.

II.D. Anisotropic Block-Based AMR

The mesh refinement strategy used here is a parallel anisotropic block-based AMR scheme which allows automatic local mesh adaptation on a parallel multi-block body-fitted mesh consisting of three-dimensional hexahedral computational cells. An example of a body-fitted multi-block mesh generated using block-based AMR is illustrated in Figure 1(b). The computational domain is divided into subdomains called “blocks”, each block containing a predefined number of cells. During isotropic refinement, a block marked for refinement is subdivided uniformly into four (2D) or eight (3D) blocks, each containing the same pre-defined number of cells, although the volume of each cell is reduced by a factor of four in 2D simulations or by a factor of eight in 3D simulations. During anisotropic refinement, a block is always divided into two blocks, in a particular direction, doubling the mesh spacing or resolution only in that preferred direction. The anisotropic block-based AMR scheme adopted here[21,33–35] has been found to be highly efficient in terms of reducing the overall mesh size for a given flow problem.

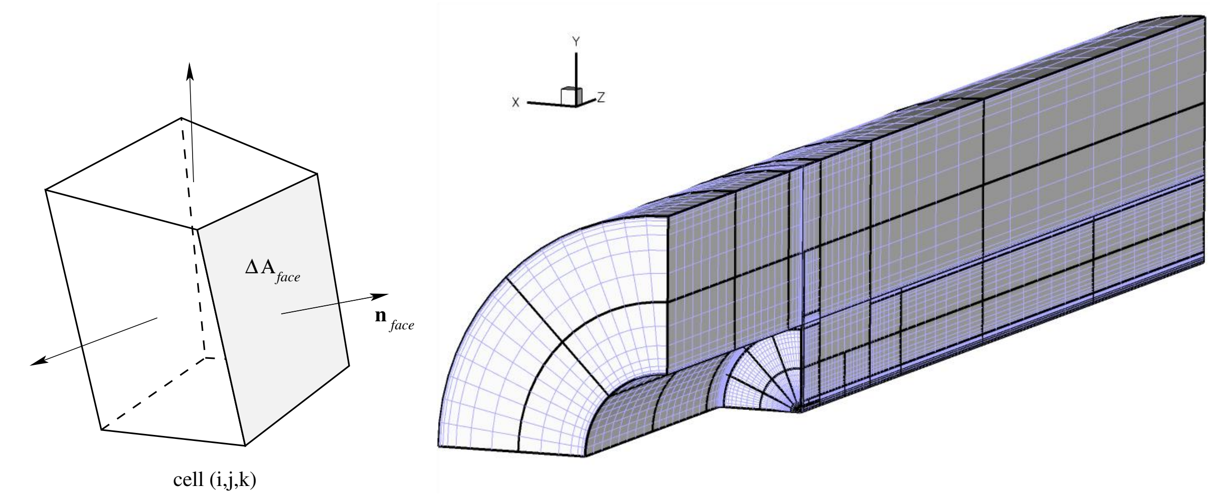
The data structure used to store the grid-block connectivity is a hierarchical flexible binary tree, which is accessible to all processors participating in the simulation. This tree not only takes into account the connectivity between individual blocks but also the splitting sequence of the block, which is not unique for anisotropic refinement. Figure 2 shows the resulting binary tree after several refinements of an initial mesh consisting of a single block. The binary tree for this anisotropic block-based AMR approach is relatively light and compact in terms of memory, as it only has to account for the connectivity between blocks and not the individual cells. As a result of this compact data structure, the block-based AMR approach is easier to implement within a parallel solution framework.

Each computational block contains two layers of ghost cells around it which carry information from the neighboring blocks. This information is exchanged between neighbouring blocks belonging to different processors through the MPI. In this implementation, which makes use of the heterogeneous block, the ghost cells for a block have the same refinement level as the neighbouring blocks overlapping the ghost cells. This approach is illustrated in Figure 3 for clarity. The domain in Figure 3 consists of 11 blocks adjacent to each other as shown in Figure 3(a). The ghost cells for the block at the centre (ID) are shown in Figure 3(b). The ghost cells and their solution values are directly provided by the neighboring blocks. This eliminates the need for prolongation of cell-averaged values from coarser to finer cells and restriction from finer to coarser cells, when exchanging information between blocks. When there are non-confirming cells or so-called “hanging nodes” where neighbouring blocks meet, the fluxes through the non-conforming faces are calculated in a systematic way, the same for neighbouring blocks, so as to maintain the conservation properties of the finite volume scheme.[21,35] In this way, the need for flux correction strategies to ensure the conservation properties of the scheme at block interfaces with resolution changes are also completely eliminated. This heterogeneous block approach is also beneficial in the construction of the matrix for the adjoint problem. For all of these reasons, the heterogeneous block approach with non-uniform refinement levels is used and exploited here.

III. Error Estimation Procedure and Output-Based Anisotropic AMR

III.A. Adjoint-Based Error Estimation

Evaluation of the solution adjoint combined with local solution error estimates enables calculation of an error estimate for the engineering functionals as well as the determination of sensitivities of the output to the solution error. These quantities can then be used to direct the mesh adaption. Adjoint methods have been extensively used for aerodynamic shape optimization[50–54] and mesh refinement.[36–43] They have also been applied to numerical simulation of reactive flows.[55,56] The adjoint problem can be broadly classified into the continuous and discrete adjoint formulations. The continuous adjoint equation is directly obtained from the differential forms of the fluid dynamical conservation equations, and then discretized and solved using a



(a)

(b)

Figure 1. (a) Hexahedral cell at grid location i, j, k showing face normals. (b) Body-fitted adapted mesh after several refinements. Grid blocks are shown with bold lines.

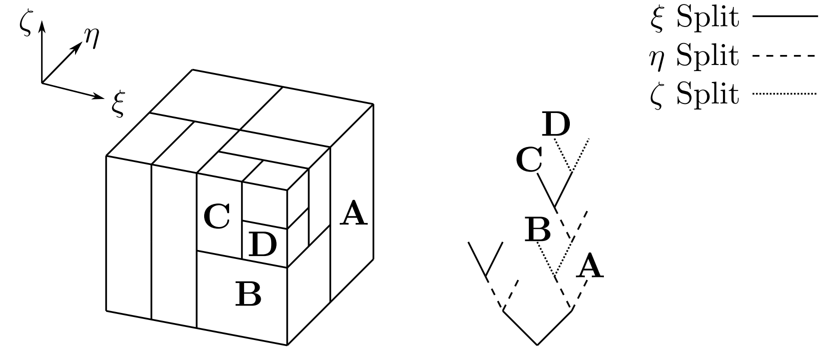


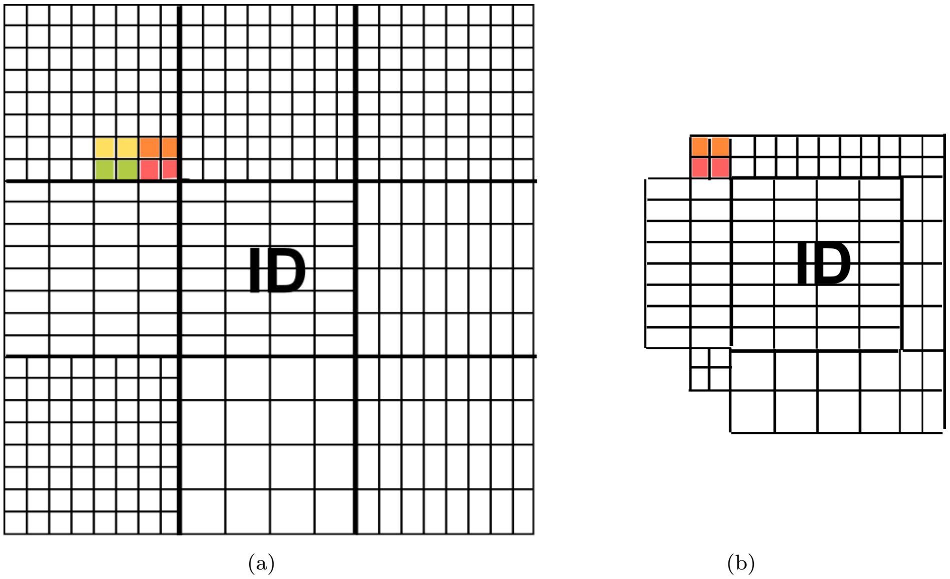
Figure 2. 3D binary tree and the corresponding blocks after several refinements

numerical technique. In this paper, the discrete adjoint[50–54] formulation is considered as it is believed to be more consistent with the cost function as pointed out by Nadarajah.[51] In the discrete adjoint formulation, the adjoint equations are derived directly from the finite volume discretization of the conservation equations through differentiation with respect to conserved variables.

Returning to Eq. (6), the semi-discrete form of the spatially discretized differential equations of interest is given by

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = -\mathbf{R}(\mathbf{U}), \tag{10}$$

where $\mathbf{R}(\mathbf{U})$ represents the residual. For a steady state, the solution has a residual $\mathbf{R}(\mathbf{U}) = 0$. The discretized governing equations are solved on the coarse grid denoted by Ω_H where H is the grid spacing on the coarse grid. The discrete residual on the coarse grid is given by $\mathbf{R}_H(\mathbf{U}_H) = 0$ where $\mathbf{U}_H$ is the discrete flow solution on the coarse grid. The primal solution vector, $\mathbf{U}$ is used to estimate the functional of interest $f(\mathbf{U})$. The approximation of the functional on the coarse grid using the coarse grid solution, is represented as $f_H(\mathbf{U}_H)$. The coarse grid solution is readily available and the functional is cheap to compute, but may not be accurate enough for the desired purpose. Hence, a solution of the problem on a finer grid, Ω_h, with a grid spacing h, is considered. The cost of computing the fine grid solution $\mathbf{U}_h$ is naturally much higher than that of the coarse grid solution $\mathbf{U}_H$. Calculation of the functional $f_h(\mathbf{U}_h)$ by first evaluating the actual fine mesh solution $\mathbf{U}_h$ would make the AMR approach pointless since the expensive computations would have been already performed.



(a) (b)

Figure 3. (a) Block under consideration (marked ID) surrounding by neighbouring blocks. (b) Ghost cells for the ID block taken directly from the neighbouring blocks.

The output-based refinement strategy considered here must instead provide an estimate for the fine grid solution in order to be able to produce a more accurate estimated value of the functional with respect to the observed integrated quantity on the original mesh. For the method to be effective, the estimation of the fine grid solution must be much less expensive than simply performing a uniform mesh refinement and solving the problem on a fine grid. The coarse grid solution $\mathbf{U}_H$ is prolonged to the fine grid as

$$\mathbf{U}_h^H = \mathbf{I}_h^H \mathbf{U}_H , \tag{11}$$

where $\mathbf{I}_h^H$ is a projection operator representing a limited piecewise linear least-squares reconstruction and $\mathbf{U}_h^H$ is the fine grid solution which has been prolonged from the coarse grid. The functional on the fine grid $f_h(\mathbf{U}_h)$ can be approximated using Taylor expansion on the prolonged solution $\mathbf{U}_h^H$. Neglecting higher order terms, one obtains

$$f_h(\mathbf{U}_h) = f_h(\mathbf{U}_h^H) + \frac{\partial f_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}(\mathbf{U}_h - \mathbf{U}_h^H) + \mathcal{O}((\mathbf{U}_h - \mathbf{U}_h^H)^2)\,. \tag{12}$$

The residual on the fine grid is $\mathbf{R}_h(\mathbf{U}_h) = 0$. This residual can also be expanded using Taylor expansion on the prolonged solution $\mathbf{U}_h^H$ as

$$\mathbf{R}_h(\mathbf{U}_h) = \mathbf{R}_h(\mathbf{U}_h^H) + \frac{\partial \mathbf{R}_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}(\mathbf{U}_h - \mathbf{U}_h^H) + \mathcal{O}((\mathbf{U}_h - \mathbf{U}_h^H)^2)\,. \tag{13}$$

Note that $\mathbf{R}_h(\mathbf{U}_h^H) \neq 0$ on the fine grid because $\mathbf{U}_h^H$ is the coarse grid prolonged solution and not the actual fine grid solution $\mathbf{U}_h$ obtained by solving the fluid equations on the fine grid. Using $\mathbf{R}_h(\mathbf{U}_h) = 0$, this becomes

$$(\mathbf{U}_h - \mathbf{U}_h^H) \approx \left[\frac{\partial \mathbf{R}_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}\right]^{-1} \mathbf{R}_h(\mathbf{U}_h^H)\,. \tag{14}$$

Neglecting higher order terms and using Eq. (14), Eq. (12) can be then written as

$$f_h(\mathbf{U}_h) \approx f_h(\mathbf{U}_h^H) + \Big(\boldsymbol{\psi}_h(\mathbf{U}_h^H)\Big)^T \mathbf{R}_h(\mathbf{U}_h^H), \tag{15}$$

where $\boldsymbol{\psi}_h(\mathbf{U}_h^H)$ is defined as the discrete adjoint solution on the fine grid. As can be seen from Eq. (15), this term can also be viewed as the sensitivity of the functional towards the residual error on the fine grid. The second term $\left(\boldsymbol{\psi}_h(\mathbf{U}_h^H)\right)^T \mathbf{R}_h(\mathbf{U}_h^H)$ in Eq. (15) is the error estimate required to drive the mesh adaptation. The fine grid discrete adjoint, $\boldsymbol{\psi}_h(\mathbf{U}_h^H)$, satisfies the discrete adjoint equation

$$\left(\frac{\partial \mathbf{R}_h}{\partial \mathbf{U}_h}\right)^T \boldsymbol{\psi}_h = \left(\frac{\partial f_h}{\partial \mathbf{U}_h}\right)^T . \tag{16}$$

on the fine grid.

Equation (16) is not solved due to its high computational expense. The fine grid discrete adjoint, $\boldsymbol{\psi}_h(\mathbf{U}_h^H)$, is instead approximated by an interpolated value $\boldsymbol{\psi}_h^H$ given by

$$\boldsymbol{\psi}_h^H = \mathbf{J}_h^H \boldsymbol{\psi}_H, \tag{17}$$

where $\boldsymbol{\psi}_H$ is the discrete adjoint on the coarse grid and the solution of the coarse grid discrete adjoint equation given by

$$\left(\frac{\partial \mathbf{R}_H}{\partial \mathbf{U}_H}\right)^T \boldsymbol{\psi}_H = \left(\frac{\partial f_H}{\partial \mathbf{U}_H}\right)^T . \tag{18}$$

Here, $\mathbf{J}_h^H$ is also a projection operator like $\mathbf{I}_h^H$. Note that a finer mesh is used here to evaluate the primal residual error. An alternate approach would be to estimate the residual error on the coarse grid itself by using a higher-order spatial discretization or reconstruction procedure for the primal flow solution quantities.[36, 43, 44] Such an approach is not considered here.

Equation (18) is a linear system of the form $\mathbf{Ax} = \mathbf{b}$. This system is a large sparse linear system and is solved using GMRES. The Trilinos software package is used in combination with MPI for this purpose. Trilinos[57–61] contains subpackages like Epetra[62] for the various matrix and vector classes for data distribution over multiple processors and AztecOO[63] for routines like GMRES and preconditioning for faster convergence of linear systems.

III.B. Criteria for Output-Based Mesh Adaptation

One way to drive the grid refinement is to directly use the local correction $\left(\boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\mathbf{U}_h^H)$ in the functional calculated as described in Section III.A, as the refinement criterion. The interpolated adjoint described in Equation (17) is used for this purpose. As described by Venditti and Darmofal[38–40] , this approach can also lead to unnecessary refinement in regions where the adjoint solution is not sufficiently resolved. Hence, Venditti and Darmofal proposed a more conservative criterion for adaptation that takes into account not only primal but also dual (adjoint) residual errors. In the current work, both choices of adaptation parameters have been explored. Consider now the more conservative criterion.

Equation (15) can be written as

$$f_h(\mathbf{U}_h) - f_h(\mathbf{U}_h^H) \approx \underbrace{\left(\boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\mathbf{U}_h^H)}_{\text{Computable Correction}} + \underbrace{\left(\boldsymbol{\psi}_h(\mathbf{U}_h^H) - \boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\mathbf{U}_h^H)}_{\text{Error in Computable Correction}}, \tag{19}$$

where $\boldsymbol{\psi}_h^H$ is the interpolated coarse-grid adjoint shown in Equation (17) whereas $\boldsymbol{\psi}_h(\mathbf{U}_h^H)$ is the more accurate value of the adjoint calculated on the fine grid. Values for $\boldsymbol{\psi}_h(\mathbf{U}_h^H)$ are not calculated due to the relatively high computational cost. Hence, $\left(\boldsymbol{\psi}_h(\mathbf{U}_h^H) - \boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\mathbf{U}_h^H)$ becomes the error in the computational correction. Equation (19) can also be written as

$$f_h(\mathbf{U}_h) - f_h(\mathbf{U}_h^H) \approx \underbrace{\left(\boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\mathbf{U}_h^H)}_{\text{Computable Correction}} + \underbrace{\mathbf{R}_h^{\boldsymbol{\psi}}(\boldsymbol{\psi}_h^H)^T \left[\frac{\partial \mathbf{R}_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}\right]^{-1} \mathbf{R}_h(\mathbf{U}_h^H)}_{\text{Error in Computable Correction}}, \tag{20}$$

where $\mathbf{R}_h^{\boldsymbol{\psi}}$ is the adjoint residual operator defined as

$$\mathbf{R}_h^{\boldsymbol{\psi}}(\boldsymbol{\psi}) = \left[\frac{\partial \mathbf{R}_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}\right]^T \boldsymbol{\psi} - \left(\frac{\partial f_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}\right)^T . \tag{21}$$

Since $\boldsymbol{\psi}_h(\mathbf{U}_h^H)$ is the value of the adjoint calculated by solving the adjoint problem on the fine grid, it satisfies the equation

$$\left[\frac{\partial \mathbf{R}_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}\right]^T \boldsymbol{\psi}_h(\mathbf{U}_h^H) = \left(\frac{\partial f_h(\mathbf{U}_h^H)}{\partial \mathbf{U}_h}\right)^T . \tag{22}$$

Hence, its residual $\mathbf{R}_h^{\boldsymbol{\psi}}(\boldsymbol{\psi}_h(\mathbf{U}_h^H))$ is equal to zero. But, the fine grid residual $\mathbf{R}_h^{\boldsymbol{\psi}}(\boldsymbol{\psi}_h^H)$ of the interpolated coarse-grid adjoint $\boldsymbol{\psi}_h^H$ would be non-zero. Calculation of this residual requires the construction of the matrix $[\partial \mathbf{R}_h(\mathbf{U}_h^H)/\partial \mathbf{U}_h]$ and vector $(\partial f_h(\mathbf{U}_h^H)/\partial \mathbf{U}_h)$ on the fine grid; solution of the linear system on the fine grid is not required.

Hence, as seen in Equation (20), the error in the computable correction can be written in terms of the primal as well as adjoint residuals on the fine mesh level. Consequently, reducing the primal and dual residuals simultaneously can reduce the error in the computable correction. The new parameter focuses on attempting to simultaneously reducing the local residual errors in both the primal and adjoint solutions. The new parameter tries to combine the two residuals into a single adaptation parameter for each cell, also taking into account the fact that the primal and dual variables have different units and magnitudes and hence need to be assigned weights appropriately. The criterion defined by Venditti and Darmofal[38–40] for a particular coarse grid cell i is given by

$$\begin{aligned}\boldsymbol{\varepsilon}_i =& \frac{1}{2}\sum_{l(i)}\Big(\Big|[Q_h^H\boldsymbol{\psi}_H - L_h^H\boldsymbol{\psi}_H]_{l(i)}^T[\mathbf{R}_h(L_h^H\mathbf{U}_H)]_{l(i)}\Big| \\ &+ \Big|[Q_h^H\mathbf{U}_H - L_h^H\mathbf{U}_H]_{l(k)}^T[\mathbf{R}_h^{\boldsymbol{\psi}}(L_h^H\boldsymbol{\psi}_H)]_{l(i)}\Big|\Big).\end{aligned} \tag{23}$$

Here, Q_h^H and L_h^H are quadratic and linear interpolation operators respectively, used for prolonging coarse grid variables into the fine grid using reconstruction. The summation is performed over all fine grid cells $l(i)$ contained within the coarse grid cell i.

IV. Numerical Results for Invisicid and Viscous Flows

In this section, application of the proposed error-based anisotropic AMR scheme to several inviscid and viscous flow problems is considered and the performance of the solution-dependent AMR strategy is assessed through comparisons to results obtained using both uniform mesh refinement and other more standard gradient-based AMR approaches.

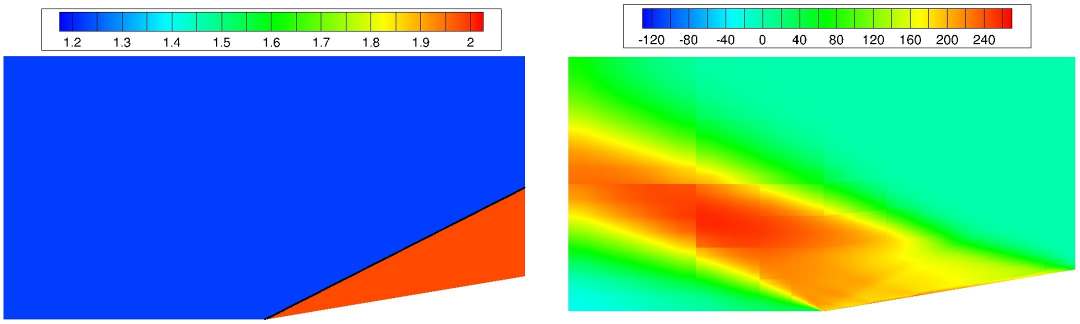
IV.A. Supersonic Flow Past a Wedge

In the first example case, prediction of the oblique shock that forms when inviscid supersonic flow of air is deflected by an inclined wedge is examined. A Mach number M=3 supersonic flow is considered, and the wedge angle α is 9.5°. Using the Rankine-Hugoniot relations, the angle of the shock and the analytical solution can be determined. The angle of the shock is given by $\beta = 26.9308°$. The ideal analytical solution on the upstream and downstream sides of the shock in terms of ρ, p and M is given by

$$\begin{bmatrix}\rho\\ p\\ M\end{bmatrix}_{left} = \begin{bmatrix}1.225\\ 101325\\ 3\end{bmatrix}, \begin{bmatrix}\rho\\ p\\ M\end{bmatrix}_{right} = \begin{bmatrix}1.98\\ 201354.51\\ 2.53\end{bmatrix} . \tag{24}$$

The functional used here for this oblique shock case is the pressure drag force on the inclined wedge in the direction of the incoming flow. For a wedge of length $l = 1$ and width $w = 0.25$, the ideal value of the drag force is given by $D_a = p_{right} \cdot w \cdot l \cdot \sin(\alpha) = 8308.27$.

Figure 4(a) shows the predicted density distribution for this case after a converged solution has been obtained after 7 gradient-based refinements on the initial mesh. Figure 4(b) shows the distribution for the 1^{st} component of the adjoint solution vector after a converged solution has been obtained after 7 output-based refinements on the initial mesh. Figure 6 shows the computational mesh for different levels of refinement for the two mesh refinement methods. Comparisons are made between gradient-based and functional error-based methods for driving the mesh refinement. The former refines the entire oblique shock whereas the latter refines only the portion of the shock that falls in the domain of influence of the functional where the adjoint variable is non-zero. In terms of mesh savings, this represents 37.4% reduction is mesh size. Using the criterion defined in Equation (23) leads to further reductions in mesh size. Figure 5 shows a plot of the error in the functional, i.e., the difference between the ideal value D_a and the computed value D_c against $\sqrt[3]{N_{cells}}$ where N_{cells} is the total number of grid cells in the computational domain. While it is evident that the gradient-based approach outperforms the convergence rate obtained with uniform refinement, the output-based methods provide further improvements in the convergence for this case.



(a)

(b)

Figure 4. Predicted density (a) and density adjoint (b) distributions for a Mach number M=3 steady supersonic inviscid flow past a wedge on refined anisotropic meshes. For (a), the total number of blocks is 890, the number of cells per block is $8 \times 8 \times 4 = 256$, and the total number of cells is 227840. For (b), the total number of blocks is 557, the number of cells per block is $8 \times 8 \times 4 = 256$, and the total number of cells is 142592.

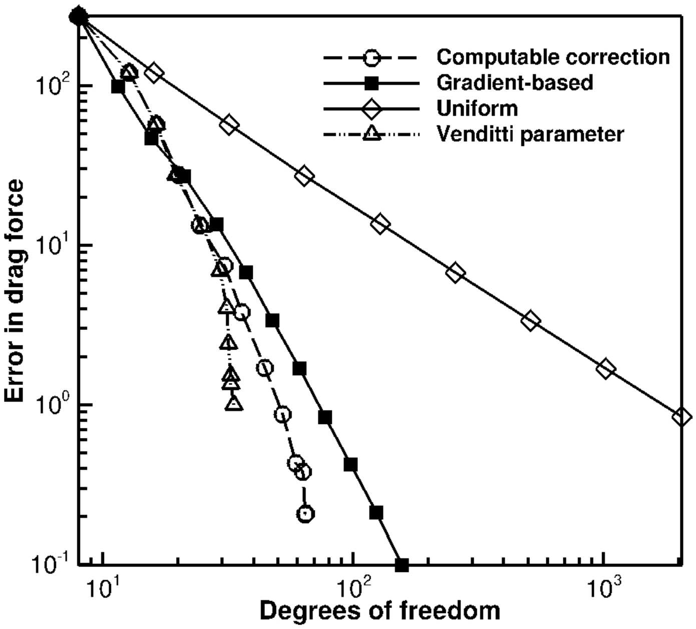
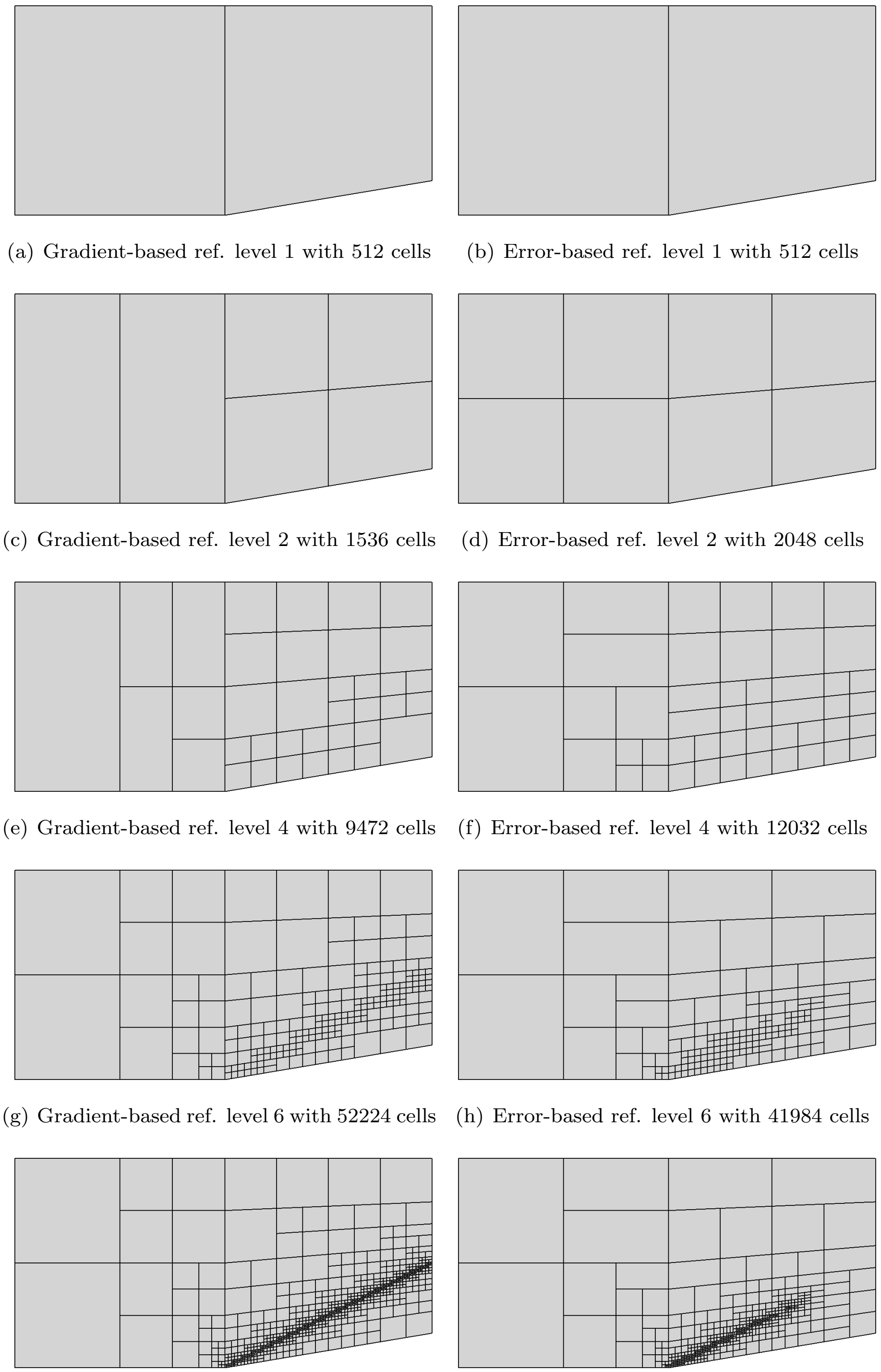


Figure 5. Convergence of the predicted error in the drag force as a function of the mesh size (numbers of degrees of freedom) for supersonic flow past an inclined wedge.

IV.B. Supersonic Flow Over a Bump

In the second inviscid flow problem, a Mach number M=1.4 supersonic flow enters a rectangular channel and is intercepted by a sinusoidal bump. The functional used here is the pressure-induced drag force on the



(a) Gradient-based ref. level 1 with 512 cells

(b) Error-based ref. level 1 with 512 cells

(c) Gradient-based ref. level 2 with 1536 cells

(d) Error-based ref. level 2 with 2048 cells

(e) Gradient-based ref. level 4 with 9472 cells

(f) Error-based ref. level 4 with 12032 cells

(g) Gradient-based ref. level 6 with 52224 cells

(h) Error-based ref. level 6 with 41984 cells

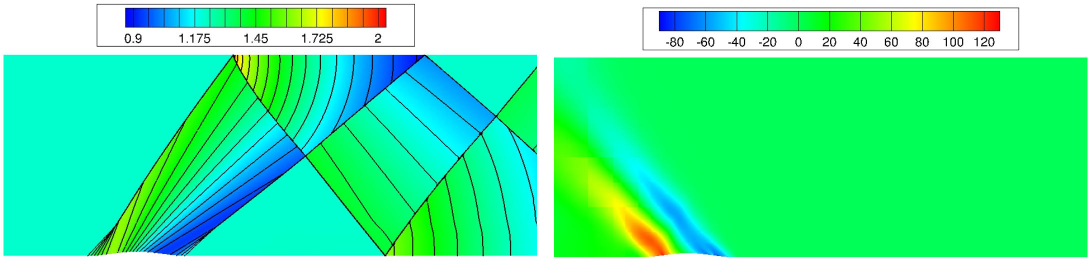
(i) Gradient-based ref. level 8 with 227840 cells

(j) Error-based ref. level 8 with 142592 cells

Figure 6. Stages of AMR for error-based refinement using computable correction (right) compared to those obtained using gradient-based approach (left) for supersonic flow past a wedge.

bump in the direction of the incoming flow. The presence of the bump results in the formation of strong oblique shocks and subsequent shock reflections from the upper and lower boundaries.

The gradient-based refinement method for this case refines heavily in the regions containing the shocks



(a) (b)

Figure 7. Predicted density (a) and density adjoint (b) distributions for a Mach number M=1.4 steady supersonic inviscid flow over a bump obtained on refined anisotropic meshes. For (a), the total number of blocks is 6940, the number of cells per block is 8 × 8 × 4 = 256,and the total number of cells is 1776640. For (b), the total number of blocks is 265, the number of cells per block is 8 × 8 × 4 = 256, and the total number of cells is 67840.

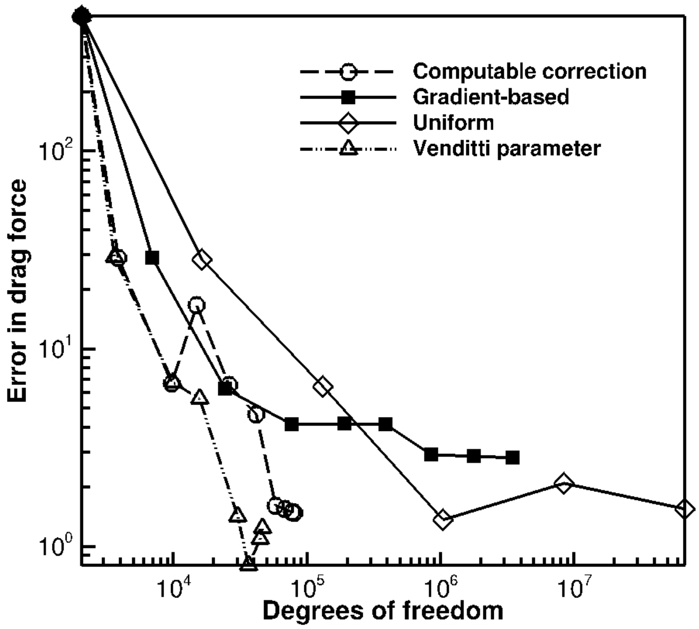
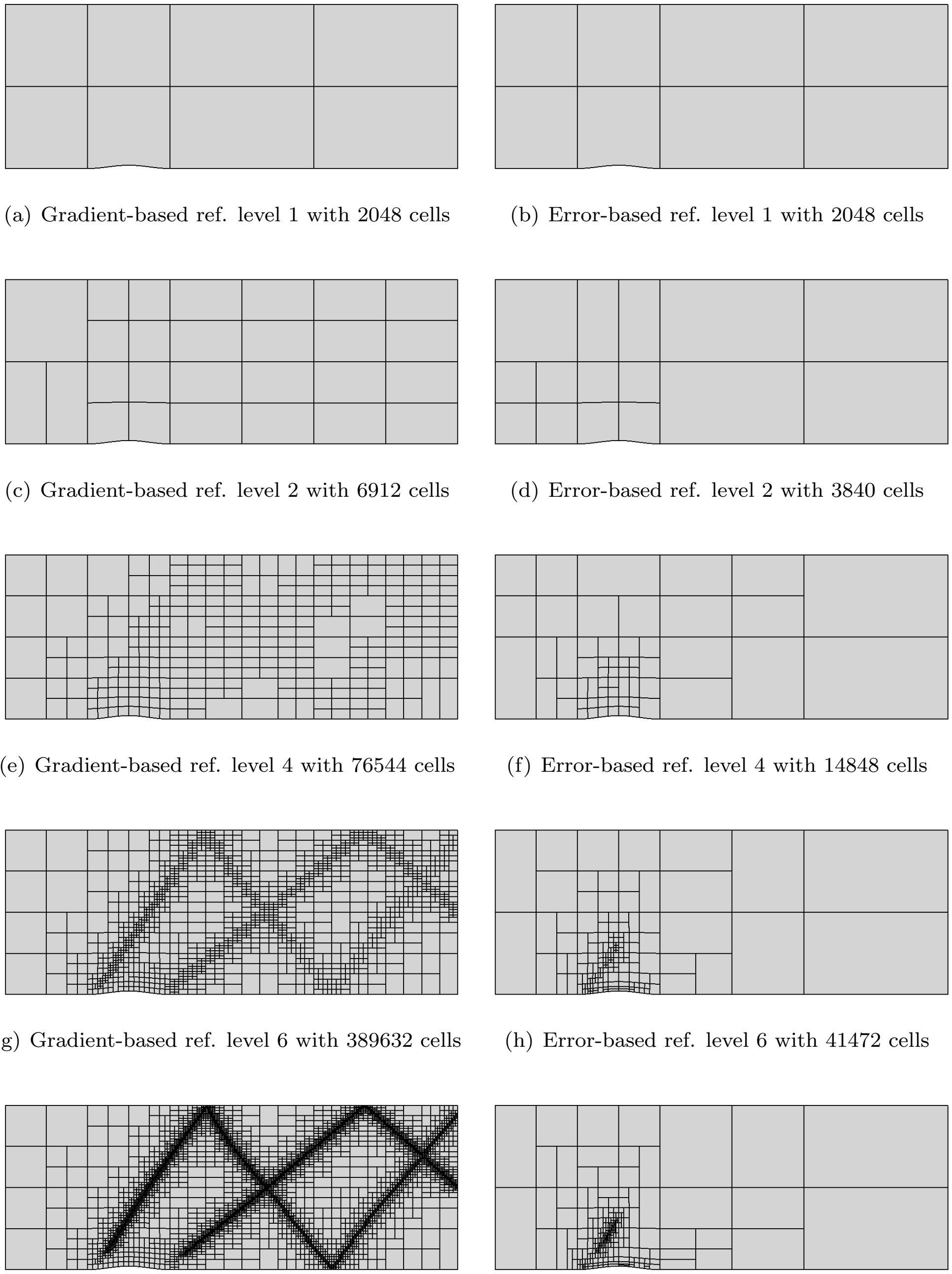


Figure 8. Convergence of the predicted error in the drag force as a function of the mesh size (numbers of degrees of freedom) for supersonic flow over a bump.

and their reflections, increasing the computational cost drastically without a commensurate increase in the accuracy of the computed value of the functional as can be seen in Figure 9-left. The error-based refinement method, on the other hand, focuses on refining the portion of the shock which influences the flow over the bump as outlined in Figure 9-right. In terms of mesh savings, this represents a 96% reduction in mesh size. A further 30% reduction in mesh size is achieved using the more conservative refinement criterion defined in Equation (23). The regions downstream of the bump have no influence over the functional since no information travels upstream in a supersonic flow. Figure 7(a) shows the predicted density distribution for this case after a converged solution has been obtained after 7 gradient-based refinements on the initial mesh. Figure 7(b) shows the distribution for the 1^{st} component of the adjoint solution vector after a converged solution has been obtained after 8 output-based refinements on the initial mesh. Figure 8 shows a plot of the error in the functional, i.e., the difference between the ideal value D_a and the computed value D_c against $\sqrt[3]{N_{cells}}$ where N_{cells} is the total number of grid cells in the computational domain. The ideal value D_a is approximated by obtaining a solution on a uniformly refined mesh at the finest level of refinement. The gradient-based refinement strategy initially slightly outperforms the uniform refinement strategy but after 4 refinements, it reaches a plateau and the uniform refinement strategy produces more accurate results. However, the output-based refinement approaches continue to produce more accurate predictions of the functional as the mesh is further refined.



(a) Gradient-based ref. level 1 with 2048 cells

(b) Error-based ref. level 1 with 2048 cells

(c) Gradient-based ref. level 2 with 6912 cells

(d) Error-based ref. level 2 with 3840 cells

(e) Gradient-based ref. level 4 with 76544 cells

(f) Error-based ref. level 4 with 14848 cells

(g) Gradient-based ref. level 6 with 389632 cells

(h) Error-based ref. level 6 with 41472 cells

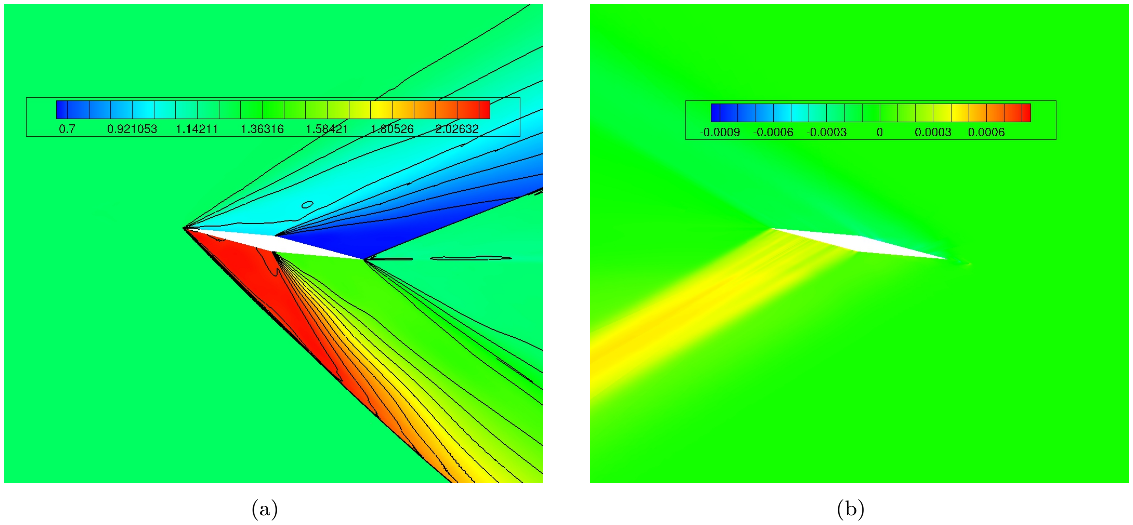
(i) Gradient-based ref. level 8 with 1776640 cells

(j) Error-based ref. level 8 with 67840 cells

Figure 9. Stages of AMR for error-based refinement using computable correction (right) compared to those obtained using gradient-based approach (left) for supersonic flow past a bump.

IV.C. Supersonic Flow Past a Diamond-Shaped Airfoil

In the next case of interest, inviscid supersonic flow is intercepted by a symmetrical diamond-shaped airfoil. A Mach number M=2 supersonic flow is considered. The chord length of the airfoil is assumed to be unity



(a)

(b)

Figure 10. Predicted density (a) and density adjoint (b) distributions for a Mach number M=2 steady supersonic inviscid flow past a diamond-shaped airfoil on refined anisotropic meshes. For (a), the total number of blocks is 9551, the number of cells per block is $8 \times 8 \times 4 = 256$, and the total number of cells is 7640800. For (b), the total number of blocks is 2314, the number of cells per block is $8 \times 8 \times 4 = 256$, and the total number of cells is 1851200.

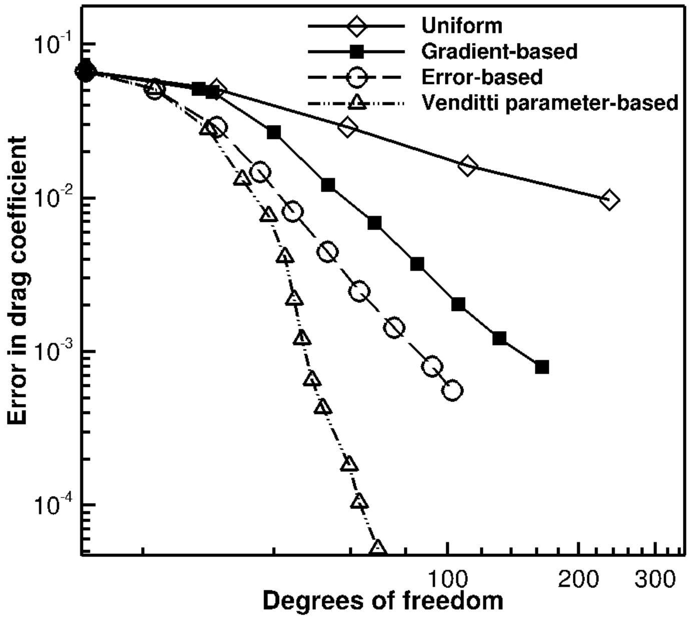


Figure 11. Convergence of the predicted error in the drag force as a function of the mesh size (numbers of degrees of freedom) for supersonic flow past a diamond-shaped airfoil.

and the far-field boundary is located 32 chord lengths away from the airfoil in a radially outward direction. The angle of attack of the airfoil is given by $\alpha = 10°$. Each of the flat sections of the airfoil makes an angle $\gamma = 5°$ with the chord line of the airfoil. Oblique shocks are formed at the bottom leading edge and the upper trailing edge of the airfoil. Prandtl-Meyer expansion fans are also generated at the upper leading edge and bottom trailing edge of the airfoil. In addition, expansion fans are generated at the intersection of the flat sections of the airfoil where the airfoil thickness is the greatest. The functional used here is the pressure drag force on the airfoil in the direction of the free-stream flow. The flow solution adjacent to the airfoil can be calculated using the Rankine-Hugoniot relations across the oblique shocks and isentropic flow relations

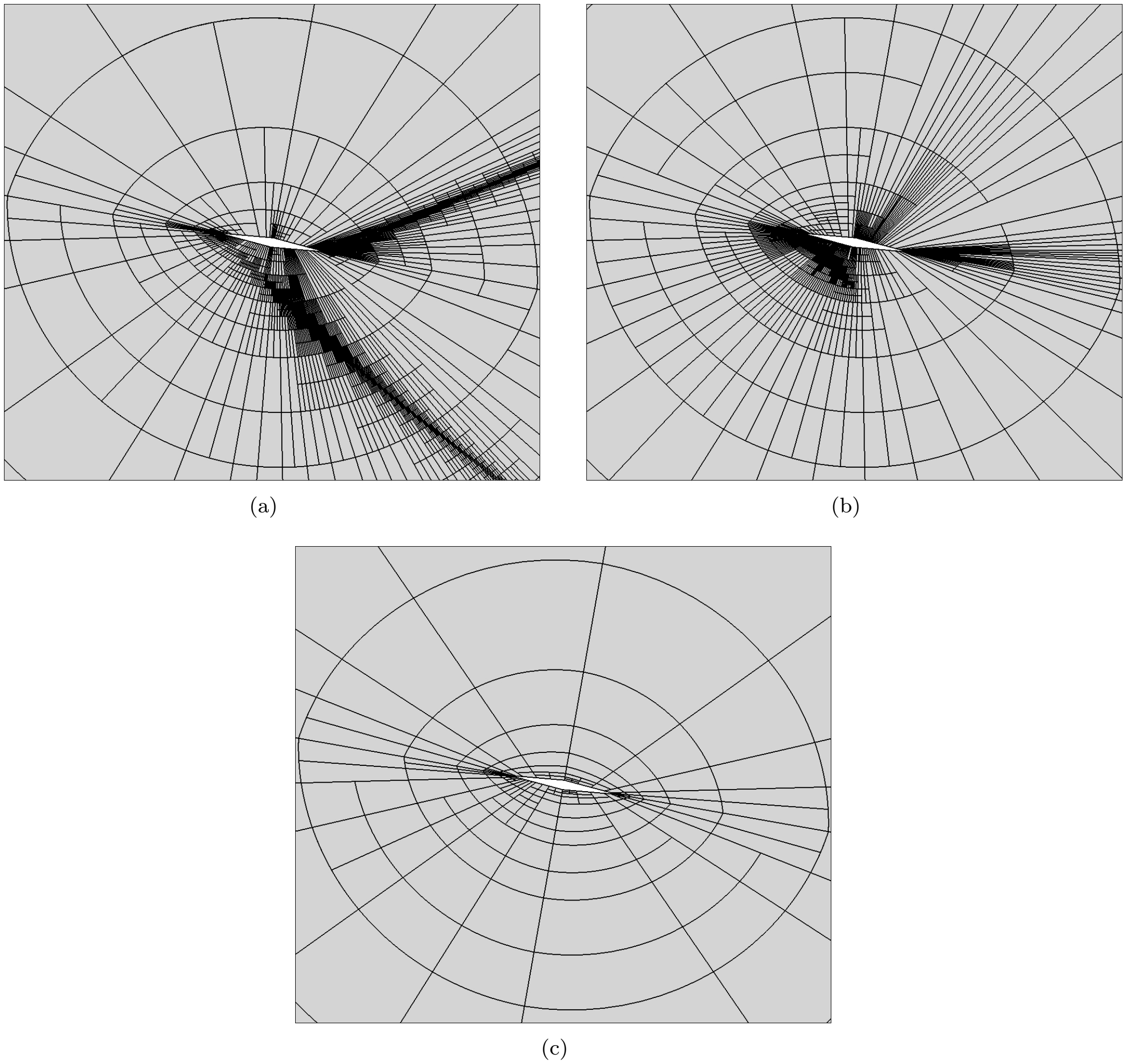


Figure 12. Computational meshes for Mach number M=2 steady supersonic inviscid flow past a diamond-shaped airfoil showing: (a) a refined anisotropic mesh after 10 refinements obtained using a gradient-based refinement strategy (total number of blocks is 9551, number of cells per block is $8 \times 8 \times 4 = 256$, and total number of cells is 7640800); (b) a refined anisotropic mesh after 10 refinements obtained using an error-based refinement strategy with computable correction (total number of blocks is 2314, number of cells per block is $8 \times 8 \times 4 = 256$, and total number of cells is 1851200); and (c) a refined anisotropic mesh after 10 refinements obtained using an error-based refinement strategy with Venditti-Darmofal criterion (total number of blocks is 822, number of cells per block is $8 \times 8 \times 4 = 256$, and total number of cells is 210432).

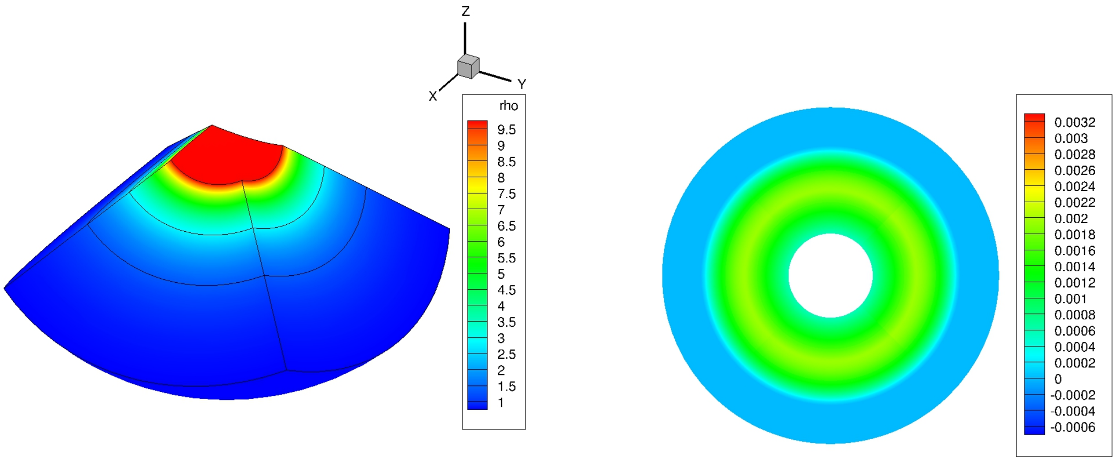
across the Prandtl-Meyer expansion fans. Thus, an analytical ideal value of the drag force can be calculated. The drag coefficient is given by

$$C_D = \frac{2D}{\rho_\infty V_\infty^2}, \tag{25}$$

where D is the drag force, ρ_∞ is the free-stream density and V_∞ is the free-stream velocity. The analytical value of the drag coefficient is found out to be 0.0926.

Figure 10(a) shows the predicted density distribution for this case after a converged solution has been obtained after 9 gradient-based refinements on the initial mesh. Figure 10(b) shows the distribution of the 1^{st} component of the adjoint solution vector after a converged solution has been obtained after 9 output-based refinements on the initial mesh. The gradient-based technique refines extensively in the regions of the shocks and also significantly in the regions containing the sharp expansion fans. On the other hand, the error-based method refines only a small portion of the leading shock and expansion fan and the central expansion fans. The trailing edge shock and expansion fan are not refined significantly, leading to huge savings in mesh. Using the computable correction as the refinement criterion achieves an 82% reduction in mesh size as compared to the gradient-based approach. The more conservative refinement criterion achieves a 97% reduction in mesh size for the same accuracy. Figure 11 shows a plot of the error in the functional, i.e., the difference between the analytical value $C_{D,a}$ and the computed value $C_{D,c}$ against $\sqrt[3]{N_{cells}}$ where N_{cells} is the total number of grid cells in the computational domain. The error-based refinement technique has a convergence rate comparable to that of the gradient-based technique but has an offset, while the more conservative error-based approach has a steeper convergence rate. As further evidence of the performance of the proposed output-based anisotropic AMR scheme, Figure 12(a) shows a refined anisotropic mesh after 10 refinements obtained using the gradient-based criteria. For comparison, Figure 12(b) and 12(c) show the adapted anisotropic meshes after 10 refinements obtained using the proposed output-based approach.

IV.D. Steady Supersonic Spherical Outflow



(a)

(b)

Figure 13. Predicted density (a) and cross-section of density adjoint (b) distributions for a steady supersonic spherical outflow problem on a portion of the spherical domain. For (a), the total number of blocks is 18, the number of cells per block is 6 × 6 × 24 = 864, and the total number of cells is 15552. For (b), the total number of blocks is 18, the number of cells per block is 6 × 6 × 24 = 864, and the total number of cells is 15552.

The last inviscid flow case considered is that of steady supersonic spherical outflow. A spherical computational domain is used for this case with a cubed-sphere multi-block mesh as described by Ivan *et al.*[29,30] The domain consists of six blocks connected, forming an inner hollow sphere and an outer spherical shell. We consider a spherical inflow with radius $R_i = 1$ m and a spherical outflow with radius $R_o = 4$ m. Air at a supersonic speed enters through the inner sphere with a velocity vector pointing in the radially outward direction. The air subsequently expands and exits supersonically through the outflow sphere. The inflow is fixed with a flow density $\rho_i = 10$ kg/m^3, radial velocity $Vr, i = 4.5$ m/s, and pressure

$$C_3 - \frac{1}{r^2 V_r \left[(C_2 - V_r^2)^{\frac{1}{\gamma-1}} \right]} = 0, \tag{26}$$

where

$$C_3 = \frac{1}{\left(\frac{2\gamma}{\gamma-1} \frac{p_i}{\rho_i}^{\frac{1}{\gamma-1}} \right) R_i^2 V_{r,i}}, \quad C_2 = \frac{2\gamma}{\gamma-1} \frac{p_i}{\rho_i} + V_{r,i}^2, \tag{27}$$

are constants depending on the inflow conditions. The functional used here is the volume-integrated average temperature of the air contained in the portion of the spherical domain contained between the radii 2 m and 3 m. The ideal value of the functional can be calculated analytically and is 0.31560688.

Figure 13(a) shows the density distribution for a portion of the spherical domain for this case after a converged solution has been obtained after 2 gradient-based refinements on the initial mesh. Figure 13(b)

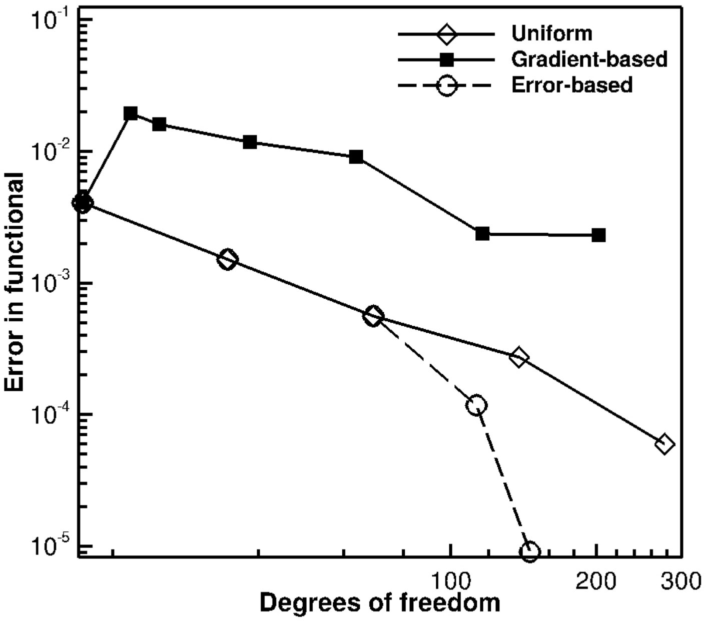
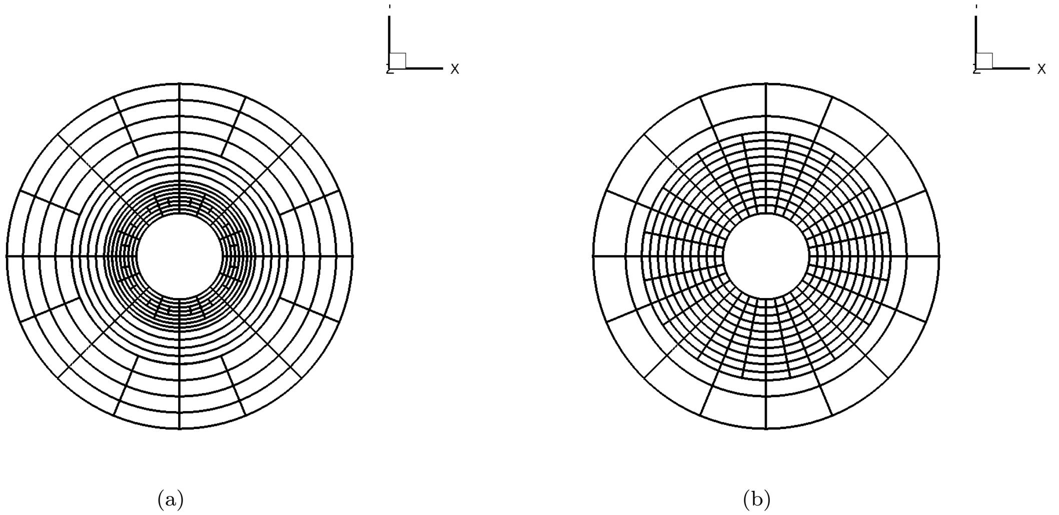


Figure 14. Convergence of the predicted error in the functional as a function of the mesh size (numbers of degrees of freedom) for steady supersonic spherical flow.

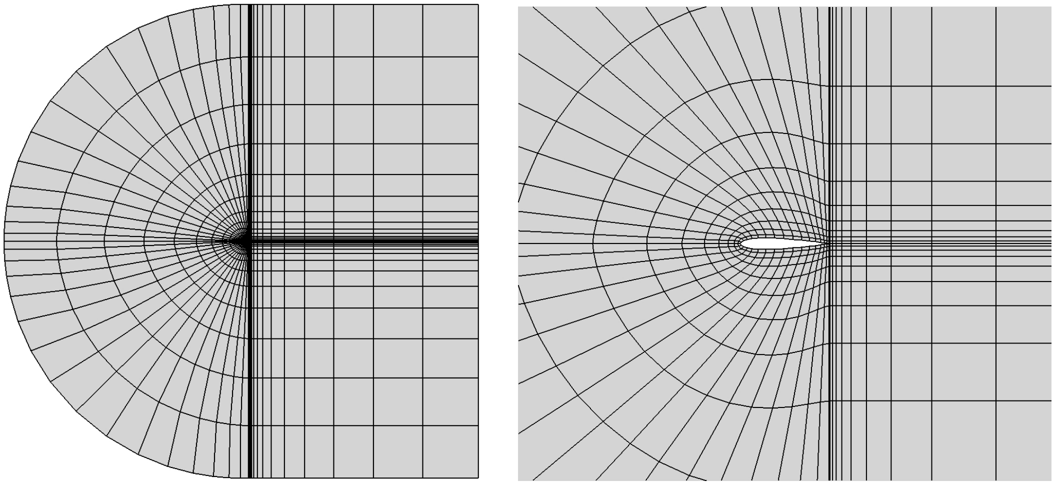


(a)

(b)

Figure 15. Computational meshes for steady supersonic spherical outflow problem showing: (a) a cross-sectional view of a refined anisotropic mesh after 6 refinements obtained using a gradient-based refinement strategy (total number of blocks is 1824, number of cells per block is $6 \times 6 \times 24 = 864$, and total number of cells is 1575936); and (b) cross-sectional view of a refined anisotropic mesh after 5 refinements obtained using an error-based refinement method using computable correction (total number of blocks is 3582, number of cells per block is $6 \times 6 \times 24 = 864$, and total number of cells is 3094848).

shows a cross-sectional view of distribution for the 1^{st} component of the adjoint solution vector after a converged solution has been obtained after 2 output-based refinements on the initial mesh. The error-based approach is able to detect errors in the azimuthal direction whereas the gradient-based approach is not. Figure 14 shows a plot of the error in the functional, i.e., the difference between the ideal and computed values against $\sqrt[3]{N_{cells}}$ where N_{cells} is the total number of grid cells in the computational domain. Figure 15(a) shows a cross-sectional view of the refined anisotropic mesh after 5 refinements using a gradient-based approach whereas Figure 15(b) shows a cross-sectional view of the refined anisotropic mesh after 5 refinements using an error-based approach.

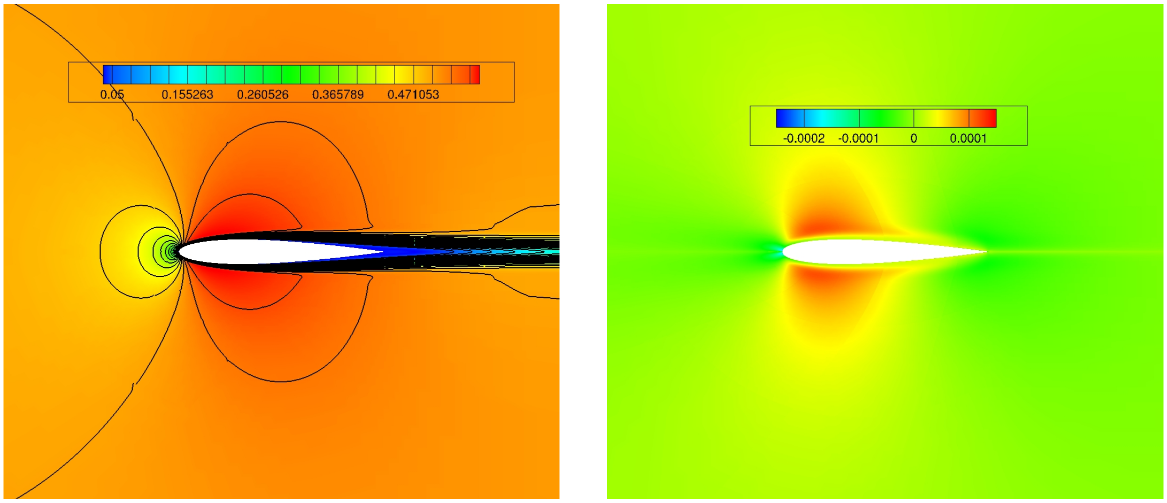


(a)

(b)

Figure 16. Initial computational mesh for Mach number M=0.5 steady subsonic viscous flow past a NACA0012 airfoil showing: (a) mesh for entire computational domian (total number of blocks is 4, number of cells per block is 16 × 16 × 4 = 1024, and total number of cells is 4096); and (b) close-up view of initial mesh.

IV.E. Subsonic Viscous Flow Past an Airfoil



(a)

(b)

Figure 17. Predicted Mach number (a) and energy adjoint (b) distributions for a Mach number M=0.5 steady subsonic viscous flow past a NACA0012 airfoil on refined anisotropic meshes. For (a), the total number of blocks is 160, the number of cells per block is 16 × 16 × 4 = 1024, and he total number of cells is 163840. For (b), the total number of blocks is 160, the number of cells per block is 16 × 16 × 4 = 1024, and the total number of cells is 163840.

In the first of two viscous flow cases considered here, a horizontal viscous Mach number M=0.5 subsonic flow at a Reynolds Number Re = 5000 is intercepted by a NACA0012 airfoil at an angle of attack given by $\alpha = 0°$. A C-shaped grid with stretching initially composed of 4 blocks as shown in Figure 16 was used for this case. The chord length is taken to be unity and the far-field boundary is located 32 chord lengths away from the airfoil boundary in the radially outward direction. A no-slip boundary condition is imposed on the airfoil boundary. The main feature of this case is the creation of a viscous boundary layer on the airfoil surface. The functional used here is the total drag force on the airfoil in the direction of the free-stream flow

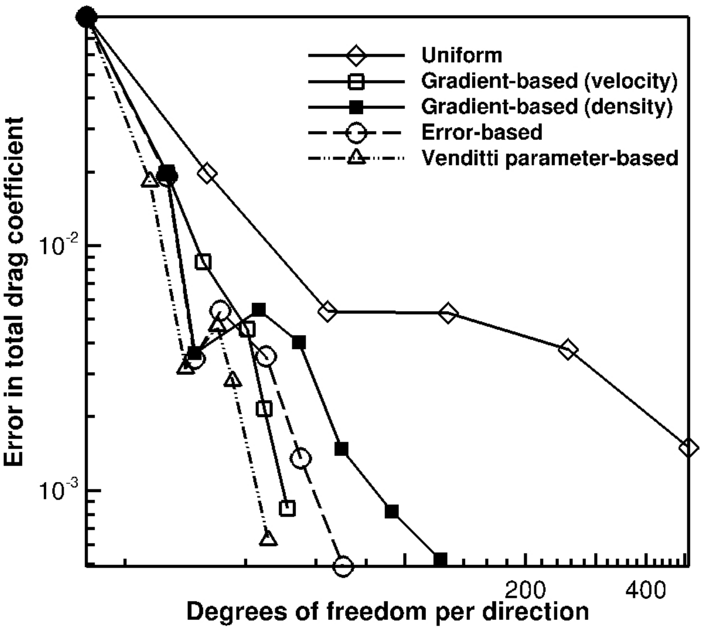


Figure 18. Convergence of the error in the predicted drag coefficient as a function of the mesh size (numbers of degrees of freedom) for subsonic viscous flow past a NACA0012 airfoil.

which comprises the pressure and viscous drag components. The ideal drag coefficient is taken to be 0.0555 based on simulations performed by Ivan.[64] This case has also been examined in several previous studies.[65–68]

Figure 17(a) shows Mach number distribution for this case after a converged solution has been obtained after 6 output-based refinements on the initial mesh. Figure 17(b) shows the distribution of the 1^{st} component of the adjoint solution vector on the same mesh. The density gradient-based method refines excessively in the region near the leading edge of the airfoil as compared to the error-based method. The velocity gradient-based method focuses on refining the region containing the boundary layer. Figure 18 shows a plot of the error in the functional, i.e., the difference between the ideal value $C_{D,a}$ and the computed value $C_{D,c}$ of the drag coefficient against $\sqrt[3]{N_{cells}}$ where N_{cells} is the total number of grid cells in the computational domain. The performance of the output-based AMR strategy employing the computable correction as the refinement criterion is better than that of the density gradient-based refinement technique, but worse than that of the velocity gradient-based refinement technique. The conservative error-based approach employing the criterion defined in Equation (23) has the best performance with maximum mesh savings. Figure 19 depicts a comparison of the AMR meshes obtained after six successive refinements using the various refinement criteria/strategies.

IV.F. Supersonic Viscous Flow Past an Airfoil

As a last case, horizontal viscous Mach number M = 1.2 supersonic flow at a Reynolds Number Re = 1000 is intercepted by a NACA0012 airfoil at an angle of attack given by $\alpha = 0°$. The C-shaped grid shown in Figure 16 was used for this case. The chord length is taken to be unity and the far-field boundary is located 32 chord lengths away from the airfoil boundary in the radially outward direction. A no-slip boundary condition is imposed on the airfoil boundary. Since the flow is just slightly supersonic, the bow shock formed is located at some distance from the leading edge of the airfoil. There are also two weak shocks emanating from the trailing edge. A viscous boundary layer is also formed on the airfoil surface. The functional used here is the total drag force on the airfoil in the direction of the free-stream flow which comprises the pressure and viscous drag components. The ideal drag coefficient is taken to be 0.20882 based on simulations performed by Hartmann and Houston.[69]

Figure 20(a) shows the density distribution for this case after a converged solution has been obtained after 8 gradient-based refinements on the initial mesh. Figure 20(b) shows the distribution of the 1^{st} component of the adjoint solution vector for this case after a converged solution has been obtained after 5 output-based

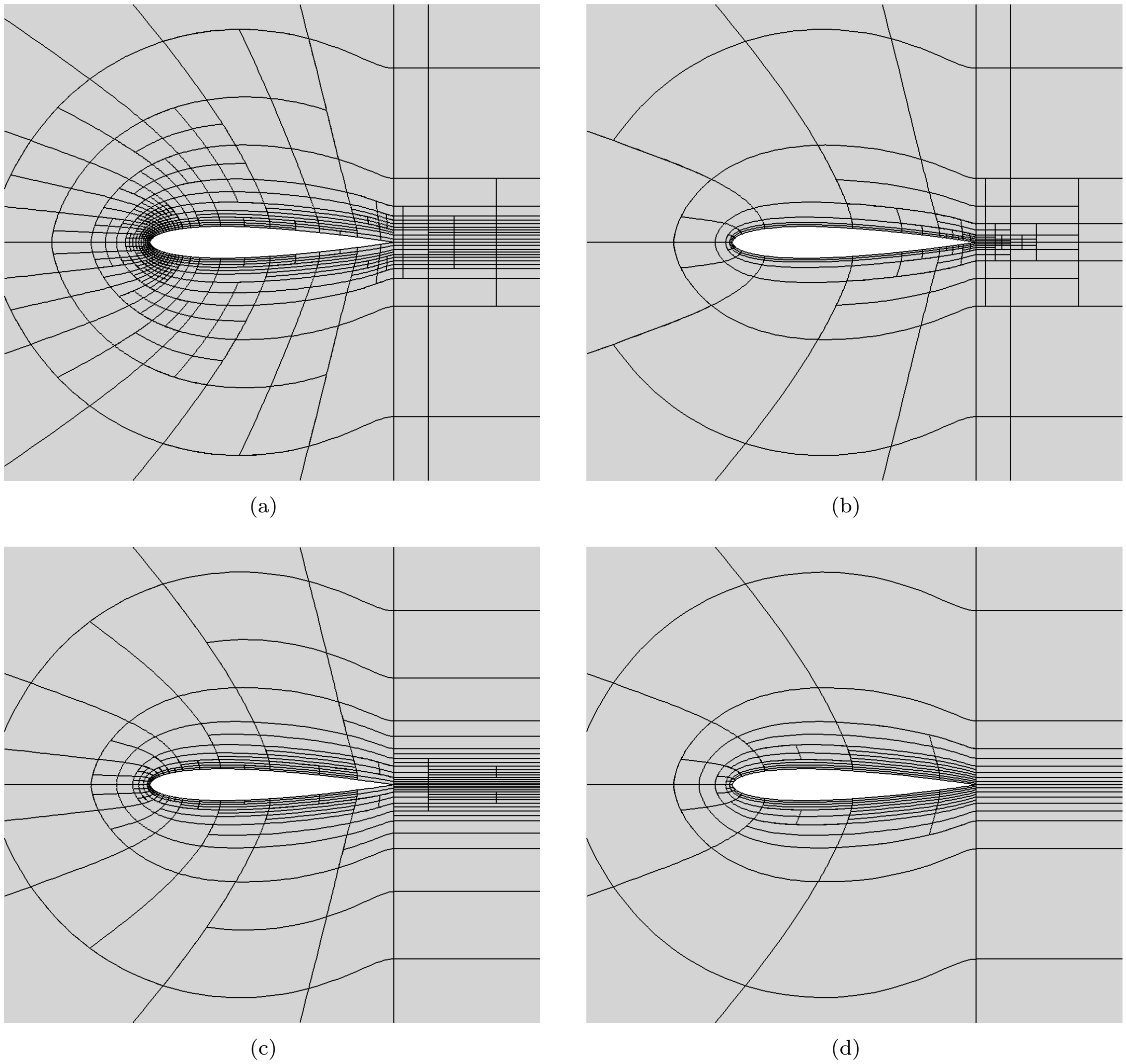
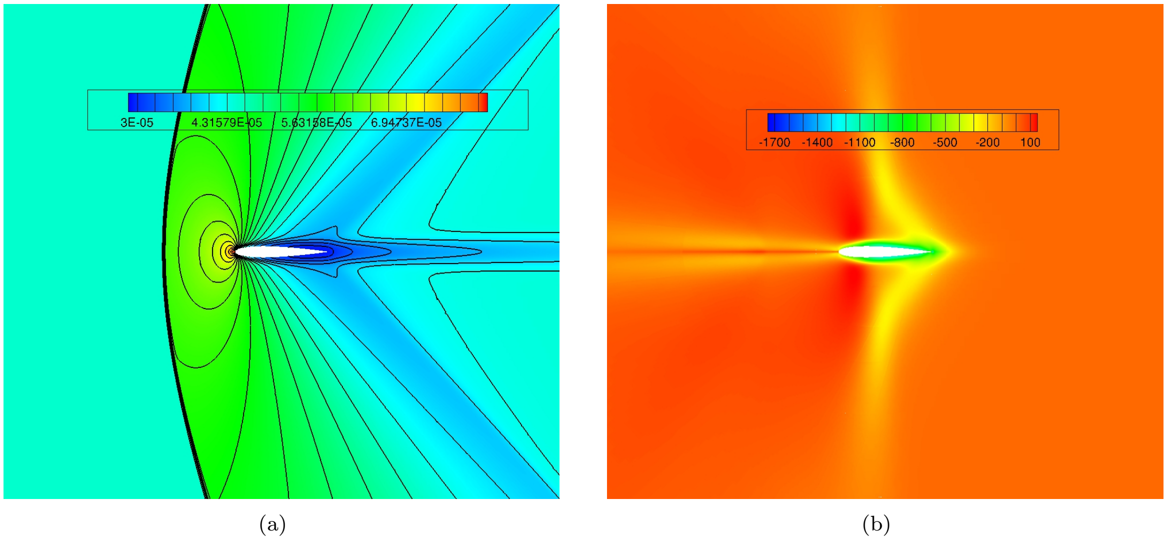


Figure 19. Computational meshes for a Mach number M=0.5 steady subsonic viscous flow past a NACA0012 airfoil showing: (a) a refined anisotropic mesh after 6 refinements obtained using a density gradient-based refinement strategy (total number of blocks is 776, number of cells per block is $16 \times 16 \times 4 = 1024$, and total number of cells is 794624); (b) a refined anisotropic mesh after 6 refinements obtained using a velocity-gradient-based refinement strategy (total number of blocks is 246, number of cells per block is $16 \times 16 \times 4 = 1024$, and the total number of cells is 251904); (c) a refined anisotropic mesh after 6 refinements obtained using an error-based refinement strategy with computable correction (total number of blocks is 334, number of cells per block is $16 \times 16 \times 4 = 1024$, and total number of cells is 342016); and (d) a refined anisotropic mesh after 6 refinements obtained using an error-based refinement strategy with Venditti-Darmofal criterion (total number of blocks is 154, total number of cells per block is $16 \times 16 \times 4 = 1024$, and the total number of cells is 157696).

refinements on the initial mesh. The gradient-based refinement method extensively refines the leading edge bow shock and also a substantial portion of the two weak trailing edge shocks. The error-based method, on the other hand, focuses primarily on a small portion of the leading edge bow shock where the adjoint variable is a non-zero value and does not refine the downstream trailing edge shocks at all, leading to significant mesh savings. Figure 21 shows a plot of the error in the functional, i.e., the difference between the ideal value $C_{D,a}$ and the computed value $C_{D,c}$ of the drag coefficient against $\sqrt[3]{N_{cells}}$ where N_{cells} is the total number of grid cells in the computational domain. The error-based methods perform better than the gradient-based method in terms of savings in the computational mesh, leading to a reduction in the mesh size by about 70%. Using the criterion defined in Equation (23) leads to further reductions in mesh size. Figure 22 depicts a comparison of the AMR meshes obtained after six successive refinements using the various refinement techniques.



(a)

(b)

Figure 20. Predicted density (a) and density adjoint (b) distributions for a Mach number M=1.2 steady supersonic viscous flow past a NACA0012 airfoil on refined anisotropic meshes. For (a), the total number of blocks is 6188, the number of cells per block is 16 × 16 × 4 = 1024, and the total number of cells is 6336512. For (b), the total number of blocks is 436, the number of cells per block is 16 × 16 × 4 = 1024, and the total number of cells is 446464.

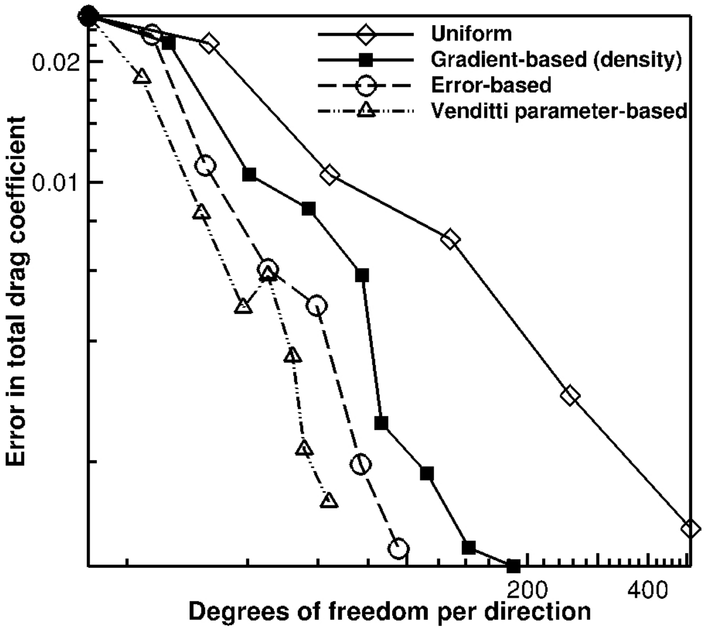
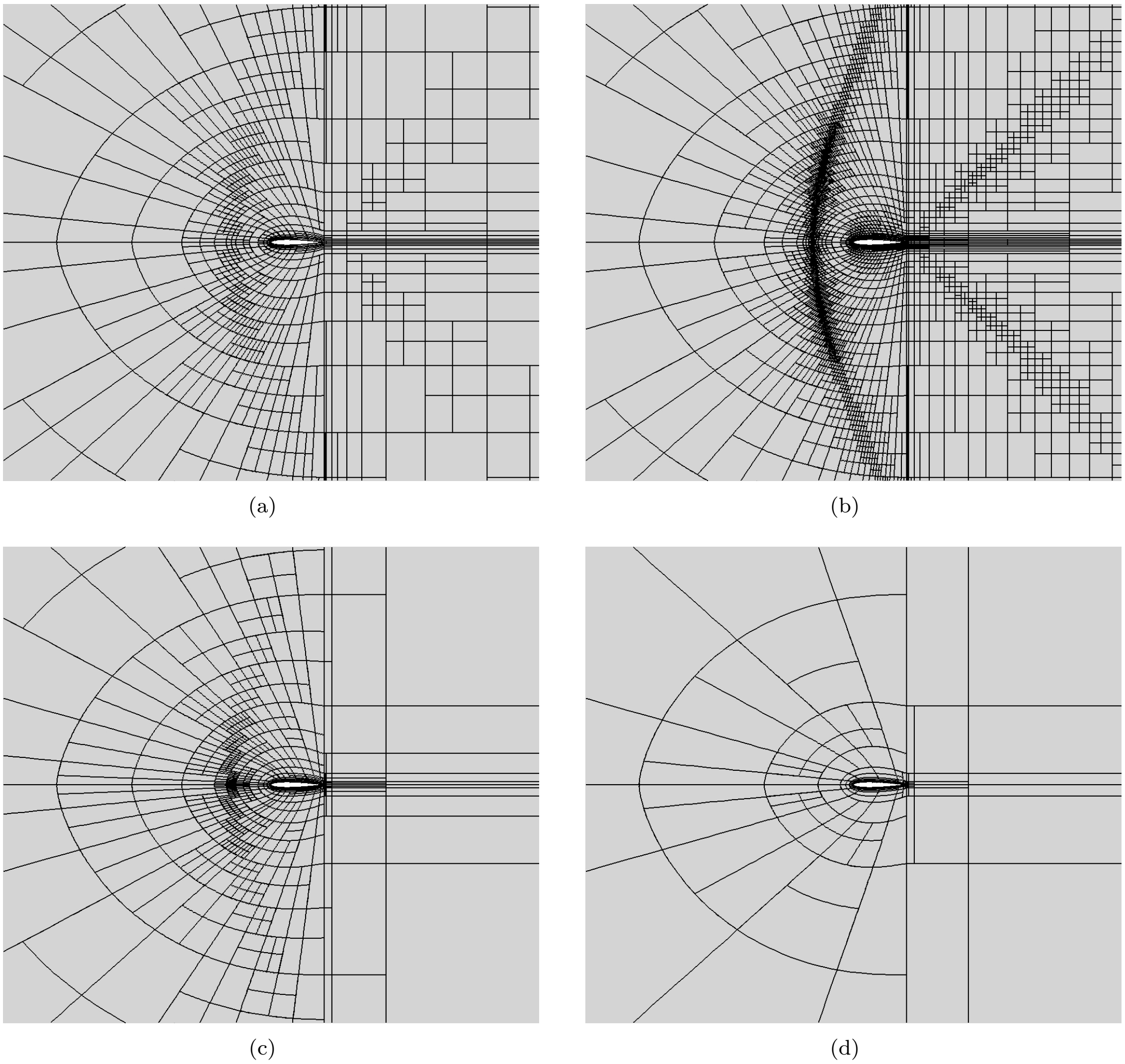


Figure 21. Convergence of the error in the predicted drag coefficient as a function of the mesh size (numbers of degrees of freedom) for supersonic viscous flow past a NACA0012 airfoil.

V. Conclusions

An adjoint-based error estimation and anisotropic mesh refinement technique that provides local solution-dependent adaptation of the computational mesh based on estimated errors in engineering functionals has been proposed and developed for the efficient prediction of three-dimensional compressible flows on hexahedral multi-block meshes using distributed-memory parallel computing systems. Results have been obtained for steady inviscid flow cases governed by the inviscid Euler equations and steady viscous flow cases governed by the Navier-Stokes equations. The potential and computational performance of the proposed output-based



(a)

(b)

(c)

(d)

Figure 22. Computational meshes for a Mach number M=1.2 steady supersonic viscous flow past a NACA0012 airfoil showing: (a) a refined anisotropic mesh after 6 refinements obtained using a density gradient-based strategy (total number of blocks is 1374, number of cells per block is $16 \times 16 \times 4 = 1024$, and total number of cells is 1406976); (b) a refined anisotropic mesh after 8 refinements obtained using a density gradient-based strategy (total number of blocks is 6188, number of cells per block is $16 \times 16 \times 4 = 1024$, and total number of cells is 6336512); (c) a refined anisotropic mesh after 6 refinements obtained using an error-based refinement strategy with computable correction (total number of blocks is 840, number of cells per block is $16 \times 16 \times 4 = 1024$, and total number of cells is 860160); and (d) a refined anisotropic mesh after 7 refinements obtained using an error-based refinement strategy with Venditti-Darmofal criterion (total number of blocks is 252, number of cells per block is $16 \times 16 \times 4 = 1024$, and total number of cells is 258048).

mesh refinement method has been demonstrated. It was found that the output-based anisotropic AMR approach is beneficial in terms of the mesh savings as compared to more standard gradient-based methods and can furnish highly accurate calculations of engineering functionals. Future research will consider the application of the proposed output-based anisotropic AMR scheme to reactive flow problems, such as premixed and non-premixed steady laminar flames, as well as to the simulation of space physics phenomena governed by the equations of ideal magnetohydrodynamics.

Acknowledgments

This work was supported by the Canadian Space Agency and by the Natural Sciences and Engineering Research Council (NSERC) of Canada. In particular, the authors would like to acknowledge the financial support received from the Canadian Space Agency through the Geospace Observatory Canada program.

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.

References

[1] Berger, M. J. and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, 1989, pp. 64–84.

[2] Berger, M. J. and Saltzman, J. S., "AMR on the CM-2," *Applied Numerical Mathematics*, Vol. 14, 1994, pp. 239–253.

[3] Berger, M. J., *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, Ph.D. thesis, Stanford University, January 1982.

[4] Berger, M. J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *Journal of Computational Physics*, Vol. 53, 1984, pp. 484–512.

[5] Berger, M. J., "Data Structures for Adaptive Grid Generation," *SIAM Journal for Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 904–916.

[6] Aftosmis, 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.

[7] Quirk, J. J. and Hanebutte, U. R., "A Parallel Adaptive Mesh Refinement Algorithm," Report 93-63, ICASE, August 1993.

[8] Pantano, C., Deiterding, R., Hill, D. J., and Pullin, D. I., "A Low Numerical Dissipation Patch-Based Adaptive Mesh Refinement Method for Large-Eddy Simulation of Compressible Flows," *Journal of Computational Physics*, Vol. 221, 2007, pp. 63–87.

[9] De 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.

[10] Berger, M. J. and LeVeque, R. J., "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries," Paper 89-1930, AIAA, June 1989.

[11] Ripley, R. C., Lien, F. S., and Yovanovich, M. M., "Adaptive Mesh Refinement of Supersonic Channel Flows on Unstructured Meshes," *International Journal of Computational Fluid Dynamics*, Vol. 18, No. 2, 2004, pp. 189–198.

[12] Groth, C. P. T., De Zeeuw, D. L., Gombosi, T. I., and Powell, K. G., "A Parallel Adaptive 3D MHD Scheme for Modeling Coronal and Solar Wind Plasma Flows," *Space Science Reviews*, Vol. 87, 1999, pp. 193–198.

[13] Groth, C. P. T., De Zeeuw, D. L., Powell, K. G., Gombosi, T. I., and Stout, Q. F., "A Parallel Solution-Adaptive Scheme for Ideal Magnetohydrodynamics," Paper 99-3273, AIAA, June 1999.

[14] van der Holst, B. and Keppens, R., "Hybrid Block-AMR in Cartesian and Curvilinear Coordinates: MHD Applications," *Journal of Computational Physics*, Vol. 226, 2007, pp. 925–946.

[15] Gao, 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.

[16] Northrup, S. A., *A Parallel Implicit Adaptive Mesh Refinement Algorithm for Predicting Unsteady Fully-Compressible Reactive Flows*, Ph.D. thesis, University of Toronto, December 2013.

[17] Charest, M. R. J., Groth, C. P. T., and Gülder, O. L., "Numerical Study on the Effect of Gravity on Flame Shape and Radiation in Laminar Diffusion Flames," *Proceedings of the Combustion Institute Canadian Section Spring Technical Meeting*, Toronto, Canada, May 12–14 2008, pp. 84–89.

[18] Gao, X. and Groth, C. P. T., "Parallel Adaptive Mesh Refinement Scheme for Three-Dimensional Turbulent Non-Premixed Combustion," Paper 2008-1017, AIAA, January 2008.

[19] Gao, X., *A Parallel Solution-Adaptive Method for Turbulent Non-Premixed Combusting Flows*, Ph.D. thesis, University of Toronto, August 2008.

[20] Charest, M. R. J. and Groth, C. P. T., "A High-Order Central ENO Finite-Volume Scheme for Three-Dimensional Turbulent Reactive Flows on Unstructured Mesh," Paper 2013-2567, AIAA, June 2013.

[21] Freret, L. and Groth, C. P. T., "Anisotropic Non-Uniform Block-Based Adaptive Mesh Refinement for Three-Dimensional Inviscid and Viscous Flows," Paper 2015-2613, AIAA, June 2015.

[22] Northrup, S. A. and Groth, C. P. T., "Parallel Implicit Adaptive Mesh Refinement Scheme for Unsteady Fully-Compressible Reactive Flows," Paper 2013-2433, AIAA, June 2013.

[23] Northrup, 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.

[24] Gao, X., Northrup, S. A., and Groth, C. P. T., "Parallel Solution-Adaptive Method for Two-Dimensional Non-Premixed Combusting Flows," *Progress in Computational Fluid Dynamics*, Vol. 11, No. 2, 2011, pp. 76–95.

[25] Gao, X. and Groth, C. P. T., "Parallel Adaptive Mesh Refinement Scheme for Turbulent Non-Premixed Combusting Flow Prediction," Paper 2006-1448, AIAA, January 2006.

[26] Gao, 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.

[27] Jha, P. K. and Groth, C. P. T., "Tabulated Chemistry Approaches for Laminar Flames: Evaluation of Flame-Prolongation of ILDM and Flamelet Methods," *Combustion Theory and Modelling*, Vol. 16, No. 1, 2012, pp. 31–57.

[28] Jha, P. K., *Modelling Detailed-Chemistry Effects on Turbulent Diffusion Flames Using a Parallel Solution-Adaptive Scheme*, Ph.D. thesis, University of Toronto, October 2011.

[29] Ivan, L., De Sterck, H., Susanto, A., and Groth, C. P. T., "High-Order Central ENO Finite-Volume Scheme for Hyperbolic Conservation Laws on Three-Dimensional Cubed-Sphere Grids," *Journal of Computational Physics*, Vol. submitted, 2014.

[30] Ivan, L., De Sterck, H., Northrup, S. A., and Groth, C. P. T., "Hyperbolic Conservation Laws on Three-Dimensional Cubed-Sphere Grids: A Parallel Solution-Adaptive Simulation Framework," *Journal of Computational Physics*, Vol. 255, 2013, pp. 205–227.

[31] Ivan, L., De Sterck, H., Northrup, S. A., and Groth, C. P. T., "Three-Dimensional MHD on Cubed-Sphere Grids: Parallel Solution-Adaptive Simulation Framework," Paper 2011-3382, AIAA, June 2011.

[32] Susanto, A., Ivan, L., Sterck, H. D., and Groth, C. P. T., "igh-Order Central ENO Finite-Volume Scheme for Ideal MHD," *Journal of Computational Physics*, Vol. 250, 2013, pp. 141–164.

[33] Zhang, Z. J. and Groth, C. P. T., "Parallel High-Order Anisotropic Block-Based Adaptive Mesh Refinement Finite-Volume Scheme," Paper 2011-3695, AIAA, June 2011.

[34] Williamschen, M. J. and Groth, C. P. T., "Parallel Anisotropic Block-Based Adaptive Mesh Refinement Algorithm For Three-Dimensional Flows," Paper 2013-2442, AIAA, June 2013.

[35] Freret, L., Ivan, L., De Sterck, H., and Groth, C. P. T., "A High-Order Finite-Volume Method with Anisotropic AMR for Ideal MHD Flows," Paper 2017-0845, AIAA, January 2017.

[36] Heuveline, V. and Rannacher, R., "Duality-Based Adaptivity in the *hp*-Finite Element Method," *Journal of Numerical Mathematics*, Vol. 11, 2003, pp. 95–103.

[37] Becker, R., Heuveline, V., and Rannacher, R., "An Optimal Control Approach to Adaptivity in Computational fluid Dynamics," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 105–120.

[38] Venditti, D. A. and Darmofal, D. L., "Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow," *Journal of Computational Physics*, Vol. 164, 2000, pp. 204– 227.

[39] Venditti, D. A. and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows," *Journal of Computational Physics*, Vol. 176, 2002, pp. 40– 69.

[40] Venditti, D. A. and Darmofal, D. L., "Anisotropic Grid Ddaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, 2003, pp. 22– 46.

[41] Nemec, M. and Aftosmis, M. J., "Adjoint Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes," Paper 2007-4187, AIAA, June 2007.

[42] Nemec, M. and Aftosmis, M. J., "Adjoint Sensitivity Computations for an Embedded-Boundary Cartesian Mesh Method," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2724–2742.

[43] Ceze, M. and Fidkowski, K., "A Robust Adaptive Solution Strategy for High-Order Implicit CFD Solvers," Paper 2011-3696, AIAA, 2011.

[44] Yano, M. and Darmofal, D. D., "An Optimization-Based Framework for Anisotropic Simplex Mesh Adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, 2012, pp. 7626 – 7649.

[45] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer-Verlag, New York, 1999.

[46] Einfeldt, B., "On Godunov-Type Methods for Gas Dynamics," *SIAM Journal on Numerical Analysis*, Vol. 25, 1988, pp. 294–318.

[47] Saad, 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.

[48] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI*, MIT Press, Cambridge, Massachussets, 1999.

[49] Gropp, W., Lusk, E., and Thakur, R., *Using MPI-2*, MIT Press, Cambridge, Massachussets, 1999.

[50] Nadarajah, S. K. and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," Paper 2000-0667, AIAA, 2000.

[51] Nadarajah, S. L., *A Discrete Adjoint Approach to Aerodynamic Shape Optimization*, Ph.D. thesis, Stanford University, January 2003.

[52] Nemec, M. and Zingg, D. W., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.

[53] Nadarajah, S. K. and Jameson, A., "Optimum Shape Design for Unsteady Flows with Time-Accurate Continuous and Discrete Adjoint Methods," *AIAA Journal*, Vol. 45, No. 7, July 2007.

[54] Thomas, J. P., Hall, K. C., and Dowell, E. H., "Discrete Adjoint Approach for Modeling Unsteady Aerodynamic Design Sensitivities," *AIAA Journal*, Vol. 43, No. 9, September 2005.

[55] Sun, S. and Wheeler, M. F., "A Posteriori Error Estimation and Dynamic Adaptivity for Symmetric Discontinuous Galerkin Approximations Reactive Transport Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, 2006, pp. 632–652.

[56] Becker, R., Braack, M., and Vexler, B., "Numerical Parameter Estimation for Chemical Models in Multidimensional Reactive Flows," *Combustion Theory and Modelling*, Vol. 8, 2004, pp. 661–682.

[57] Heroux, M., Bartlett, R., Howle, V., Hoekstra, R., Hu, J., Kolda, T., Lehoucq, R., Long, K., Pawlowski, R., Phipps, E., Salinger, A., Thornquist, H., Tuminaro, R., Willenbring, J., and Williams, A., "An Overview of Trilinos," Tech. Rep. SAND2003-2927, Sandia National Laboratories, 2003.

[58] Heroux, M. A., Willenbring, J. M., and Heaphy, R., "Trilinos Developers Guide," Tech. Rep. SAND2003-1898, Sandia National Laboratories, 2003.

[59] Heroux, M. A., Willenbring, J. M., and Heaphy, R., "Trilinos Developers Guide Part II: ASCI Software Quality Engineering Practices Version 1.0," Tech. Rep. SAND2003-1899, Sandia National Laboratories, 2003.

[60] Sala, M., Heroux, M. A., and Day, D. M., "Trilinos Tutorial," Tech. Rep. SAND2004-2189, Sandia National Laboratories, 2004.

[61] Heroux, M. A. and Willenbring, J. M., "Trilinos Users Guide," Tech. Rep. SAND2003-2952, Sandia National Laboratories, 2003.

[62] Heroux, M. A., "Epetra Reference Manual, 2.0 Edition," Tech. rep., Sandia National Laboratories, 2002.

[63] Heroux, M. A., "AztecOO Users Guide," Tech. Rep. SAND2004-3796, Sandia National Laboratories, 2004.

[64] Ivan, L. and Groth, C. P. T., "High-Order Solution-Adaptive Central Essentially Non-Oscillatory (CENO) Method for Viscous Flows," *Journal of Computational Physics*, Vol. 257, 2013, pp. 830–862.

[65] Kannan, R. and Wang, Z. J., "A Study of Viscous Flux Formulations for a p-Multigrid Spectral Volume Navier Stokes Solver," *Journal of Scientific Computing*, Vol. 41, No. 2, 2009, pp. 165–199.

[66] Kannan, R. and Wang, Z. J., "A Variant of the LDG Flux Formulation for the Spectral Volume Method," *Journal of Scientific Computing*, Vol. 46, No. 2, 2011, pp. 314–328.

[67] Sun, Y., Wang, Z. J., and Liu, Y., "Efficient Implicit Non-linear LU-SGS Approach for Compressible Flow Computation Using High-Order Spectral Difference Method," *Communications in Computational Physics*, Vol. 5, No. 2–4, 2009, pp. 760–778.

[68] D. J. Mavriplis, A. J. and Martinelli, J., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," Paper 89-0120, AIAA, 1989.

[69] Hartmann, 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.