Output-Based Error Estimation and Anisotropic AMR for Numerical Prediction of Steady Ideal Magnetohydrodynamic Flows

by

Nishant Narechania

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto

Abstract

Output-Based Error Estimation and Anisotropic AMR for Numerical Prediction of
Steady Ideal Magnetohydrodynamic Flows

Nishant Narechania
Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto
2020

An anisotropic output-based adaptive mesh refinement (AMR) scheme is proposed and developed for the numerical prediction of ideal magnetohydrodynamic (MHD) 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 to solve the governing system of partial-differential equations on multi-block body-fitted hexahedral meshes. An inexact Newton's method is used to obtain steady state solutions of the discretized governing equations. The anisotropic block-based refinement technique provides significant reductions in the size of the computational mesh by locally refining the grid only in certain directions as dictated by the flow physics and numerical solution. A flexible binary tree data structure facilitates efficient parallel AMR via domain decomposition. An output-based error estimation procedure enables formal evaluation of *a posteriori* estimates of the errors of solution-dependent engineering functionals of interest 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 adjoint solution 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 anisotropic mesh adaptation and the output-based refinement strategy generates meshes which are customized for the accurate calculation of functionals of interest. The performance of the output-based mesh refinement scheme is first demonstrated for several representative steady time-invariant aerodynamic flows governed by the Euler and Navier-Stokes equations and finally for several space physics flows governed by the equations of ideal MHD. Comparisons are made between the output-based mesh refinement technique and more traditional gradient-based and uniform refinement approaches in terms of efficient reduction in error and number of cells required to obtain a certain accuracy level. A solar wind-MHD model employing photospheric magnetic field maps obtained from solar magnetograms as inputs is also developed as a test case for the proposed output-based anisotropic mesh refinement scheme.

This thesis is dedicated to my parents,
Jayashree and Mayur Narechania,
and sister, Dr. Shraddha Narechania.

Acknowledgements

Firstly, I would like to thank my advisor, Professor Clinton P. T. Groth for his guidance, patience, insight, enthusiasm and for sharing his tremendous knowledge and experience in the field of computational fluid dynamics (CFD). I am also grateful to him for providing me opportunities to showcase my research and gain deeper knowledge of the field through several conferences. I am also grateful to Professors Ömer Gülder and Masayuki Yano for their constructive evaluations of my research work.

I am extremely grateful to Dr. Lucie Freret who helped me in learning crucial aspects of the software Trilinos and MPI, understanding and developing various modules of the group code and who was always ready to help in dealing with various obstacles I faced throughout the course of my thesis. It would not have been possible to complete this thesis in a timely manner without her guidance.

I would like to thank Dr. Ljubomir Nikolic for his contributions in developing the space weather prediction framework and Natural Resources Canada (NRCan) for hosting me as a visiting researcher during summer 2017. I would like to thank my colleagues in the CFD and Propulsion Group for their fruitful discussions. Dr. Scott Northrup deserves a special mention for his prompt assistance in tackling issues relating to SciNet.

UTIAS introduced me to my girlfriend, Lu-Yin Wang, and gave me an opportunity to forge a strong friendship with Pankaj Saini. I am grateful to UTIAS for introducing me to many other friends such as Ehsan Vaziri, Allen Chee, Moah Christensen, Ravindu Lokuwithana, Gregory Moszczynski, Kattie Smilovsky, Neel Young, Jenmy Zhang, Jacek Khan and more recently Frank Lang, Christopher Schneider, Rahul Vishwanath, Rajat Sawanni, Suraj Bansal, Ral Bielawski and Joshua Furtado.

I am thankful to my former landlords Vanja and Zoran Velimirovic for providing a loving and caring home away from home for the first three years of my thesis. Most importantly, I would like to thank my parents and sister for their unconditional love and support for all the pursuits and endeavours I have undertaken so far in my life.

Computational resources for this work were provided by the SciNet High Performance Computing Consortium and Compute Canada. A portion of my research funding also came from the NSERC Collaborative Research and Training Experience (CREATE) Program in Environmentally Sustainable Aviation.

Nishant Narechania
University of Toronto Institute for Aerospace Studies

Contents

List of Figures

Chapter 1

Introduction and Motivation

1.1 Space Weather

With the advent of the space age, the study of space weather and its impact on human activity and life on Earth has gained in significance. The development of novel space and ground-based technologies for communication, Earth observation and space exploration will continue to play a significant role in the advancement of human life in the foreseeable future. "Space Weather" [2, 3] is a term that is used to refer to conditions on the Sun and in the solar wind and geospace environment, which includes the lithosphere, hydrosphere, atmosphere, ionosphere, and magnetosphere, that can influence the performance and reliability of both space-borne and ground-based technological systems or can adversely affect human life or health on Earth. Adverse changes in space weather can cause disruptions to ground-based electric power transmission grids. Geomagnetically induced currents can lead to increased corrosion of pipeline systems for oil, natural gas and water supply. Space weather can also disrupt the operations of satellites in near-Earth orbits and communication and GPS navigation systems. Astronauts are also vulnerable to the energetic radiation from the Sun.

The Sun and the resulting plasma environment within the heliosphere are the key drivers of space weather. With the continual depletion of Earth's natural resources and man's possible dependence on space as a potential frontier of industrial growth, space weather is now considered a natural hazard as the risk it poses to human activity and even health cannot be ignored. There is a current need for a significantly improved fundamental understanding of the solar-related physical processes affecting space weather in order to be able to both forecast and assess the impact of events in an analogous fashion to that now possible for atmospheric weather prediction.

1.2 Numerical Simulation of Space Plasmas

Numerical simulations offer a convenient path to understanding the behaviour of space plasmas, as they circumvent what can be the enormous costs of performing in situ and even remote measurements in outer space. Numerical techniques can be used to study phenomena covering distances of interplanetary scales that are otherwise impossible to capture with the current technological resources. However, the development of numerical techniques for the study and realistic prediction of space weather is extremely challenging. For instance, the solar wind contains electrons, protons, α-particles and traces of heavier ions, requiring multiple species equations for realistic and meaningful solutions. In addition, the expansion of the solar wind as it propagates outward from the Sun into interplanetary space leads to a large variation in length and time scales that pose a challenge to capture accurately by numerical methods. Due to the current capabilities of available computational resources, numerical studies of space plasmas heavily rely on global modelling techniques based on the equations of ideal magnetohydrodynamics (MHD) [4,5], wherein the underlying physics is accounted for in an approximate manner depending on the degree of sophistication of the particular MHD model.

Theoretical and numerical modelling of the solar wind and heliospheric flows has a long history and it is beyond the scope of this thesis to provide a complete review of previous research in this area. Nevertheless, a brief summary of previous research efforts relevant to the thesis are provided in what follows. In the last 30-40 years, various attempts have been made to model space weather using the equations of MHD. Pizzo [6] studied the structure of the solar wind between $35R_{Sun}$ and 1 AU using an idealized model of the inner heliosphere. Usmanov [7] subsequently developed a fully 3D steady state model of the solar corona and heliosphere using magnetic field observations for Carrington rotation CR 1682 and made comparisons with spacecraft observations at 1 AU. Lionello *et al.* [8] also solved the equations of MHD to study the propagation of the solar wind on a cylindrical mesh. Linker *et al.* [9] modelled the solar corona during Whole Sun Month[1] from $1R_{Sun}$ to $30R_{Sun}$ using photospheric field observations as boundary conditions and compared results with *SOHO*, *Ulysses* and *WIND* data. BATS-R-US, a parallel block-adaptive numerical framework was developed at the University of Michigan for the global MHD simulation of space weather [10–17]. This global MHD model was used by Powell *et al.* [10] to simulate the interaction between the solar wind

[1] Whole Sun Month was a two-year collaborative effort of the Inter-Agency Consultative Group (IACG) and the Solar and Heliospheric Observatory (SOHO) Working Group in which an international and interdisciplinary group of scientists studied the Sun from August 8 to September 10, 1996, a period known as solar minimum.

and a planetary magnetosphere. Moreover, Groth *et al.* [11] studied the propagation of a coronal mass ejection (CME) in a steady background solar wind, using an octupole model for the Sun's magnetic field, a pressure pulse for modelling the CME and source terms for the solar wind acceleration and heating effects. The latter represented the first-in-the-world global MHD simulation of a complete fully three-dimensional space weather event, spanning the initiation of a solar wind disturbance at the Sun's surface to its interaction with the Earth's magnetosphere.

In subsequent follow-on studies by the Michigan group, Roussev *et al.* [12] modelled the corona-heliosphere system by implementing a continuous variation in the polytropic index in a radially outward direction from the Sun. Manchester *et al.* [13, 14] used an idealized model of the steady state solar wind conditions near solar minimum, studied the three-dimensional propagation of a flux rope-driven CME and made comparisons with coronagraph observations of CMEs. Tóth *et al.* [15] and Manchester *et al.* [16] used synoptic magnetograms to model the CME event of October 28, 2003 and made comparisons with observations. The Michigan BATS-R-US global MHD model was eventually extended to solve other forms of the MHD equations, including the Hall MHD, multi-fluid MHD, and radiative MHD models [17]. Van der Holst *et al.* [18] implemented a two-temperature MHD model wherein low-frequency Alfvén wave turbulence was modelled to account for coronal heating and solar wind acceleration. They simulated extreme ultraviolet (EUV) images of Carrington rotation CR2107 and made comparisons with STEREO and SDO observations.

In other MHD modelling efforts, Riley *et al.* [19] decoupled the MHD simulation in the corona and heliosphere using suitable polytropic indices for each model and subsequently used the coronal simulation as a driver for the heliospheric calculation. The combined model was used to generate the heliospheric structure during Carrington rotations CR 1913, CR 1892 and CR 1947. Similarly, Odstrčil *et al.* [20] studied CME propagation using a resistive MHD model with a ratio of specific heats, $\gamma = 1.05$ for the coronal calculation upto $20R_{Sun}$ and an ideal MHD model with a ratio of specific heats, $\gamma = 5/3$ for the heliospheric calculation. Lionello *et al.* [21] performed MHD simulations of the corona using various coronal heating models and reproduced observed multispectral properties of the corona. More recently, Feng *et al.* [22] employed a six-component overset grid to study the background solar wind from Sun to Earth during Carrington rotation CR 1911 using line-of-sight photospheric field observations and validated their MHD model using *SOHO* and *WIND* observations. Feng *et al.* [23] later added isotropic block-based AMR capabilities to this model. Merkin *et al.* [24] applied an ideal MHD model for the heliosphere and a more realistic MHD model for the corona that contained terms to account for radiative losses, coronal heating, thermal conduction and magnetic resistivity.

Lastly, the MPI-AMRVAC software is an open source toolkit for parallel, block-adaptive MHD simulations of solar and non-relativistic astrophysical plasmas [25, 26]. MPI-AMRVAC also has a relativistic counterpart, BHAC [27], for solving the equations of ideal general-relativistic magnetohydrodynamics (GRMHD) for studying astrophysical phenomena such as black holes. The MPI-AMRVAC software has also been used for studying the formation of prominences in the solar corona using magnetic flux rope models of various levels of complexity [28–32]. Most recently, MPI-AMRVAC source has been used for the study of magnetic reconnection of solar flares and to simulate the trans-Alfvénic solar wind from the Sun to the Earth using a solar wind model replicating solar minimum conditions and artificial heating/cooling source terms [33].

In many of the global MHD models described above, rather sophisticated numerical simulation techniques have been adopted and applied to the solution of the governing equations, including both adaptive mesh refinement strategies and high-order discretization schemes. Nevertheless, the computational costs associated with performing such global MHD simulations are still significantly high and improved and more efficient numerical solution techniques are required in order to make such simulations more routine for the prediction of space weather and solar wind phenomena.

1.2.1 Adaptive Mesh Refinement (AMR)

Adaptive mesh refinement (AMR) [34–39] is one of the ways to make more optimal use of the available degrees of freedom in order to maximize the solution accuracy when used in combination with numerical methods. AMR involves an increase in the number of grid cells in regions having relatively smaller length scales and conversely a decrease in number of grid cells in regions having relatively larger length scales. In this way, AMR is extremely beneficial for the treatment of problems with a wide range of length scales, since the mesh can coarsen or refine automatically as dictated by the length scales prevalent in a particular region of the flow field. For fixed computational resources, AMR also provides a way to capture more of the essential flow physics without having to resort to simplified flow models. Additionally, automatic adaption of meshes also reduces the need for human intervention. Currently, a significant amount of effort is spent on the initial mesh generation for a given flow problem. For instance, the mesh near the surface of an airfoil must be sufficiently resolved in order to capture the boundary layer for accurate prediction of the viscous drag force. An initial mesh with inadequate resolution of the boundary layer can give inaccurate or even completely wrong solutions. However, effective automatic adaptation of meshes would be able to recognize the need for increased resolution in the boundary layer, thus increasing the number of cells in that region. Hence, an initially coarse mesh that requires a relatively shorter amount of human time and effort to generate would be all that is needed, as long as it automatically refines during runtime, as per the requirements for the desired quality of the solution.

To date, a number of hierarchical-type AMR techniques have been developed and they can be broadly categorized into four major types: patch-based AMR, cell-based AMR, block-based AMR and hybrid AMR. These different categories of AMR are depicted in Figure 1.1. Figure 1.1(a) shows the starting mesh with cells flagged for refinement. Figures 1.1(b)-1.1(d) indicate the consecutive mesh obtained after employing the patch-based, cell-based and block-based AMR techniques, respectively. In the patch-based AMR, originally proposed by Berger and Oliger [36] and Collela [34], individual cells are first flagged for refinement. Flagged cells that are close to each other are grouped together using a sophisticated algorithm to form rectangular patches as shown in Figure 1.1(b). These patches can be further refined to form nested patches having a higher refinement level. Patch-based AMR has been considered by a number of researchers to study a variety of fluid flows [34, 36, 37, 40, 41].

In cell-based AMR [38, 42–45], the flagged cells are individually refined as shown in Figure 1.1(c). Each flagged cell forms four new cells in two dimensions and eight new cells in three dimensions. The connectivity between individual cells is stored using a flexible tree data structure. In block-based AMR [1, 17, 25, 26, 33, 35, 46–55], the computational domain is subdivided into clusters of grid cells that are called blocks. These blocks are composed of a pre-defined, generally fixed, number of cells. Entire blocks, that contain cells flagged for refinement, are refined to form multiple blocks containing the same number of cells, as shown in Figure 1.1(d). Tree-based data structures are again used to store the connectivity between individual blocks. It should be noted that Holst and Keppens [56] have developed a hybrid AMR approach which combines aspects of the patch-based and block-based AMR strategies.

1.2.2 Block-Based AMR

Typically, for the same number of individual cells flagged on a given mesh, the subsequent refined mesh created by a block-based AMR approach is generally larger in terms of the total number of cells than that created by a similar cell-based AMR method, due to the application of the refinement and coarsening procedures to the entire original

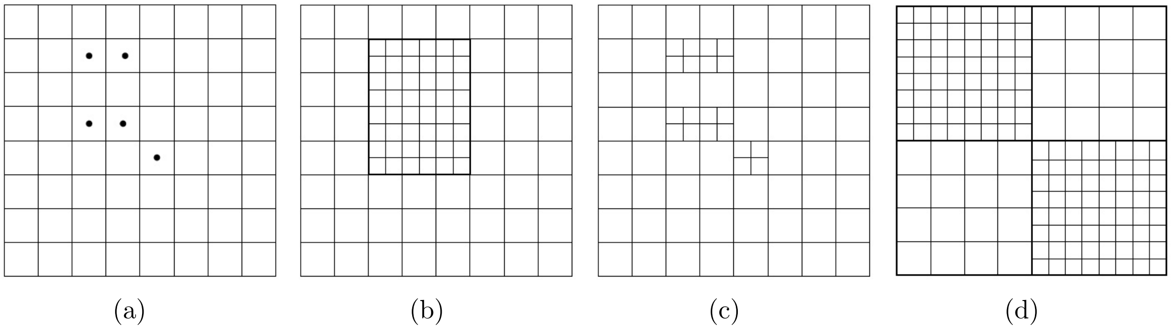


Figure 1.1: Mesh refinement categories: (a) cells flagged, (b) patch-based, (c) cell-based, (d) block-based.

grid block. In block-based AMR, each block also contains additional layers of "ghost" cells that overlap with neighbouring blocks. These "ghost" cells are used to share and exchange solution information between neighbouring blocks, allowing independent updates of solution quantities on each block. In this sense, the block-based methods are therefore sub-optimal. Nevertheless, in block-based AMR, the blocks can be self-similar and load-balancing is then almost trivial to achieve in a parallel setting [1, 47–50]. Also, in block-based AMR, the tree structure is much lighter than that in cell-based AMR in terms of storage as it must only account for connectivity between entire blocks and not individual cells. Therefore, due to the self-similar nature of blocks, easily attainable load-balancing, and lighter data structure and connectivity, block-based AMR strategies are highly scalable and suitable for implementation on today's parallel high-performance computers, as has been demonstrated in various previous studies [1, 47–50, 57, 58].

Block-based AMR methods [1,47–50] can be further classified as being either isotropic or anisotropic in nature. The isotropic block-based AMR approach, as originally proposed by Gao and Groth [47–49], employs an 8-membered octree data structure, used to keep track of the connectivity between neighbouring blocks and between a block and its "parent" and "children" blocks. In a 3D simulation, each block undergoing refinement is divided into 8 blocks. Conversely, in case of coarsening, eight flagged blocks combine together to form one coarsened "parent" block. Isotropic block-based AMR has been applied to the numerical prediction of 3D non-equilibrium gaseous flows [59] and 2D compressible gaseous flows [60]. Isotropic block-based mesh refinement has also been extensively applied to combustion problems by Groth and co-workers. For example, Northrup and Groth [57, 61, 62] used the 3D isotropic block-based AMR for simulation of laminar steady and unsteady premixed and non-premixed flames. Gao and Groth [47–49,58,63,64] and Jha [65, 66] applied isotropic block-based AMR to turbulent diffusion flames. Ivan *et al.* [67–70] and Susanto *et al.* [71] have also used block-based AMR for magnetohydrodynamic simulations. Finally, the block-based AMR approach has also been combined with higher-order discretization methods by Groth and co-workers [67, 72–77] with the goal of further reducing computational costs.

In isotropic block-based AMR, each block marked for refinement is refined equally in each coordinate direction, although the solution quantities such as gradients may favour refinement in certain directions more than others, leading to unnecessary refinement of the mesh. In the anisotropic block-based AMR strategy, as recently proposed by Zhang and Groth [76], Williamschen and Groth [50] and Freret and Groth [1], a "parent" block undergoing refinement is always divided into just two blocks, in one of the three coordinate directions as dictated by the refinement criteria. For this anisotropic approach, a binary tree data structure is then used to keep track of the connectivity and the sequence of splitting of a block. Use of anisotropic AMR [50,76] can potentially reduce unnecessary refinement and subsequent CPU time and memory. Zhang and Groth [76] developed a block-based anisotropic AMR approach and applied it to 2D convection-diffusion problems and inviscid supersonic flows governed by the Euler equations. They found a 50% reduction in mesh size after using anisotropic AMR. Williamschen and Groth [50] subsequently extended this block-based anisotropic AMR approach and applied it to 3D inviscid supersonic flow solutions governed by the Euler equations. The latter found a 75% reduction in mesh size for equivalent mesh resolution and solution accuracy. Most recently, Freret and Groth [1] developed a non-uniform block-based anisotropic AMR approach and applied it to 3D inviscid and viscous flows and found similar reductions in mesh size. This non-uniform block-based anisotropic AMR approach was then extended for use with high-order central essentially non-oscillatory (CENO) finite-volume schemes in a follow-on study by Freret *et al.* [78, 79].

1.2.3 Output-Based Error Estimation for AMR

Traditionally, the goal of AMR has been to adapt the mesh to capture dominant physical features of a flow problem, such as discontinuities like shocks, flame fronts and current sheets or other characteristics like rarefactions and boundary layers. Heuristic or physics-based measures such as gradient, divergence or curl of key flow properties such as pressure, Mach number, species fraction, magnetic field, etc., are typical error indicators employed by such AMR strategies. In all of the AMR research and development carried out so far by Groth and co-workers [1,47–49,57,80,81], heuristic physics-based criteria of this type were used to direct the mesh refinement. As observed by Williamschen and Groth [50] and Freret and Groth [1], gradient-based or physics-based methods do not necessarily directly respond to errors in the numerical solution, hampering grid convergence and error control as the mesh refinement proceeds. This phenomenon can be observed in an example flow problem described in the study by Freret and Groth [1], wherein a supersonic flow of air at density $\rho_i = 10$ kg/m^3, radial velocity $V_{r,i} = 4.5$ m/s, and pressure $p_i = 26$ Pa enters through the inner boundary of a concentric spherical shell, expands and exits supersonically through the outer boundary. A spherical inflow radius of $R_i = 1$ m and a spherical outflow radius of $R_o = 4$ m were considered. An analytical solution to this outflow problem is available as described by Ivan et al. [69]. The L_1 and L_2 norms of the density solution errors were computed on a sequence of adaptively refined meshes as directed by the gradient in the solution density and are shown in Figure 1.2. It was observed that the solution error ceases to reduce after a number of gradient-based mesh adaptations. Clearly, the regions of high solution gradients were not associated with high solution error resulting in ineffective mesh refinement. The choice of adaption criteria for adaptive mesh refinement is therefore an important factor to be considered when the goal is to achieve efficient reductions in the solution error. It is expected that mesh refinement criteria based on solution error would make the AMR significantly more effective.

In particular, the application of output-based or 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. Engineering functionals are typically integrated quantities such as aerodynamic forces or moments, that are calculated by integrating solution variables over portions of the computational domain or the domain boundaries. Adjoint-based error estimation techniques allow calculation of adjoint variables that effectively represent sensitivities of the functionals to the solution error. The adjoint, coupled with the solution error can thus be used to calculate formal error estimates of the functionals. The distributions of these errors over the computational domain can be used to direct the mesh adaptation, generating customized meshes for accurately calculating functionals. Output-based methods that utilize functional error estimates for driving the AMR have been implemented in combination with various mesh refinement techniques by previous researchers for the purpose of efficiently studying a variety of fluid flows [82–90]. These methods have shown significant promise in obtaining accurate predictions of functionals, while at the same time overcoming the need to perform expensive computations on very large meshes.

As noted above, output-based methods have been used extensively for simulating fluid flow phenomena such as diffusion, heat transfer, advection, inviscid and viscous

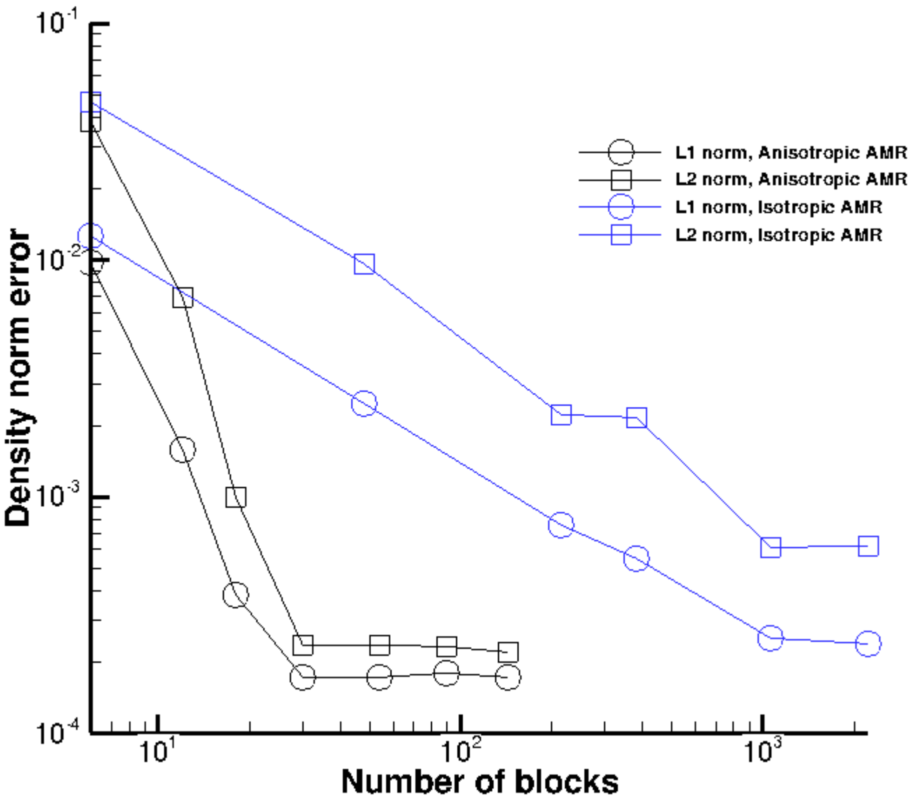


Figure 1.2: Solution error does not drop below a certain accuracy level for supersonic spherical outflow problem studied by Freret and Groth [1].

aerodynamic flows and also turbulent aerodynamic flows. For example, Heuveline and Rannacher [83] applied adjoint-based error estimation in combination with AMR to the solution of second-order elliptic equations. They used hp-adaptation on structured Cartesian finite-element meshes, wherein the mesh is adapted by increasing the mesh refinement level, h, and/or increasing the polynomial degree, p. Maier and Rannacher [91] also developed an adjoint-based isotropic AMR method with separate quantification of modeling and discretization errors for elliptic diffusion problems on Cartesian finite-element meshes. Becker *et al.* [84] applied an output-based adaptive finite-element technique to viscous flow problems involving computation of aerodynamic forces. This research was followed by the work of Venditti and Darmofal [85–87] who used error estimation strategies to adapt unstructured finite-volume triangular meshes for computing quasi-1D and 2D inviscid and viscous flows over airfoils. Nemec and Aftosmis [88, 89] also used the solution of the discrete adjoint for performing mesh refinement on 3D polyhedral meshes for compressible inviscid flows using cell-based AMR.

More recently, Ceze and Fidkowski [90] studied 2D and 3D Navier-Stokes and RANS flows using adjoint methods to direct *hp*-adaptation on body-fitted, cell-based finite-element meshes (coarsening was not permitted in their study). Yano and Darmofal [92] have also used adjoint methods for performing anisotropic *h*-adaptation on simplex meshes for high-order solutions of advection-diffusion problems, arriving at a very effective approach for optimized mesh adaptation on simplexes. Copeland *et al.* [93] considered adjoint-based goal-oriented mesh adaptation on a 2D mesh containing triangular and rectangular cells for studying the nonequilibrium hypersonic flow surrounding a blunt body during re-entry. Leicht and Hartmann [94] developed an error estimation and anisotropic mesh refinement framework for accurate calculation of force coefficients for 3D laminar aerodynamic flows using a discontinuous Galerkin discretization.

In other work, Hartmann *et al.* [95] further developed this framework for turbulent flows as well. Estep *et al.* [96] applied adjoint-based adaptation techniques to fluid-solid heat transfer problems on unstructured triangular grids. Dunne [97] developed a goal-oriented approach for modelling fluid-structure interaction on *h*-adapted Cartesian finite-element meshes. Power *et al.* [98] used adjoint-based error estimation to drive the anisotropic mesh adaptation on unstructured tetrahedral finite-element meshes to study 3D diffusion problems. More recently, Ahrabi *et al.* [99] developed *h*- and *hp*-adaptive finite-element methods for aerodynamic flows governed by the Euler and RANS equations. Output-based methods have also been used for sonic boom prediction for 3D supersonic inviscid flows governed by the Euler equations, wherein the functional is a pressure differential integrated across a predefined region in the computational domain. For example, Wintzer and Nemec [100] obtained sonic boom predictions for various aircraft configurations on Cartesian hexahedral finite-volume meshes with cut-cell boundaries and compared the obtained pressure signatures with experimental data. Similarly, Park and Darmofal [101] and Jones *et al.* [102] performed sonic boom prediction studies on 3D tetrahedral finite-volume meshes using output-based error estimation. Loseille *et al.* [103, 104] developed a 3D anisotropic goal-oriented adaptive scheme for steady Euler flows on unstructured tetrahedral finite-element meshes. Belme *et al.* [105] developed a time-accurate anisotropic goal-oriented mesh adaptation algorithm for unsteady 2D and 3D Euler flows on unstructured finite-volume meshes. Dwight [106] established a novel adjoint-based adaptive method for 2D steady Euler flows on unstructured finite-volume meshes wherein only the component of the functional error arising from the numerical diffusion present in the flux scheme was used to drive the mesh adaptation. Peter *et al.* [107] used gradients of aerodynamic forces with respect to mesh coordinates to drive the goal-oriented mesh adaptation through insertion and displacement of mesh nodes for studying 2D and 3D Euler flows. Wang and Mavriplis [108] implemented a goal-oriented *hp*-adaptive discontinuous Galerkin method for the 2D compressible Euler equations and compared its performance with pure *h*-refinement and pure *p*-refinement algorithms in terms of mesh savings obtained for a certain accuracy level. Georgoulis *et al.* [109] developed an anisotropic goal-oriented *hp*-adaptive discontinuous Galerkin method for hyperbolic-elliptic problems and made comparisons with isotropic *hp*- and *h*-anisotropic/*p*-isotropic refinement schemes.

Output-based error estimation for mesh adaptation has also been applied to reactive flows pertaining to problems such as combustion. Sun and Wheeler [110] used *a posteriori* error estimation to drive 2D isotropic cell-based *h*-adaptation on discontinuous Galerkin finite-element grids for solving the linear advection-diffusion-reaction equations. Becker *et al.* [111,112] applied *a posteriori* error estimation to mesh adaptation for the estimation of Arrhenius parameters and diffusion coefficients in steady hydrogen and ozone flames. Becker *et al.* [113] employed adjoint-based error estimation to drive patch-based AMR on structured finite-element meshes for studying low Mach number laminar flames. Braack *et al.* [114] also used adjoint methods for adaptive chemical transport modelling for studying pollutant formation on structured finite-element meshes. For the latter, in certain regions of the computational domain a more complex chemical model was used whereas in other regions a simpler model was used for accuracy reasons. Hence, the error estimates were not used for adapting the mesh but for assigning different chemical models to different regions of the computational domain. Bourlouix *et al.* [115] used a similar approach to study non-premixed flames using subgrid flamelet models. Burman *et al.* [116] used *a posteriori* error estimation to drive unstructured finite-element Delaunay mesh adaptation for low Mach, steady, laminar combustion of a methane/air Bunsen flame. Braack and Ern [117] applied these methods for the control of modeling and discretization errors on structured finite-element meshes for the study of convection-diffusion-reaction equations. Finally, Formaggia *et al.* [118] have also applied adjoint-based error estimation to drive the mesh adaptation involving refinement as well as stretching of individual elements of an unstructured finite-element mesh to study the convection-diffusion-reaction equations.

Adjoint methods have also been used to drive mesh adaptation in many other areas of research. Power *et al.* [119] developed an unsteady adjoint-based AMR strategy for studying time-varying flow problems relating to ocean modelling on unstructured tetrahedral finite-element meshes. Similarly, Davis and LeVeque [120] used adjoint methods for time-varying flows relating to tsunami modelling on adaptive block-structured finite-volume meshes. Zdunek *et al.* [121] developed an adjoint-based *hp*-adaptive finite-element approach for 3D electromagnetic scattering problems. Gomez-Revuelto *et al.* [122] also developed a goal-oriented *hp*-adaptive finite-element strategy for electromagnetic scattering and radiation problems. Pardo *et al.* [123,124] developed a 2D goal-oriented adaptive finite-element strategy for electrostatics [123] and electrodynamics [124]. Lathouwers [125] applied goal-oriented adaptivity to the S_N equations to study neutron diffusion on unstructured triangular finite-element meshes. Ju and Mahnken [126] developed an adjoint-based adaptive finite-element method for problems related to microscopic elasticity in solids. Hart *et al.* [127] developed an adaptive adjoint-based error estimation strategy for the accurate calculation of frictional forces in elastohydrodynamic lubrication problems. Joshi *et al.* [128] employed an adjoint-based adaptive finite-element method for developing a 3D optical tomography scheme. Burstedde *et al.* [129] developed a parallel, adjoint-based, 2D isotropic adaptive strategy for studying variable-viscosity Stokes flow problems on structured finite-element meshes. Mathelin and De Maître [130] implemented an adaptive *a posteriori* error estimation algorithm for stochastic finite-element methods and investigated the uncertain Burger's equation. Li and Key [131] developed a 2D adjoint-based adaptive finite-element algorithm for marine controlled-source electromagnetic modelling.

The use of solution errors to guide mesh adaptation has also been explored in the numerical study of electrically conducting flows such as those governed by the equations of MHD. For example, Ervin and Layton [132] considered *a posteriori* error estimation for finite-element methods for incompressible magnetohydrodynamic flows with a small magnetic Reynolds number. They used local error estimates of velocity, electric potential and pressure fields to adaptively refine the mesh based on the equidistribution of these errors. Zhao *et al.* [133] developed an anisotropic adaptive finite-element method using *a posteriori* error estimates of velocity and magnetic field for incompressible, high Hartmann number 2D magnetohydrodynamic flows on unstructured triangular meshes. Lankalapalli *et al.* [134] devised an *h*-adaptive finite-element method for the equations of 2D single-fluid incompressible ideal magnetohydrodynamics on unstructured triangular meshes. The latter used *a posteriori* temporal and spatial error estimates of the magnetic field to drive the mesh adaptation. Baty [135] developed an adaptive finite-element method for studying magnetic reconnection using the equations of resistive MHD on 2D unstructured triangular meshes. The Hessian matrix of the electric current density was used for anisotropic adaptation in order to obtain an equidistribution of errors in the computed current density solution. None of these researchers, however, used adjoint-based error estimates of integrated target functionals to direct the mesh adaptation for compressible MHD flows. Furthermore, hierarchical AMR methods were not considered.

It must also be noted that adjoint-based techniques have been used extensively for multidisciplinary design optimization due to their ability to estimate gradients and sensitivities of target quantities with respect to design parameters. See, for example, the work by Pironneau [136], Jameson [137], Nemec and Zingg [138], Truong *et al.* [139], and Hicken and Zingg [140]. Moreover, for conducting gaseous flows, Marta and Alonso [141,142] developed 3D discrete adjoint MHD solvers for the ideal MHD and low magnetic Reynolds number regimes with the goal of optimizing hypersonic vehicle designs while taking into account the magnetic effects of locally ionized hypersonic flow. They calculated sensitivities of lift, drag and pitching moment coefficients with respect to vehicle attitude angles, magnetic field as well as electrical conductivity. The discrete adjoint calculation was performed with the help of a methodology developed by Marta *et al.* [143] that made the use of automatic differentiation tools. Marta and Alonso [144] also used a discrete adjoint MHD approach for drag minimization and maximization of blunt bodies in a hypersonic magnetized flow. However, this adjoint approach was used only to perform sensitivity analysis and was not used for mesh adaptation purposes. This thesis seeks to address the scarcity of available literature on output-based error estimation used to direct the mesh adaptation for MHD flows.

A key advantage of the preceding output-based refinement methods is that, unlike gradient-based strategies, they can respond directly to estimates of the solution error and therefore 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 that are customized for minimizing errors and accurately evaluating the engineering functionals of particular interest. With this in mind, the present thesis seeks to combine the benefits of output-based error estimation and block-based anisotropic AMR. The proposed combined approach of parallel anisotropic block-based adaptive mesh refinement coupled with *a posteriori* output-based error estimation has already been studied by Narechania *et al.* [145] and Ngigi *et al.* [146] and compared with traditional gradient-based refinement schemes for aerodynamic flows. This combination has proven to be very beneficial in terms of savings in the computational mesh size and efficient reduction of functional error. The main focus of this thesis then pertains to further application of these ideas to the solution of flow cases governed by the ideal MHD equations.

1.3 Thesis Objectives

The primary objective of this thesis is the development of a parallel output-based mesh refinement algorithm that will lead to significant reductions in the size of the computational mesh required to achieve accurate predictions of engineering functionals of interest for steady two and three-dimensional ideal MHD flows. Applications of the proposed method will focus on problems of relevance to space plasmas and space weather simulation.

The thesis focuses only on output-based AMR for steady MHD simulations and unsteady flows are deemed beyond the scope of the study. Converged steady state solutions will be obtained on consecutive meshes using the parallel, implicit, inexact Newton solver developed by Northrup and Groth [57,60,61]. The proposed scheme will extend the parallel, body-fitted, anisotropic block-based AMR scheme previously developed by Freret and Groth [1] to employ criteria based on *a posteriori* error estimates of engineering functionals to direct the mesh adaptation. The output-based error estimation scheme and the anisotropic block-based AMR scheme will be combined to form a highly scalable parallel framework with the ability to compute in parallel functional error estimates on three-dimensional adapted meshes of varying levels of refinement, and also use them for driving the mesh adaptation. The proposed scheme will first be tested on steady aerodynamic flow cases governed by the Euler and Navier-Stokes equations. Finally, steady flow cases governed by the equations of ideal MHD will be considered to demonstrate the performance of the proposed algorithm. The performance of the algorithm will be assessed based on the reduction in the size of the computational mesh achieved by the output-based refinement method for a certain level of accuracy in the value of the functional. The error in the functional is given by the absolute value of the difference between, $F_{\text{numerical}}$, a numerically calculated value of the functional using a converged steady state solution on a given computational mesh, and, $F_{\text{reference}}$, a reference value of the functional. The value, $F_{\text{reference}}$, is calculated using analytical means for cases where an analytical solution is available. For cases where analytical solutions are not possible to obtain and functional values from literature are unavailable, $F_{\text{reference}}$ is evaluated using steady state solutions obtained on very fine uniformly refined meshes with a refinement level higher than the highest level of refinement achieved using adaptive methods. For cases where a fine mesh calculation is infeasible due to limitations in the available computational resources, the functional error is simply estimated using *a posteriori* adjoint-based error estimation. A data-driven solar wind-MHD model is also developed using photospheric magnetic field maps obtained from solar magnetograms as inputs to the MHD simulation, and is used as a test case for the proposed output-based anisotropic mesh refinement scheme.

The focus of this thesis is the development and testing of the proposed output-based AMR algorithm and not the study of physical flow features or the comparison between various flow models in terms of their abilities to reproduce particular characteristics of the flow field. As such, most of the cases investigated here are relatively simple standardized test cases that are easy to find in previous literature. These cases, however, still require significant computational time and resources for obtaining accurate solutions using traditional mesh refinement techniques, thus providing ample opportunities for demonstrating the computational performance improvements offered by the proposed output-based, anisotropic AMR solution strategy in terms of mesh size.

1.4 Thesis Organization

The thesis is organized as follows. The introduction and motivation has been provided above here in Chapter 1. This is followed by descriptions of the governing equations, the finite-volume scheme and the parallel block-based adaptive mesh refinement scheme that are provided in Chapter 2. Chapter 3 focuses on the *a posteriori* output-based error estimation scheme applied herein for the calculation of functional error estimates that are used to drive the mesh refinement. The performance of the proposed output-based mesh refinement scheme for electrically neutral or non-conducting aerodynamic flows is demonstrated in Chapter 4. The details of the solar wind-MHD model later employed as a final test case for the proposed output-based mesh refinement scheme are described in Chapter 5. Chapter 6 discusses the performance of the proposed output-based mesh refinement scheme for electrically conducting plasma flows governed by the equations of ideal MHD. Finally, Chapter 7 concludes the thesis with a summary of major research contributions and possible avenues for future study.

Chapter 2

Parallel Adaptive Mesh Refinement Finite-Volume Scheme

This chapter describes the details of the proposed parallel adaptive mesh refinement (AMR) finite-volume method used to solve numerically the set of governing equations. Section 2.1 describes the finite-volume scheme comprising the conservative form of the various systems of governing equations considered here, the finite-volume method, the semi-discrete formulation and the evaluation of fluxes and source terms. The inexact Newton's method adopted to solve the semi-discrete form of the governing equations is described in Section 2.2 for steady flows and Section 2.3 for unsteady flows. Section 2.4 describes the second-order explicit time-stepping scheme used for unsteady flows. Section 2.5 describes the parallel anisotropic block-based AMR technique comprising the refinement procedure, solution block connectivity, information exchange using ghost cells and parallel implementation of the AMR scheme.

2.1 Finite-Volume Method

2.1.1 Conservative Form of Governing Equations

The three systems of governing equations considered in this thesis are the Euler and Navier-Stokes equations governing inviscid and viscous flows of a compressible gas and the ideal MHD equations governing the behaviour of a perfectly conducting compressible plasma. A summary of these equation sets now follows.

Euler Equations

The Euler equations describe the behaviour of an inviscid, compressible gas. An inviscid fluid is assumed to have zero viscosity and zero thermal conductivity. Hence, processes such as diffusion, heat transfer and viscous dissipation are neglected. Energy dissipation only occurs through phenomena such as shock waves. Although various aspects of the underlying physics of the fluid flow are neglected, the Euler equations provide an accurate description of many relevant aerodynamic flows.

The Euler equations in conservative form are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.1}$$

$$\frac{\partial \left(\rho \mathbf{u}\right)}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} + p\mathbf{I}) = \mathbf{0}, \tag{2.2}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot ((e + p)\mathbf{u}) = 0. \tag{2.3}$$

Equations (2.1), (2.2) and (2.3) are the differential forms of the equations representing the conservation of mass, momentum and energy for the gas, respectively. Here, ρ, $\mathbf{u}$ and e are the gas density, velocity and specific total energy, respectively. The divergence operator in the equations above is associated with the physical coordinate vector, $\mathbf{x}$. Also, t is the time and p is the gas thermal pressure. Assuming the gas is calorically perfect, the specific total energy e is given by

$$e = \frac{\rho |\mathbf{u}|^2}{2} + \frac{p}{\gamma - 1}, \tag{2.4}$$

where $\gamma = C_p/C_v$ is the ratio of specific heats, which is taken to be constant. The terms C_p and C_v are the specific heat capacities at constant pressure and volume, respectively. Equations (2.1)–(2.3) can be expressed in a convenient vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{0}, \tag{2.5}$$

where the vector of conserved variables, $\mathbf{U}$, and the flux tensor, $\mathbf{F}$, are then given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ e \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho \mathbf{u} \\ \rho \mathbf{u}\mathbf{u} + p\mathbf{I} \\ (e + p)\mathbf{u} \end{bmatrix}. \tag{2.6}$$

Navier-Stokes Equations

The Navier-Stokes equations account for aspects of real fluids that are neglected by the description offered by the Euler equations. They account for diffusive transport phenomena such as viscous stresses and heat conduction, thus providing a more complete description of real gases and fluids in general. For aerodynamic flows where the viscous forces become important, solution of the Navier-Stokes equations is generally required. The Reynolds number is an important parameter that characterizes viscous flow regimes and is defined as the ratio of inertial to viscous forces. It is given by

$$Re = \frac{\rho_0 V_0 D}{\mu}, \tag{2.7}$$

where ρ_0, V_0, D and μ are the characteristic density, characteristic velocity, characteristic length scale and dynamic viscosity of the fluid, respectively. For the laminar viscous aerodynamic flows considered in this thesis, these characteristic values are taken to be free-stream values and the characteristic length is taken to be the airfoil chord length.

For laminar flows, the Navier-Stokes equations in conservative form can be expressed as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.8}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} + p\mathbf{I} - \boldsymbol{\tau}) = \mathbf{0}, \tag{2.9}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot ((e+p)\mathbf{u} - \mathbf{u}\cdot\boldsymbol{\tau} - \mathbf{q}) = 0. \tag{2.10}$$

Again, Equations (2.8), (2.9) and (2.10) are the differential forms of the laws reflecting the conservation of mass, momentum and energy, respectively. Here, ρ, $\mathbf{u}$ and e are again the gas density, velocity and specific total energy, respectively. The heat flux vector, $\mathbf{q}$, and viscous stress tensor, $\boldsymbol{\tau}$, are given by

$$\mathbf{q} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix}. \tag{2.11}$$

Here, 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 components of the viscous stress tensor. The former can be expressed in terms of the gradient of the temperature, T, and the latter can be expressed in terms of the strain rates of the gas [147, 148]. Also, t is again the time and p is the gas thermal pressure. As for the

Euler equations assuming a calorically perfect gas, the specific total energy e is given by

$$e = \frac{\rho|\mathbf{u}|^2}{2} + \frac{p}{\gamma - 1}, \tag{2.12}$$

where $\gamma = C_p/C_v$ is again the ratio of specific heats. Equations (2.8)–(2.10) can be expressed in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F}_{\rm I} - \mathbf{F}_{\rm V}) = \mathbf{0}, \tag{2.13}$$

where the vector of conserved variables $\mathbf{U}$, the inviscid flux tensor $\mathbf{F}_{\rm I}$ and the viscous flux tensor $\mathbf{F}_{\rm V}$ are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ e \end{bmatrix}, \ \mathbf{F}_{\rm I} = \begin{bmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} + p\mathbf{I} \\ (e+p)\mathbf{u} \end{bmatrix}, \ \mathbf{F}_{\rm V} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \\ \mathbf{u}\cdot\boldsymbol{\tau} - \mathbf{q} \end{bmatrix}. \tag{2.14}$$

Ideal MHD Equations

The ideal MHD equations describe the behaviour of a perfectly electrically conducting, fully ionized, quasi-neutral, inviscid, ideal gas. In the ideal MHD approximation, the various species in the plasma are treated as a single fluid of thermally perfect gases. It is assumed that enough collisions occur so as to make the pressure isotropic. The plasma is treated as quasi-neutral. If we consider any small volume of the plasma, the net charge density is zero, i.e., the total amount of negative charge is equal to the total amount of positive charge. However, in spite of quasi-neutrality of the plasma, electric currents are possible and the opposite charges do not necessarily move together. The plasma is treated as a perfect conductor with zero resistivity, thus, approaching the limit of an infinitely high magnetic Reynolds number. Therefore, the study of micro-scale phenomena such as reconnection is beyond the scope of the present study. Due to the assumption of zero resistivity, the magnetic field is assumed to be frozen in the fluid. All velocities are assumed to be much smaller than the speed of light and relativistic effects are ignored. In spite of all these what may seem rather limiting assumptions in some cases, the ideal MHD model can provide a good approximation to the behaviour of a large variety of space plasmas.

The equations of ideal MHD in conservative form, for a rotating frame, are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0, \tag{2.15}$$

$$\frac{\partial\left(\rho\mathbf{u}\right)}{\partial t}+\nabla\cdot\left(\rho\mathbf{u}\mathbf{u}-\mathbf{B}\mathbf{B}+p_T\mathbf{I}\right)=-\rho\left[\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{x}\right)\right]-2\rho\left(\boldsymbol{\Omega}\times\mathbf{u}\right), \tag{2.16}$$

$$\frac{\partial e}{\partial t}+\nabla\cdot\left((e+p_T)\mathbf{u}-(\mathbf{u}\cdot\mathbf{B})\mathbf{B}\right)=-\rho\mathbf{u}\cdot\left[\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{x}\right)\right], \tag{2.17}$$

$$\frac{\partial\mathbf{B}}{\partial t}+\nabla\cdot\left(\mathbf{B}\mathbf{u}-\mathbf{u}\mathbf{B}\right)=0. \tag{2.18}$$

Here, Equations (2.15), (2.16), (2.17) and (2.18) are the differential forms of the equations representing conservation of mass, momentum, energy and magnetic flux for the plasma, respectively. Equation (2.18) describes the time-evolution of the magnetic field given by Faraday's law expressed in divergence form. Equations (2.15)–(2.18) are supplemented by the solenoidality condition for the magnetic field, which is given by

$$\nabla\cdot\mathbf{B}=0. \tag{2.19}$$

Here, ρ, $\mathbf{u}$, e and $\mathbf{B}$ are the plasma density, velocity, specific total energy and magnetic field, respectively. For calculations performed in a non-inertial rotating frame, $\boldsymbol{\Omega}$ is the angular velocity of the reference frame. Again, the time is denoted by t and $\mathbf{x}$ is the position vector. The total pressure, p_T, is given by

$$p_T=p+\frac{|\mathbf{B}|^2}{2}, \tag{2.20}$$

where p is the thermal pressure of the plasma. Again assuming a calorically perfect gas, the specific total plasma energy, e, is given by

$$e=\frac{\rho|\mathbf{u}|^2}{2}+\frac{p}{\gamma-1}+\frac{|\mathbf{B}|^2}{2}, \tag{2.21}$$

where $\gamma=C_p/C_v$ is the ratio of specific heats. The units adopted for the ideal MHD system above are such that the magnetic permeability of vacuum, μ_0, is taken to be equal to unity, as is done in many other previous studies [67,68,71,78,149,150].

As for the Euler and Navier-Stokes equation sets, Equations (2.15)–(2.18) can be expressed in vector form as

$$\frac{\partial\mathbf{U}}{\partial t}+\nabla\cdot\mathbf{F}=\mathbf{Q}, \tag{2.22}$$

where the vector of conserved variables $\mathbf{U}$ and the flux tensor $\mathbf{F}$ are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ e \\ \mathbf{B} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} - \mathbf{B}\mathbf{B} + p_T\mathbf{I} \\ (e + p_T)\mathbf{u} - (\mathbf{u}\cdot\mathbf{B})\mathbf{B} \\ \mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B} \end{bmatrix}. \tag{2.23}$$

The source term for rotational effects $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{bmatrix} 0 \\ -\rho\left[\boldsymbol{\Omega}\times(\boldsymbol{\Omega}\times\mathbf{x})\right] - 2\rho\left(\boldsymbol{\Omega}\times\mathbf{u}\right) \\ -\rho\mathbf{u}\cdot\left[\boldsymbol{\Omega}\times(\boldsymbol{\Omega}\times\mathbf{x})\right] \\ \mathbf{0} \end{bmatrix}. \tag{2.24}$$

Here, $\boldsymbol{\Omega}\times(\boldsymbol{\Omega}\times\mathbf{x})$ is the centrifugal force and $2\,\boldsymbol{\Omega}\times\mathbf{u}$ is the Coriolis force.

2.1.2 Treatment of the Divergence Constraint for Ideal MHD

For 1D MHD problems, the solenoidality condition given in Equation (2.19) simply reduces to the constraint B_x = constant, where the $x-$direction is normal to the cell interface. Hence, in this case, no evolution equation is required for the magnetic field component in the direction perpendicular to the cell interface, i.e., B_x. It is relatively straightforward to develop a numerical solver for 1D problems in MHD. However, constructing a numerical method that is capable of solving 2D or 3D ideal MHD problems is unfortunately not as straightforward as simply applying 1D Riemann solvers in a direction normal to the face as is the case for Euler and Navier-Stokes solvers. For 2D and 3D MHD problems, the finite-volume discretized form of the divergence constraint given in Equation (2.19) becomes

$$\sum_{f=1}^{N_f} \left(\mathbf{B}_{\text{face}}\cdot\mathbf{n}_{\text{face}}\Delta A_{\text{face}}\right) = 0, \tag{2.25}$$

and must be satisfied where $\mathbf{B}_{\text{face}}$, $\mathbf{n}_{\text{face}}$ and ΔA_{face} denote the magnetic field vector, outward pointing unit normal vector and the area of a cell face, respectively, and N_f represents the number of faces for a generic hexahedral cell. Hence, a variation in the component of the magnetic field normal to the cell interface is allowed and must be balanced by the variations in the other components of the magnetic field. In general, a separate procedure is required in order to maintain this constraint on the discrete MHD solution and various divergence control techniques have been developed to tackle this issue.

One technique for magnetic field divergence control is the constrained transport (CT) approach by Evans and Hawley [151], wherein a particular finite-difference discretization of the divergence $\nabla \cdot \mathbf{B}$, is maintained to be equal to zero to machine accuracy. Such an approach usually requires a staggered mesh wherein the magnetic field variable is stored at the faces of the cells instead of at the cell centroids. More recently, variants of the constrained transport scheme that do not require a staggered magnetic field representation have also been developed by Dai and Woodward [152, 153], Ryu *et al.* [154] and Balsara and Spicer [155]. There have also been various successful attempts to develop projection schemes, wherein the Poisson equation is solved in addition to the governing equations in order to subtract the portion of the magnetic field that leads to a non-zero divergence. One such scheme is the "Hodge projection" scheme [156, 157]. Divergence cleaning schemes, such as generalized Lagrange multiplier (GLM) scheme wherein the divergence is convected out of the computational domain through the introduction of a new transport variable [158], have also been proposed. Comparisons between various techniques can be found in the work by Teunissen and Keppens [159] and the work by Tóth [160]. Divergence cleaning schemes such as GLM are also available in the framework used for this study, but have not been considered here. For this study, the approach originally proposed by Powell [161] has been used exclusively, wherein the divergence form of Faraday's law is used to modify the system of Equations (2.15)–(2.18) to yield a new system to be numerically solved. This approach is described next.

Powell's Approach

The governing partial-differential equations for ideal MHD given previously as Equations (2.15)–(2.18) can be re-expressed in weak conservation form as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.26}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} - \mathbf{B}\mathbf{B} + p_T \mathbf{I}) = -\rho \left[\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right] - 2\rho \left(\boldsymbol{\Omega} \times \mathbf{u}\right) - \left(\nabla \cdot \mathbf{B}\right) \mathbf{B}, \tag{2.27}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot ((e + p_T)\mathbf{u} - (\mathbf{u} \cdot \mathbf{B})\mathbf{B}) = -\rho \mathbf{u} \cdot \left[\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right] - \left(\nabla \cdot \mathbf{B}\right) \mathbf{u} \cdot \mathbf{B}, \tag{2.28}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B}) = -\left(\nabla \cdot \mathbf{B}\right)\mathbf{u}. \tag{2.29}$$

This new set of Equations (2.26)–(2.29) can be derived in a similar manner to that used for Equations (2.15)–(2.18), i.e., using the equations of hydrodynamics, Ohm's law, the Lorentz force and Maxwell's laws of electromagnetism. However, the $\nabla\cdot\mathbf{B} = 0$ constraint is not enforced. The terms on the right hand side, that are proportional to $\nabla \cdot \mathbf{B}$ are terms arising from expressing Faraday's law in divergence form. These terms are retained in the new set of equations as the $\nabla \cdot \mathbf{B} = 0$ constraint is not applied. This modified set of equations can be expressed in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{Q} + \mathbf{S}, \tag{2.30}$$

where the vector of conserved variables, $\mathbf{U}$, and the flux tensor, $\mathbf{F}$, are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ e \\ \mathbf{B} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} - \mathbf{B}\mathbf{B} + p_T\mathbf{I} \\ (e + p_T)\mathbf{u} - (\mathbf{u}\cdot\mathbf{B})\mathbf{B} \\ \mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B} \end{bmatrix}. \tag{2.31}$$

The source term for rotational effects, $\mathbf{Q}$, is the same as that in Equation (2.24). The new column vector contains terms arising from the new formulation of the equations and contains terms that cannot be expressed in divergence form. This vector is given by

$$\mathbf{S} = -\begin{bmatrix} 0 \\ \mathbf{B} \\ \mathbf{u}\cdot\mathbf{B} \\ \mathbf{u} \end{bmatrix} \nabla\cdot\mathbf{B} = \mathbf{S}_1 \nabla\cdot\mathbf{B}. \tag{2.32}$$

Note that enforcing $\nabla \cdot \mathbf{B} = 0$ in the modified system of Equations (2.30) would simply yield the Equations (2.15)–(2.18). However, as we saw earlier, the discrete form of the divergence does not automatically become zero in a finite-volume numerical solution procedure applied to the MHD equations. It can be shown that the original MHD system of Equations (2.15)–(2.18) with $\nabla \cdot \mathbf{B} = 0$ is not symmetrizable and also not Galilean-invariant [162–164]. The original system of equations also possesses a degenerate eigensystem, i.e., has only seven identifiable characteristic fields. The system of Equations (2.26)–(2.29), on the other hand, has a non-degenerate eigenstructure and is symmetrizable as well as Galilean-invariant [162–164]. Powell [161] first advocated solution of this system containing the column vector **S**, which permits the construction of a finite-volume Riemann solver, that satisfies the discrete divergence constraint shown in Equation (2.25) to the order of the truncation error of the solver. The key advantages of Powell's approach are the symmetrizability and Galilean-invariance of this system of equations and an eigenstructure that is not degenerate. The evaluation of the discrete divergence source term within the proposed finite-volume method is described in Section 2.1.7 below.

It should be noted however that Tóth [160] has shown that for problems containing strong shocks, Powell's approach can in some cases produce incorrect Rankine Hugoniot jump conditions, which can also generate incorrect results away from the discontinuity. Furthermore, such errors have been shown to not decrease with grid resolution. This problem arises due to the non-conservative nature of the source terms in Powell's 8-wave Riemann solver.

2.1.3 Finite-Volume Method

In a finite-volume approach [165], the computational domain is first discretized into many small contiguous control volumes. The governing partial-differential equations are integrated over each of these volumes to enforce local and global conservation of conserved flow quantities. In the case of ideal MHD, these quantities are mass, momentum, energy and magnetic induction. Similar sets of equations are treated for non-conducting gases governed by the Euler and Navier-Stokes equations also described above. Integrating the system of Equations (2.5), (2.13) or (2.30) over a three-dimensional control volume in physical space, and applying the divergence theorem to the conservation form, one arrives at the integral form of the governing equations that can be written as

$$\frac{\mathrm{d}}{\mathrm{dt}}\int_V \mathbf{U}\ \mathrm{dV} + \oint_\Omega \mathbf{F}\cdot\mathbf{n}\ \mathrm{d}\Omega = \int_\mathrm{V} \mathbf{S}\ \mathrm{dV} + \int_\mathrm{V} \mathbf{Q}\ \mathrm{dV}. \tag{2.33}$$

where V is the control volume, Ω is the closed surface of the control volume and $\mathbf{n}$ is the outward unit normal to the closed surface. The flux dyad, $\mathbf{F}$, represents the flux of solution quantities through the boundaries of the control volume. For a time-invariant three-dimensional control volume as shown in Figure 2.1, the cell-averaged solution and the Powell and rotational source term vectors can be defined as

$$\bar{\mathbf{U}} = \frac{1}{V}\int_V \mathbf{U}\ \mathrm{dV}, \tag{2.34}$$

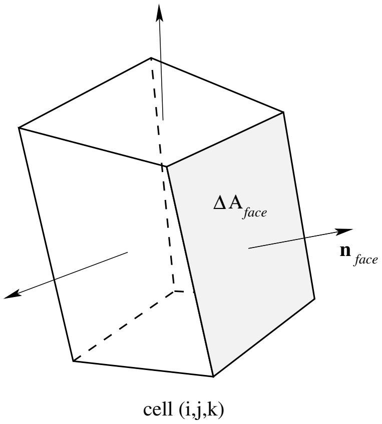


Figure 2.1: Hexahedral cell at grid location i, j, k showing face normals.

$$\bar{\mathbf{S}} = \frac{1}{V} \int_V \mathbf{S} \; \mathrm{dV}, \tag{2.35}$$

$$\bar{\mathbf{Q}} = \frac{1}{V} \int_V \mathbf{Q} \; \mathrm{dV}, \tag{2.36}$$

where V refers to the control volume of interest.

2.1.4 Semi-Discrete Form of Governing Equations

If one considers the hexahedral computational cell, (i, j, k), shown in Figure 2.1, and substitutes the definitions given in Equations (2.34), (2.35) and (2.36) in the integral form of the governing equations shown in Equation (2.33), one arrives at the semi-discrete form of the governing equations given by

$$\begin{aligned} \frac{\mathrm{d}\bar{\mathbf{U}}_{i,j,k}}{\mathrm{dt}} = -\mathbf{R}_{i,j,k}(\bar{\mathbf{U}}) = &-\frac{1}{V_{i,j,k}} \sum_{f=1}^{N_{f_{i,j,k}}} \left(\mathbf{F}_{\mathrm{face}} \cdot \mathbf{n}_{\mathrm{face}} \Delta A_{\mathrm{face}}\right)_{i,j,k} \\ &+ \frac{\bar{\mathbf{S}}_{1,i,j,k}}{V_{i,j,k}} \sum_{f=1}^{N_{f_{i,j,k}}} \left(\mathbf{B}_{\mathrm{face}} \cdot \mathbf{n}_{\mathrm{face}} \Delta A_{\mathrm{face}}\right)_{i,j,k} \\ &+ \bar{\mathbf{Q}}_{i,j,k}, \end{aligned} \tag{2.37}$$

where $\bar{\mathbf{U}}_{i,j,k}$ is the cell-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 and source terms for cell (i, j, k). The variables $V_{i,j,k}$, $\mathbf{F}_{\mathrm{face}}$, $\bar{\mathbf{Q}}_{i,j,k}$, $\mathbf{B}_{\mathrm{face}}$, $\mathbf{n}_{\mathrm{face}}$ and ΔA_{face} denote the cell volume, flux vector, rotational effect source term, magnetic field vector, outward pointing unit normal vector and the area of the shaded cell face, respectively, and $N_{f_{i,j,k}}$ represents the number of faces for cell (i, j, k). The term, $\bar{\mathbf{S}}_{1,i,j,k}$, associated with the divergence of the magnetic field is given by

$$\bar{\mathbf{S}}_{1,i,j,k} = -[0 \;\; \bar{\mathbf{B}}_{i,j,k} \;\; (\bar{\mathbf{u}} \cdot \bar{\mathbf{B}})_{i,j,k} \;\; \bar{\mathbf{u}}_{i,j,k}]^T. \tag{2.38}$$

2.1.5 Inviscid (Hyperbolic) Flux Evaluation

The inviscid or hyperbolic fluxes at the volume or cell boundaries represent the net rate of solution flux of conserved quantities and the effect of thermal and magnetic pressure forces across cell interfaces between neighbouring cells. In the current work, these forces are evaluated using a Godunov-type upwind finite-volume method. A second-order limited piecewise linear reconstruction using a least-squares curve fit is first used for calculating primitive flow variables such as density, velocity, pressure and magnetic field. Using these reconstructed variables at either side of the cell interface, the inviscid flux is calculated by use of a Riemann-solver-based flux function, namely the HLLE approximate Riemann solver.

The Riemann Problem

Godunov's method employs the solution of a localized one-dimensional Riemann problem for calculating the hyperbolic fluxes at cell boundaries. A Riemann problem is a one-dimensional initial value problem consisting of an interface with two different solution states on either side of the interface. This is depicted in Figure 2.2 for the ideal MHD equations, where $\mathbf{U}_L$ and $\mathbf{U}_R$ are the two states on the left and right sides of the interface, respectively. The vertical line at the center is the time axis while the solid horizontal line is the space axis. The slanted lines or rays emanating from the origin represent waves generated at time $t = 0$ as a result of the discontinuity between the two solution states. For the ideal MHD equations, a total of eight waves are generated, as can be seen in Figure 2.2. These are the contact discontinuity or entropy wave combined with the $\nabla \cdot \mathbf{B} = 0$ convective wave (C), two slow magnetosonic waves (SW), two Alfvén waves (AW), and two fast magnetosonic waves (FW) all of which are shown in Figure 2.2. For gas dynamical systems represented by the Euler and Navier-Stokes equations, the Riemann problem for a three-dimensional flow only consists of five waves, namely the contact discontinuity and shear waves and a shock or rarefaction on either side of the contact discontinuity.

In the current work, the fluxes are calculated by constructing a Riemann problem

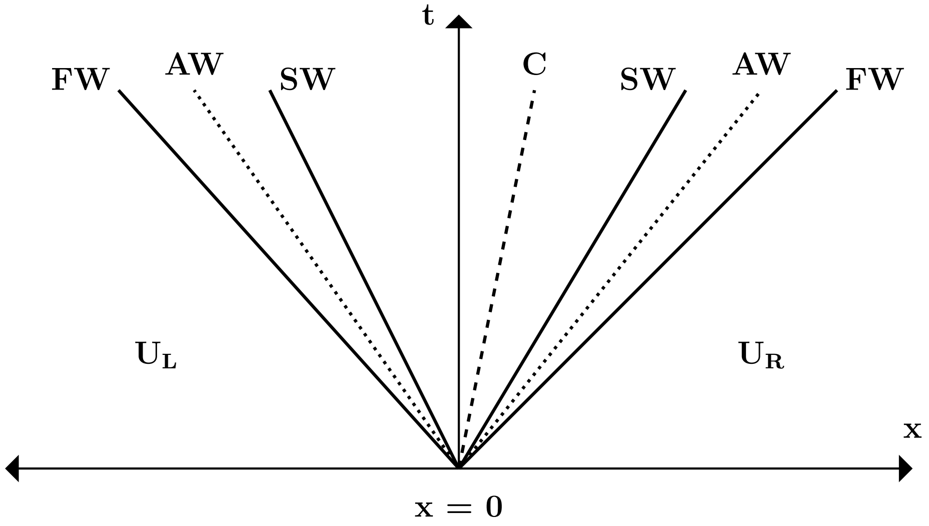


Figure 2.2: The Riemann problem for ideal MHD.

at every cell interface, with $\mathbf{U}_L$ and $\mathbf{U}_R$ being the conserved states in the neighbouring cells sharing the interface, and the normal to the cell interface being the x-axis shown in Figure 2.2. After solving the Riemann problem, the solution on the t-axis is used to calculate the flux. For a polytropic gas, the exact solution of the Riemann problem can be obtained as described by Gottlieb and Groth [166]. However, an exact solution to the Riemann problem for MHD in the general case is not available in the general case. Moreover, several computationally less expensive approximate solvers have also been developed over the years, such as those by Roe [167], Harten-Lax-van Leer-Einfeldt (HLLE) [168], Toro *et al.* [169], Linde [170], Liou [171,172] and Osher and Solomon [173]. The reader is referred to the textbook by Toro [165] for a detailed description of such solvers for gas dynamical flows. Several MHD variants of such solvers have also been developed, such as the non-linear solver of Dai and Woodward [174], linearized Roe-type solvers by Powell *et al.* [10], Zachary and Collela [175] and Brio and Wu [176], HLLC-type solvers developed by Li [177], Gurski [178] and Miyoshi and Kusano [179], and the HLLE solver by Linde [170]. For the purpose of the current work, the Harten-Lax-van Leer-Einfeldt (HLLE) [168, 170] flux function has been exclusively used for hyperbolic flux evaluation. The HLLE flux function will be described in a later section to follow. The second-order reconstruction scheme used to calculate $\mathbf{U}_L$ and $\mathbf{U}_R$ at either side of the cell interface is now described, followed by a description of the HLLE flux.

Limited Piecewise Linear Least-Squares Reconstruction

Solving the Riemann problem using the cell-averaged values, $\mathbf{U}_L$ and $\mathbf{U}_R$, as described in the previous section produces first-order spatial accuracy. The accuracy and resolution of the solution can be improved by spatially reconstructing the solution inside the computational cell. For second-order accuracy, a linear reconstruction is used wherein the solution inside the cell is represented by a linear function. The reconstructed values obtained at the cell interfaces are then used to solve the Riemann problem for better accuracy and resolution. Higher-order reconstructions, such as the CENO method [72], have also been developed by various researchers, but are not considered in the current work.

Second- and higher-order spatial reconstructions can often lead to solutions showing non-monotonic oscillatory behaviour in regions containing discontinuities in the flow variables. To ensure monotonicity in the solution near discontinuities, while at the same time maintaining the correct order of accuracy in smooth regions, spatial reconstruction techniques are implemented in combination with slope limiters that ensure monotonicity in reconstruction. This ensures that the reconstructed values are bounded by the maximum and minimum values in the cell under consideration and its neighbours. It also ensures that the solution is physically realizable and takes on meaningful values devoid of any unwanted oscillations. Thus, for a scalar component, U, of the solution vector, $\mathbf{U}$, the limited second-order piecewise linear reconstructed solution in the computational cell is given by

$$U_{i,j,k}(x,y,z) = \bar{U}_{i,j,k} + \phi_{i,j,k}\left[\left.\frac{\partial U}{\partial x}\right|_{ijk}(x - x_{i,j,k}) + \left.\frac{\partial U}{\partial y}\right|_{ijk}(y - y_{i,j,k}) + \left.\frac{\partial U}{\partial z}\right|_{ijk}(z - z_{i,j,k})\right], \tag{2.39}$$

where $\bar{U}_{i,j,k}$ is the cell-averaged value of the solution variable U in the cell (i,j,k) and $\partial U/\partial x|_{ijk}$, $\partial U/\partial y|_{ijk}$ and $\partial U/\partial z|_{ijk}$ are gradients of this solution variable in the x-, y- and z-directions, respectively. The vector, $\mathbf{x}_{i,j,k} = (x_{i,j,k}, y_{i,j,k}, z_{i,j,k})$, is the location of the centroid of the cell and $\phi_{i,j,k}$ is the slope limiter.

The solution gradients, $\partial U/\partial x|_{ijk}$, $\partial U/\partial y|_{ijk}$ and $\partial U/\partial z|_{ijk}$, are calculated by applying a least-squares approach that entails the minimization of the error defined by

$$\epsilon^2 = \sum_{n=1}^{N} \epsilon_n^2 = \sum_{n=1}^{N} \left[\Delta\bar{U}_n - \nabla U_{i,j,k}\cdot(\mathbf{x}_n - \mathbf{x}_{i,j,k})\right], \tag{2.40}$$

where $\Delta \bar{U}_n = \bar{U}_n - \bar{U}_{i,j,k}$ for $\bar{U}_n$ being the cell-averaged value of the solution variable in the neighbouring cell n, $\nabla U_{i,j,k}$ is the gradient vector $(\partial U/\partial x|_{ijk}, \partial U/\partial y|_{ijk}, \partial U/\partial z|_{ijk})$ and $\mathbf{x}_n$ is the location of the centroid of the neighbouring cell n. Differentiating Equation (2.40) with respect to the gradients $\partial U/\partial x|_{ijk}$, $\partial U/\partial y|_{ijk}$ and $\partial U/\partial z|_{ijk}$ to minimize the error, ϵ, results in a linear system given by

$$\begin{bmatrix} \overline{(\Delta x)^2_{i,j,k}} & \overline{(\Delta x \Delta y)_{i,j,k}} & \overline{(\Delta x \Delta z)_{i,j,k}} \\ \overline{(\Delta x \Delta y)_{i,j,k}} & \overline{(\Delta y)^2_{i,j,k}} & \overline{(\Delta y \Delta z)_{i,j,k}} \\ \overline{(\Delta x \Delta z)_{i,j,k}} & \overline{(\Delta y \Delta z)_{i,j,k}} & \overline{(\Delta z)^2_{i,j,k}} \end{bmatrix} \begin{bmatrix} \partial U/\partial x|_{ijk} \\ \partial U/\partial y|_{ijk} \\ \partial U/\partial z|_{ijk} \end{bmatrix} = \begin{bmatrix} \overline{(\Delta U \Delta x)_{i,j,k}} \\ \overline{(\Delta U \Delta y)_{i,j,k}} \\ \overline{(\Delta U \Delta z)_{i,j,k}} \end{bmatrix}, \tag{2.41}$$

where

$$\begin{aligned} \overline{(\Delta x)^2_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}(x_n - x_{i,j,k})^2, \\ \overline{(\Delta y)^2_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}(y_n - y_{i,j,k})^2, \\ \overline{(\Delta z)^2_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}(z_n - z_{i,j,k})^2, \\ \overline{(\Delta x \Delta y)_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}(x_n - x_{i,j,k})(y_n - y_{i,j,k}), \\ \overline{(\Delta x \Delta z)_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}(x_n - x_{i,j,k})(z_n - z_{i,j,k}), \\ \overline{(\Delta y \Delta z)_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}(y_n - y_{i,j,k})(z_n - z_{i,j,k}), \\ \overline{(\Delta U \Delta x)_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}\Delta \bar{U}_n(x_n - x_{i,j,k}), \\ \overline{(\Delta U \Delta y)_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}\Delta \bar{U}_n(y_n - y_{i,j,k}), \\ \overline{(\Delta U \Delta z)_{i,j,k}} &= \frac{1}{N}\sum_{n=1}^{N}\Delta \bar{U}_n(z_n - z_{i,j,k}). \end{aligned} \tag{2.42}$$

The linear system in Equation (2.41) can be easily solved using Cramer's rule.

The limiter proposed by Venkatakrishnan [180] has been used in the current work to ensure solution monotonicity. For a given scalar solution variable, U, this limiter is given by

$$\phi_{i,j,k} = \begin{cases} \phi\left(\frac{U_{\max}-\bar{U}_{i,j,k}}{U_k-\bar{U}_{i,j,k}}\right) \text{ for } U_k - \bar{U}_{i,j,k} < 0 \\ \phi\left(\frac{U_{\min}-\bar{U}_{i,j,k}}{U_k-\bar{U}_{i,j,k}}\right) \text{ for } U_k - \bar{U}_{i,j,k} > 0 \\ 1 \qquad\qquad\qquad \text{otherwise} \end{cases}, \tag{2.43}$$

where $\phi(y)$ is given by

$$\phi(y) = \frac{y^2+2y}{y^2+y+2}. \tag{2.44}$$

Here, $U_{\max} = \max(\bar{U}_{i,j,k}, \bar{U}_{\text{neighbours}})$, $U_{\min} = \min(\bar{U}_{i,j,k}, \bar{U}_{\text{neighbours}})$, U_k is the unlimited reconstructed value at the kth reconstructed flux quadrature point and $\bar{U}_{i,j,k}$ and $\bar{U}_{\text{neighbours}}$ are the cell-averaged values of U for the cell (i, j, k) and its neighbours, respectively.

The HLLE Flux Function

The HLLE flux function calculation is now described for the system of ideal MHD equations. For the Euler and Navier-Stokes equations, this procedure is further simplified by the fact that the magnetic field is zero for these systems. Hence, for the Euler and Navier-Stokes equations, the fast magnetosonic speed, c_f, shown below in Equation (2.52) simplifies to the speed of sound, $a = \sqrt{\gamma p/\rho}$, for a zero magnetic field. The remaining steps are the same for the calculation of the flux. The left and right states obtained at the cell face using the reconstruction described in the previous section are rotated to align with the coordinate system wherein the x-axis points along the unit normal of the face. A localized Riemann problem is set up where the two initial states are prescribed as

$$\mathbf{U}(x, t=0) = \begin{cases} \mathbf{U}_L \text{ for } x > 0 \\ \mathbf{U}_R \text{ for } x < 0 \end{cases}, \tag{2.45}$$

with $\mathbf{U}_L$ and $\mathbf{U}_R$ being the left and right states, respectively. The HLLE solver assumes a self-similar solution $\mathbf{U}_*$ for all time $t > 0$, such that

$$\mathbf{U}\left(\frac{x}{t}=0\right) = \begin{cases} \mathbf{U}_L \text{ for } \frac{x}{t} \leq \lambda^- \\ \mathbf{U}_* \text{ for } \lambda^- < \frac{x}{t} < \lambda^+ \\ \mathbf{U}_R \text{ for } \lambda^+ \leq \frac{x}{t} \end{cases}, \tag{2.46}$$

where λ^- and λ^+ are the speeds of the leftward- and rightward-moving characteristic waves. This is depicted in Figure 2.3, where the intermediate state, $\mathbf{U}_*$, occupies the

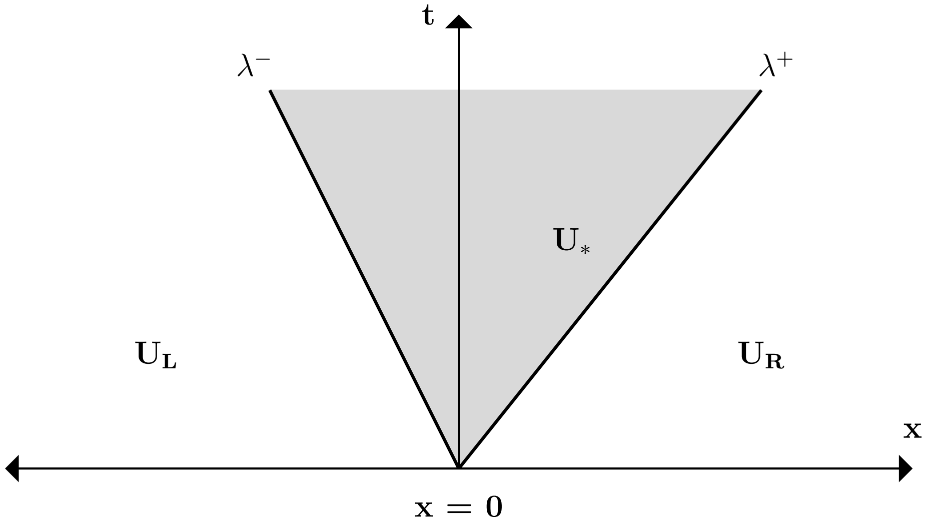


Figure 2.3: The HLLE solver.

shaded region. Correspondingly, the inviscid flux vector is given by

$$\mathbf{F}\left(\mathbf{U}_L, \mathbf{U}_R, \frac{x}{t} = 0\right) = \begin{cases} \mathbf{F}_L \text{ for } \frac{x}{t} \leq \lambda^- \\ \mathbf{F}_* \text{ for } \lambda^- < \frac{x}{t} < \lambda^+ \\ \mathbf{F}_R \text{ for } \lambda^+ \leq \frac{x}{t} \end{cases}, \tag{2.47}$$

where $\mathbf{F}_L = \mathbf{F}(\mathbf{U}_L)$, $\mathbf{F}_R = \mathbf{F}(\mathbf{U}_R)$ and $\mathbf{F}_* = \mathbf{F}(\mathbf{U}_*)$. Using jump conditions, the intermediate state, $\mathbf{U}_*$, is given by

$$\mathbf{U}_* = \frac{\lambda^+ \mathbf{U}_R - \lambda^- \mathbf{U}_L}{\lambda^+ - \lambda^-} - \frac{\mathbf{F}_R - \mathbf{F}_L}{\lambda^+ - \lambda^-}, \tag{2.48}$$

and the corresponding flux is given by

$$\mathbf{F}_* = \frac{\lambda^+ \mathbf{F}_L - \lambda^- \mathbf{F}_R}{\lambda^+ - \lambda^-} + \frac{\lambda^+ \lambda^-}{\lambda^+ - \lambda^-}(\mathbf{U}_R - \mathbf{U}_L). \tag{2.49}$$

The speeds λ^- and λ^+ are given by

$$\lambda^- = \max(\lambda_L^{\min}, \hat{\lambda}^{\min}) = \min(u_L - c_{f,L}, \hat{u} - \hat{c_f}), \tag{2.50}$$

$$\lambda^+ = \max(\lambda_R^{\max}, \hat{\lambda}^{\max}) = \max(u_R + c_{f,R}, \hat{u} + \hat{c_f}). \tag{2.51}$$

Here, u_L and u_R are the velocities along the face normal for the left and right states, respectively. The fast magnetosonic speed, c_f, is given by

$$c_f^2 = \frac{1}{2}\left[\frac{\gamma p + |\mathbf{B}|^2}{\rho} + \sqrt{\left(\frac{\gamma p + |\mathbf{B}|^2}{\rho}\right)^2 - \frac{4\gamma p B_x^2}{\rho^2}}\right]. \tag{2.52}$$

The speeds indicated using overhats are calculated using Roe-averaged quantities for density, velocity, enthalpy and magnetic field [10,161]. For the density, velocity or enthalpy variables denoted by ω, the Roe-averaged quantity, $\hat{\omega}$, is calculated by mass-weighting the left and right states as

$$\hat{\omega} = \frac{\omega_L\sqrt{\rho_L} + \omega_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}. \tag{2.53}$$

Here, ω_L represents the density, velocity or enthalpy values for the left state while ω_R represents the density, velocity or enthalpy values for the right state. The Roe-averaged density simplifies to $\hat{\rho} = \sqrt{\rho_L\rho_R}$. However, the Roe-averaged magnetic field is calculated as

$$\hat{\mathbf{B}} = \frac{\mathbf{B}_L\sqrt{\rho_R} + \mathbf{B}_R\sqrt{\rho_L}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \tag{2.54}$$

where the mass-weightings are interchanged [176,181]. These Roe-averaged quantities are then used to calculate $\hat{c_f}$. Finally, the solution on the line $(x/t = 0)$ is used to calculate the flux. As mentioned earlier, the fast magnetosonic speed, c_f, shown in Equation (2.52) simplifies to the speed of sound, $a = \sqrt{\gamma p/\rho}$, for the Euler and Navier-Stokes equations. It must be noted that the above Roe-average for ideal MHD exists only for the special case where $\gamma = 2$ [176,181]. For this special case, The Roe matrix can be expressed as a flux Jacobian of the Roe-averaged state. At this value of γ, the ideal MHD system decouples into two systems. The first system looks like the Euler equations of non-conducting gas dynamical flows and the second system is a set of advection equations describing the evolution of the transverse components of the magnetic field. In this case, the Roe-averages for density, velocity and enthalpy correspond to those for the gas dynamical Euler equations whereas the Roe-average for the magnetic field is given by Equation (2.54). For the general case where $\gamma \neq 2$, it is not possible to express the Roe matrix as a flux Jacobian of an averaged state and therefore, a general result for the Roe-averaged state is difficult to obtain [176,181].

2.1.6 Viscous (Elliptic) Flux Evaluation

The viscous or elliptic fluxes at cell boundaries represent the net rate of diffusion of various conserved quantities across interfaces between neighbouring cells. As the Euler and ideal MHD equations do not contain such terms, the numerical evaluation of these fluxes applies only to the Navier-Stokes equations. Unlike the hyperbolic fluxes, the viscous forces depend not only on the values of the solution at the cell faces but also on their gradients and can be expressed as

$$\mathcal{F}_V = \mathcal{F}_V(\mathbf{W}_{i+1/2,j,k}, \nabla\mathbf{W}_{i+1/2,j,k}), \tag{2.55}$$

where $\mathbf{W}_{i+1/2,j,k}$ is the value of the solution at the face $(i+1/2,j,k)$ shared by the cells (i,j,k) and $(i+1,j,k)$. Here, $\mathbf{W}_{i+1/2,j,k}$, is calculated by averaging the left and right reconstructed values as

$$\mathbf{W}_{i+1/2,j,k} = \frac{\mathbf{W}_L + \mathbf{W}_R}{2}. \tag{2.56}$$

In the current work, the cell face gradients are evaluated using the formula proposed by Mathur and Murthy [48, 49, 182] given by

$$\nabla\mathbf{W}|_{i+1/2,j,k} = \frac{\mathbf{W}_{i+1,j,k} - \mathbf{W}_{i,j,k}}{\Delta s}\frac{\mathbf{n}}{\mathbf{n}\cdot\mathbf{e}_s} + \left(\overline{\nabla\mathbf{W}} - \overline{\nabla\mathbf{W}}\cdot\mathbf{e}_s\frac{\mathbf{n}}{\mathbf{n}\cdot\mathbf{e}_s}\right), \tag{2.57}$$

where $\mathbf{W}_{i,j,k}$ and $\mathbf{W}_{i+1,j,k}$ are values at the centroids of the cells $(i+1,j,k)$ and (i,j,k), respectively, and $\overline{\nabla\mathbf{W}}$ is the weighted average of the cell centered gradient at the cell interface given by

$$\overline{\nabla\mathbf{W}} = \alpha\nabla\mathbf{W}_{i,j,k} + (1-\alpha)\nabla\mathbf{W}_{i+1,j,k}. \tag{2.58}$$

Here, α is a weighting factor based on cell volume ratios and is given by

$$\alpha = \frac{V_{i,j,k}}{V_{i,j,k} + V_{i+1,j,k}}, \tag{2.59}$$

$\mathbf{n}$ is the unit normal vector at the cell interface, Δs is the distance between the centroids and $\mathbf{e}_s$ is the unit normal vector between the centroids. This is depicted in the Figure 2.4.

2.1.7 Source Term Evaluation

The ideal MHD equations contain the source term, $\mathbf{Q}$, that takes into account the effects of performing the calculation in a rotating frame of reference and also the so-called "Powell source term", $\mathbf{S}$, arising from the divergence constraint on the magnetic field,

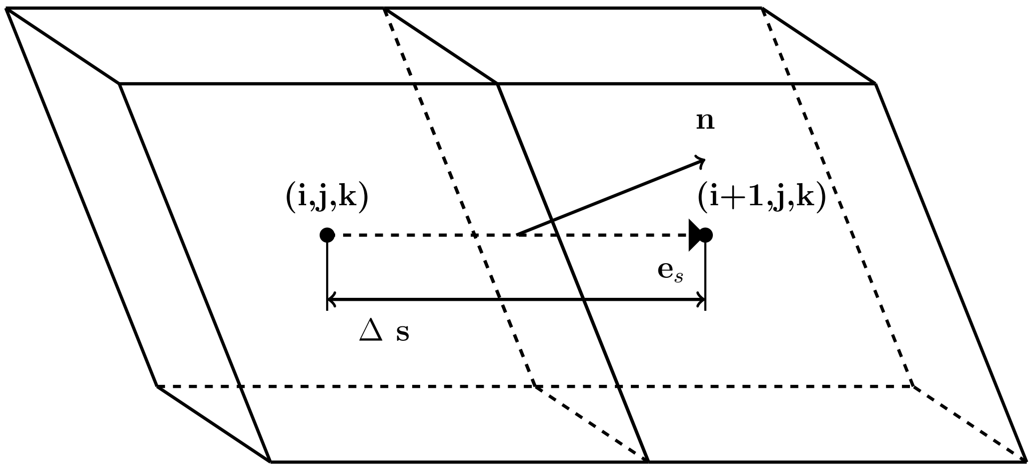


Figure 2.4: Gradient calculation for a hexahedral cell face.

as given previously in Equation (2.24). The former contains terms depending on the values of the primitive flow variables and the position vector $\mathbf{x}$. The latter contains terms depending on the values of the primitive flow variables and on the divergence of the magnetic field, $\nabla \cdot \mathbf{B}$. In the evaluation of $\mathbf{Q}$ and $\mathbf{S}$, the primal flow variables are simply taken to be the cell-averaged values of these quantities stored in the cell while $\mathbf{x}$ is taken to be the cell centroid. The term, $\nabla \cdot \mathbf{B}$, is evaluated by the application of the divergence theorem applied to the volume of the computational cell as

$$\int_V \nabla \cdot \mathbf{B} \; \mathrm{dV} = \oint_\Omega \mathbf{B} \cdot \mathbf{n} \; \mathrm{d}\Omega, \tag{2.60}$$

where V is the control volume, Ω is the closed surface of the control volume and $\mathbf{n}$ is the outward unit normal to the closed surface. Hence, the divergence of a vector field integrated over a volume equals the net flux of the field through the outer surface of the volume. The second-order discretization of the right-hand side of Equation (2.60) is given by

$$\oint_\Omega \mathbf{B} \cdot \mathbf{n} \; \mathrm{d}\Omega = \sum_{f=1}^{N_{f_{i,j,k}}} \left(\mathbf{B}_{\text{face}} \cdot \mathbf{n}_{\text{face}} \Delta A_{\text{face}}\right)_{i,j,k}, \tag{2.61}$$

where $\mathbf{B}_{\text{face}}$, $\mathbf{n}_{\text{face}}$ and ΔA_{face} denote the magnetic field vector, outward pointing unit normal vector and the area of the shaded cell face, respectively, and $N_{f_{i,j,k}}$ represents the number of faces for cell (i, j, k). Here, $\mathbf{B}_{\text{face}}$ is calculated as

$$\mathbf{B}_{\text{face}} = \frac{\mathbf{B}_L + \mathbf{B}_R}{2}, \tag{2.62}$$

where $\mathbf{B}_L$ and $\mathbf{B}_R$ are values of the magnetic field obtained at the centroid of the cell face using the reconstructed profiles from the two adjacent cells sharing the face.

2.1.8 Accuracy of Spatial Discretization Scheme

The unlimited form of the proposed finite-volume spatial discretization procedure is nominally second-order accurate with discretization errors of $O(\Delta x^2)$, where Δx is the spatial discretization step size. The accuracy of the scheme drops to first-order when the limiter is applied. The second-order spatial accuracy of this finite-volume scheme used in combination with AMR method has been previously verified by Sachdev *et al.* [183], Gao and Groth [64], Gao *et al.* [58], Ivan *et al.* [69], and McDonald *et al.* [184] for both 2D and 3D inviscid and laminar compressible flows, as well as ideal MHD plasma flows. While a limited number of similar verification studies were also performed as part of the present study which confirm the accuracy of the scheme, they are not reported herein due to concerns associated with thesis length.

2.2 Inexact Newton's Method for Steady Flows

The parallel inexact Newton's method previously developed by Northrup and Groth for multi-block hexahedral meshes with AMR [57,60,61] is used to solve the system given in Equation (2.37). The system of algebraic equations describing steady state solutions of the semi-discrete form of the governing equations given in Equation (2.37) for all cells in the computational domain is given by

$$\frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{d}t} = -\mathbf{R}(\bar{\mathbf{U}}) = 0, \tag{2.63}$$

where $\bar{\mathbf{U}}$ represents the cell-averaged conserved state in all the computational cells. This is a large coupled system of non-linear algebraic equations that can be solved using iterative techniques. The solution to Equation (2.63) is obtained using a dual-time-stepping approach, wherein a pseudo-temporal derivative with an additive Schwarz global preconditioner, $\mathbf{\Gamma}$, is introduced in Equation (2.63) as given by

$$\mathbf{\Gamma}\frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{d}\tau} = -\mathbf{R}(\bar{\mathbf{U}}), \tag{2.64}$$

where τ is the pseudo-time-step. Applying the implicit Euler time-marching method to the pseudo-time derivative yields

$$\left[-\frac{\mathbf{\Gamma}}{\Delta\tau^n} + \left(\frac{\partial \mathbf{R}}{\partial \bar{\mathbf{U}}}\right)^{(n)}\right]\Delta\bar{\mathbf{U}}^{(n)} = -\mathbf{R}(\bar{\mathbf{U}}^{(n)}), \tag{2.65}$$

where $\Delta\tau^n$ is the pseudo-time-step for the nth iteration. As $\Delta\tau^n \to \infty$, Equation (2.65) reduces to

$$\left[\frac{\partial \mathbf{R}}{\partial \bar{\mathbf{U}}}\right]^{(n)}\Delta\bar{\mathbf{U}}^{(n)} = \mathbf{J}^{(n)}\Delta\bar{\mathbf{U}}^{(n)} = -\mathbf{R}(\bar{\mathbf{U}}^{(n)}), \tag{2.66}$$

where $\mathbf{J}^{(n)} = [\partial\mathbf{R}/\partial\bar{\mathbf{U}}]^{(n)}$ is the Jacobian of the residual with respect to the conserved solution vector. Equation (2.66) can also be obtained by simply applying Newton's iterative method to Equation (2.63). An initial guess, $\bar{\mathbf{U}}^{(n=0)}$, of the solution is used to calculate the vector $\mathbf{R}(\bar{\mathbf{U}}^{(n)})$ and the matrix $\mathbf{J}^{(n)}$ and the system in Equation (2.65) is solved to obtain the vector $\Delta\bar{\mathbf{U}}^{(n)}$. An improved estimate of the solution, $\bar{\mathbf{U}}^{(n+1)}$, is then obtained as

$$\bar{\mathbf{U}}^{(n+1)} = \bar{\mathbf{U}}^{(n)} + \Delta\bar{\mathbf{U}}^{(n)}. \tag{2.67}$$

For the purpose of obtaining steady state solutions to Equation (2.63), the switched evolution/relaxation (SER) approach, as proposed by van Leer and Mulder [185], has been used here. In this approach, the pseudo-time-step, $\Delta\tau^n$, is varied, starting from a finite value, and gradually increased to very large values as the steady state solution is obtained. A time-step multiplier, ν^n, is introduced that increases as the L_2 norm of the solution residual $||\mathbf{R}||_2$ decreases, and is given by

$$\nu^n = \nu_{\min}\max\left(1, \frac{1}{||\mathbf{R}||_2}\right), \tag{2.68}$$

where $\nu_{\min}$ is the minimum or initial multiplier. The pseudo-time-step, $\Delta\tau^n$, is then calculated according to the Courant-Friedrichs-Lewy (CFL) and von Neumann stability criteria [186] for inviscid and viscous compressible non-conducting flows, respectively, as

$$\Delta\tau^n = \nu^n \min\left(\frac{\Delta x_{ref}}{|\mathbf{u}| + a}, \frac{\rho\Delta x_{ref}^2}{\mu}\right). \tag{2.69}$$

Here, $\mathbf{u}$, a, Δx_{ref}, ρ and μ are the flow velocity, speed of sound, reference grid cell size, density and dynamic viscosity of the fluid, respectively. Similarly, for ideal MHD flows, $\Delta\tau^n$ is calculated as

$$\Delta\tau^n = \nu^n \min\left(\frac{\Delta x_{ref}}{|\mathbf{u}| + c_f}\right), \tag{2.70}$$

where c_f is the fast magnetosonic speed. The Equation (2.65) is solved to obtain $\Delta\bar{\mathbf{U}}^{(n)}$ and the solution is subsequently updated using Equation (2.67). These steps are subsequently repeated within an iterative procedure until an appropriate norm of the solution residual is reduced below a predefined threshold, i.e. $||\mathbf{R}(\bar{\mathbf{U}}^{(n)})||_2 < \epsilon||\mathbf{R}(\bar{\mathbf{U}}^{(0)})||_2$. A user-defined tolerance level of $\epsilon = 10^{-8}$ has been typically used in this work. Due to issues with equation scaling, in practice, it is generally sufficient for the residual norms of just one or two of the solution variables to be monitored for convergence. As all of the flow problems considered in this thesis involve a compressible fluid and the density variations are significant, the L_2 norm of the density residual was used here and proved very effective in assessing the convergence of the Newton scheme.

Equation (2.65) is a large, sparse, non-symmetric, banded linear system of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is the residual Jacobian and $\mathbf{b}$ is the residual, and needs to be solved in every iteration, n, of Newton's method. For such systems, iterative Krylov subspace methods have proven to be quite efficient. In the present work, a class of Krylov subspace methods known as the Generalized Minimum Residual (GMRES) method, is used. This method was originally developed by Saad and Schultz [187]. A useful feature of the GMRES algorithm is that it does not require explicit calculation of the global Jacobian matrix $\mathbf{A}$, permitting the use of a "matrix-free" or "Jacobian-free" approach in which numerical differentiation based on Fréchet derivatives is used. For the GMRES method to be effective, an additive Schwarz global preconditioner is used in conjunction with block incomplete lower-upper (BILU) local preconditioning. The application of GMRES within each step of Newton's method results in a nested iterative procedure where the inner iterations determine the solution of the linear system in Equation (2.66) at each step of Newton's method while the outer iterations solve the non-linear system, $\mathbf{R}(\bar{\mathbf{U}}) = 0$, using Newton's method. Convergence of the linear system at every Newton step is not necessarily required to obtain a converged solution to the system, $\mathbf{R}(\bar{\mathbf{U}}) = 0$. Hence, for efficient performance, the GMRES is only partially converged at each step of Newton's method. Accordingly, the inner GMRES iterations are stopped when $||\mathbf{R}(\bar{\mathbf{U}}^{(n)}) + \mathbf{J}^{(n)}\Delta\bar{\mathbf{U}}^{(n)}||_2 < \zeta||\mathbf{R}(\bar{\mathbf{U}}^{(n)})||_2$, where ζ is a nominal tolerance value. A value of $\zeta \approx 0.01 - 0.05$ was used for the purpose of this work.

2.3 Implicit Time-Marching for Unsteady Flows

Unsteady time-varying flows can be described by the semi-discrete form of the governing equations given in Equation (2.37) by not equating the residual to zero, i.e., by solving

$$\frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{d}t} = -\mathbf{R}(\bar{\mathbf{U}}). \tag{2.71}$$

In order to solve this system via an implicit time-marching scheme, a dual time-stepping approach is adopted for which a pseudo-temporal derivative with an additive Schwarz global preconditioner, $\mathbf{\Gamma}$, is also introduced resulting in a modified residual, $\mathbf{R}^*(\bar{\mathbf{U}})$, given by

$$\mathbf{\Gamma}\frac{\partial\bar{\mathbf{U}}}{\partial\tau} = -\frac{\partial\bar{\mathbf{U}}}{\partial t} - \mathbf{R}(\bar{\mathbf{U}}) = -\mathbf{R}^*(\bar{\mathbf{U}}), \tag{2.72}$$

where τ is the pseudo-time-step. The modified residual, $\mathbf{R}^*(\bar{\mathbf{U}})$, includes not only the temporal derivative but also the actual residual, $\mathbf{R}(\bar{\mathbf{U}})$. The system, $\mathbf{R}^*(\bar{\mathbf{U}}) = 0$, can be solved at each physical time-step using Newton's iterative method. In order to do this, Equation (2.72) is first discretized in time using an implicit second-order backward time-marching scheme (BDF2) which gives

$$\mathbf{R}^*(\bar{\mathbf{U}}^{n+1}) = \frac{3\bar{\mathbf{U}}^{n+1} - 4\bar{\mathbf{U}}^{n} + \bar{\mathbf{U}}^{n-1}}{2\Delta t} + \mathbf{R}(\bar{\mathbf{U}}^{n+1}), \tag{2.73}$$

where Δt is the physical time-step.

The system of Equation (2.73) is a large coupled system of non-linear equations similar to those in Equation (2.63) that can again be solved using an iterative technique. Given the values of the solution, $\bar{\mathbf{U}}^{n-1}$, at time level $(n-1)$ and solution, $\bar{\mathbf{U}}^{n}$, at time level n, the solution at time level $(n+1)$, $\bar{\mathbf{U}}^{n+1}$, can be obtained by solving Equation (2.73). This system is first linearized through the application of Newton's method to yield

$$\left[\frac{2}{3\Delta t}\mathbf{I} + \frac{\partial\mathbf{R}}{\partial\bar{\mathbf{U}}}\right]^{(n+1,k)} \Delta\bar{\mathbf{U}}^{(n+1,k)} = \mathbf{J}^{*(n+1,k)}\Delta\bar{\mathbf{U}}^{(n+1,k)} = -\mathbf{R}^*(\bar{\mathbf{U}}^{(n+1,k)}). \tag{2.74}$$

Here, $\bar{\mathbf{U}}^{(n+1,k+1)} = \bar{\mathbf{U}}^{(n+1,k)} + \Delta\bar{\mathbf{U}}^{(n+1,k)}$. Thus, $\bar{\mathbf{U}}^{(n+1,k=0)}$ is an initial guess for the solution $\bar{\mathbf{U}}^{n+1}$ at time level $(n+1)$, while $\bar{\mathbf{U}}^{(n+1,k+1)}$ is an improved approximation obtained by solving Equation (2.74) at every step, k, of Newton's method. The earlier solution, $\bar{\mathbf{U}}^{n}$, at time level n is an obvious candidate for the initial guess, $\bar{\mathbf{U}}^{(n+1,k=0)}$, since it is readily available. Hence, Equation (2.73) is a non-linear system of equations to be solved iteratively (outer loop) at each physical time-step, n, while Equation (2.74) is a linearized equation obtained from Newton's inexact method which is solved iteratively (inner loop) at each iteration, k, of the outer loop. This is similar to the nested procedure from Section 2.2. Correspondingly, Equation (2.74) is the large, sparse, non-symmetric, banded linear system of the form $\mathbf{Ax} = \mathbf{b}$, solved using GMRES with an additive Schwartz global preconditioner used in conjunction with local BILU preconditioning. The linear system in Equation (2.74) is easier to solve as compared to that in Equation (2.65) on account of the temporal term, $2/3\Delta t$, which causes the linear system to be diagonally dominant.

It should be noted that although this formulation of the second-order BDF2 implicit time-marching scheme for the solution of time-variant unsteady flows is also available in the current framework used for this work, it has not been used for any of the cases presented herein.

2.4 Explicit Time-Marching for Unsteady Flows

Unsteady time-dependent solutions of the solar wind are considered in Chapter 5. These are obtained by using a standard explicit, second-order accurate, Runge-Kutta time-marching scheme [188] which can be described as follows. Given the initial value problem defined by

$$\frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{dt}} = -\mathbf{R}(\bar{\mathbf{U}}), \tag{2.75}$$

$$\bar{\mathbf{U}}(t_0) = \bar{\mathbf{U}}_0, \tag{2.76}$$

where $\bar{\mathbf{U}}_0$ is the initial value of the cell-averaged solution variable $\bar{\mathbf{U}}$ at time $t = 0$ and its rate of change with respect to time is given by the residual, -$\mathbf{R}(\bar{\mathbf{U}})$. The solution, $\bar{\mathbf{U}}^n$, at time t_n, can be advanced to the solution, $\bar{\mathbf{U}}^{n+1}$, at time $t_{n+1} = t_n + h$, through a time step, h, as follows. First, an intermediate value of the solution, $\bar{\mathbf{U}}^{n+1/2}$, is calculated by considering a half time-step

$$\bar{\mathbf{U}}^{n+1/2} = \bar{\mathbf{U}}^n - \mathbf{R}(\bar{\mathbf{U}}^n)\frac{h}{2}. \tag{2.77}$$

This intermediate value is then used to calculate the corresponding residual $\mathbf{R}(\bar{\mathbf{U}}^{n+1/2})$. The final value for $\bar{\mathbf{U}}^{n+1}$ is obtained as

$$\bar{\mathbf{U}}^{n+1} = \bar{\mathbf{U}}^n - \mathbf{R}(\bar{\mathbf{U}}^{n+1/2})h. \tag{2.78}$$

2.5 Block-Based Adaptive Mesh Refinement

The anisotropic block-based AMR scheme adopted here has been found to be highly efficient in terms of reducing the overall mesh size for a given flow problem [1,50,76,79]. A brief summary of this approach is now given in what follows.

2.5.1 Refinement and Coarsening of Individual Blocks

In the proposed block-based AMR approach, the computational domain is divided into subdomains called 'blocks' with each block containing a fixed predetermined number of cells. Each block consists of $N_i \times N_j \times N_k$ cells, with N_i, N_j and N_k being the number of cells in the logical i-, j- and k-directions, respectively. The cell counts in each logical direction, N_i, N_j and N_k, must be even numbers greater than or equal to four, but do not necessarily have to be the same. An example of such a body-fitted multi-block mesh generated using block-based AMR is illustrated in Figure 2.5. Here, a portion of an initial mesh is shown in Figure 2.5(a) and an adapted mesh obtained after 4 adaptive anisotropic refinements on the initial mesh is shown in Figure 2.5(b). As the type of AMR technique used here is anisotropic, a block flagged for refinement is always divided into two, four or eight smaller blocks depending on the number of directions the block is flagged to refine along. The cell resolution is therefore increased only in selected directions, as opposed to all directions as in an isotropic refinement procedure. These new 'children' blocks each contain the same predetermined number of cells. Conversely, two, four or eight neighbouring blocks flagged for coarsening can combine to form a single block. The linear reconstruction procedure described in Section 2.1.5 is used to prolong coarse mesh solution variables from the 'parent' block to its 'children' blocks during refinement. During coarsening, the fine mesh solution variables are restricted back to the 'parent' block. The AMR procedure is restricted such that adjacent blocks having different levels of refinement can differ by a maximum of one refinement level in a particular logical direction.

In this thesis, output-based mesh refinement techniques have been developed and compared with traditional gradient-based refinement techniques for directing the adaptation of the mesh. For output-based refinement techniques, the refinement criteria used to determine the flagging of blocks are based on *a posteriori* error estimates of integrated engineering functionals. The evaluation of these criteria is the focus of Chapter 3. For gradient-based refinement techniques, the refinement criteria used for flagging blocks are gradients of primal flow quantities such as density, pressure, velocity and magnetic field [49, 58, 64]. Blocks with refinement criteria higher than a predetermined refine-

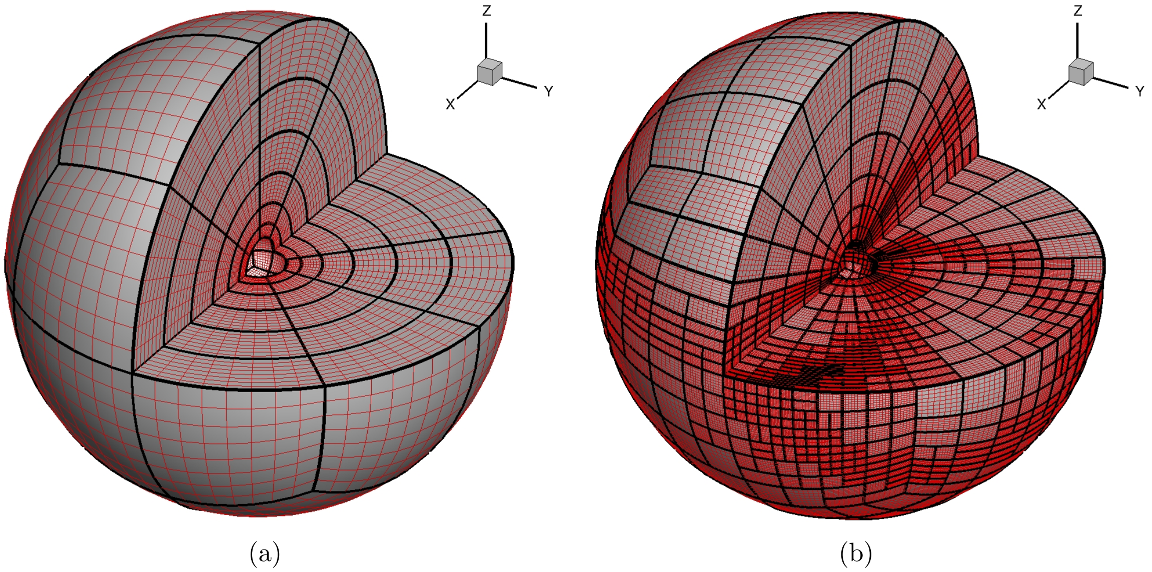


Figure 2.5: (a) An example of an initial bodyfitted mesh. (b) Adapted mesh obtained after 4 adaptive refinements on the initial mesh. The thicker black lines are the block edges while the thinner red lines are the cell edges.

ment threshold are flagged for refinement in the corresponding directions while those with refinement criteria lower than a predetermined coarsening threshold are flagged for coarsening. During AMR, the refinement approach also preserves the primary stretching of the initial mesh required for better capturing of flow features such as boundary layers.

2.5.2 Solution Block Connectivity

In order to update the solution in a computational cell by solving the discretized governing equations, information about the neighbouring cells is required for the purpose of solution reconstruction and flux calculation. In structured cell-based mesh refinement methods, the connectivity between cells is inherent in the indexing of cells using the coordinates i, j, and k representing the position of the cell with respect to the logical directions. For unstructured meshes, the connectivities between all cells must be stored in an appropriate data structure. As the mesh is refined, updated connectivities for the new mesh must be tracked for individual cells for cell-based refinement methods. This makes the connectivity data structure rather memory-intensive for cell-based methods and even more so for unstructured cell-based methods. In the proposed block-based AMR method, the inherent indexing of cells is retained inside individual blocks. Hence, during AMR, only the connectivity between blocks must be maintained and updated,

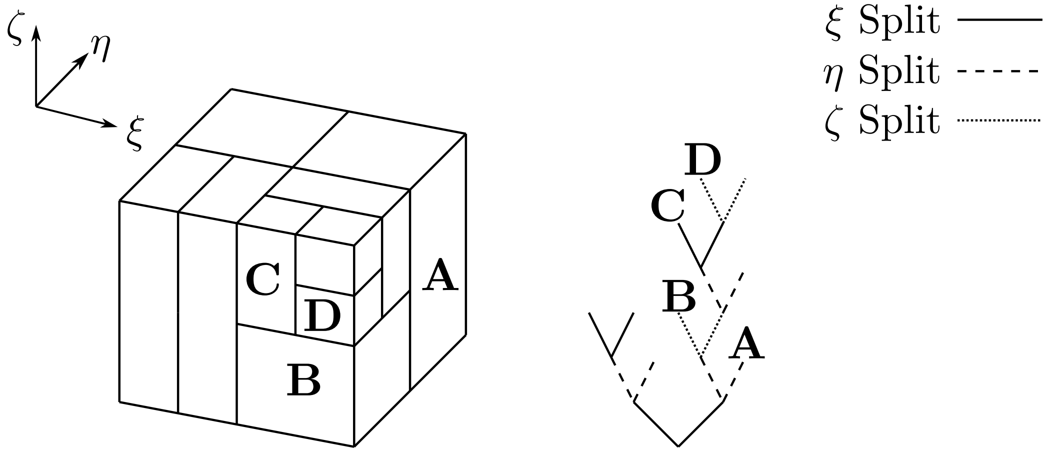


Figure 2.6: 3D binary tree and the corresponding blocks after several refinements.

significantly reducing the memory requirement as compared to that of a cell-based mesh containing the same number of cells.

For the anisotropic block-based AMR technique used here, a hierarchical binary tree structure is used to store and track connectivities between individual blocks. Being lightweight in terms of memory requirements, it is stored on all processors participating in the simulation. This also reduces the time spent in communication between various processors, as they do not need to receive information from other processors to access the binary tree. The binary tree data structure not only keeps track of the connectivity between blocks but also the direction in which a block is refined into two new blocks. A root is assigned to each of the blocks formed during initial mesh generation. As blocks are refined, branches are formed out of each refined block and the leaves of the tree represent blocks in active use, that have no ‘children’. Every branch also stores the logical direction of refinement, while the ‘children’ blocks are assigned a sector, i.e., east or west, north or south, top or bottom, corresponding to the specific half of the ‘parent’ block the ‘children’ occupy. Figure 2.6 shows the resulting binary tree after several refinements of an initial mesh consisting of a single block.

2.5.3 Information Exchange Between Blocks

Each computational block contains two additional layers of ghost cells that overlap with neighbouring blocks. These ghost cells are used to share and exchange solution information between adjacent blocks sharing a common interface. For adjacent blocks belonging to different processors, this information is exchanged through the Message Passing Interface (MPI). These ghost cells are also essential for the solution reconstruction required for

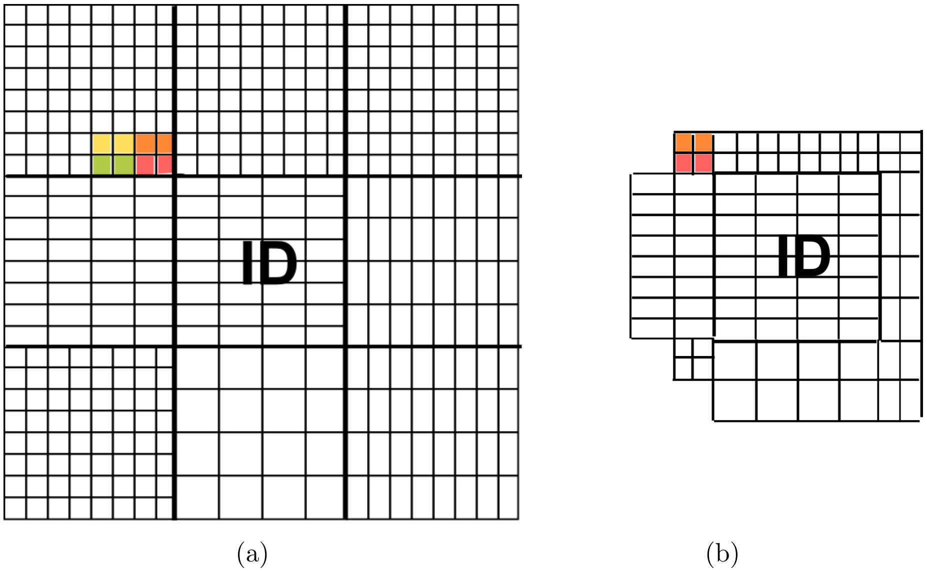


Figure 2.7: (a) Block under consideration (marked ID) surrounding by neighbouring blocks. (b) Ghost cells for the ID block taken directly from the neighbouring blocks.

second order spatial accuracy. For blocks situated at the boundary of the computational domain, ghost cells are also used for imposing appropriate boundary conditions.

For the purpose of this work, the heterogenous block approach developed by Freret and Groth [1, 79] is used here. In this approach, the ghost cells for a block have the same refinement level as the neighbouring blocks overlapping the ghost cells. This is illustrated in Figure 2.7 for clarity. The domain in Figure 2.7 consists of 11 blocks adjacent to each other as shown in Figure 2.7(a). The ghost cells for the block of interest at the centre (ID) are shown in Figure 2.7(b). The ghost cells and their solution values are directly provided by the neighbouring 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 [1, 79]. 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. Eliminating the need for prolongation, restriction and flux corrections also simplifies the construction of the matrix for the adjoint problem. For all of these reasons, the heterogeneous block approach, with ghost cells having the same refinement levels as the neighbouring blocks overlapping them, is used and exploited here.

2.5.4 Parallel Implementation

All the blocks constituting the computational domain are distributed using domain decomposition equally among the processors participating in the simulation. This parallel implementation is carried out using the C++ programming language and the Message Passing Interface (MPI) library of subroutines [189, 190]. The self-similar nature of the proposed block-based AMR approach makes the parallel implementation and load-balancing trivial. Most of the simulations presented in this thesis were performed on an IBM iDataPlex General Purpose Cluster (GPC), built using 3,780 nodes, with each node containing eight 2.53 GHz Intel Xeon E5540 processors and 16 GB of memory per node. More recent simulations were performed on the Niagara supercomputer, built using 1,500 nodes, with each node containing forty 2.4 GHz Intel Skylake cores and 202 GB of memory per node.

Chapter 3

Output-Based Error Estimation for Steady Flows

This chapter focuses on the *a posteriori* output-based error estimation algorithm developed and applied herein for calculating functional error estimates that drive the mesh adaptation. Section 3.1 describes the discrete adjoint formulation, the system of equations required to calculate the adjoint solution and the numerical techniques used to solve this system. Section 3.2 describes the calculation of Jacobians of various residual terms required to construct the linear system for solving the adjoint equation. Section 3.3 discusses the influence of primal boundary conditions on the adjoint solution. Section 3.4 describes the two different kinds of functional error estimates that can be used for driving the mesh adaptation using output-based error estimation. Finally, a summary of the output-based error estimation algorithm used in conjunction with the h-refinement procedure is given in Section 3.5.

3.1 Calculation of the Discrete Adjoint Solution

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 functional 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 [191–195] and mesh refinement [83–89, 196] due to their ability to single out regions of the flow that are more relevant for the accurate calculation of certain quantities of interest.

The adjoint problem can be broadly classified into the continuous and discrete adjoint formulations. The continuous adjoint equation is directly obtained from the differential form of the fluid dynamical conservation equations, and then discretized and solved using a numerical technique.

For this reason, the continuous adjoint approach automatically leads to a discretization that is a consistent approximation of the corresponding adjoint or dual partial-differential equations [197,198]. In the discrete adjoint formulation [191–195], the adjoint equations are derived directly from the spatial discretization of the conservation equations through differentiation with respect to the conserved or primal variables. As discussed by Hicken and Zingg [198], the resulting discrete adjoint equations are typically easier to implement, since they amount to an exercise in differentiation and techniques such as automatic differentiation [199, 200] or the complex-step method [201] can be used to evaluate the Jacobian. Furthermore, as pointed out by Nadarajah [192], the discrete formulation is more consistent with the computed functional as the functional is evaluated numerically using the discretized solution. It is also possible to discretize the governing conservation or primal equations in such a manner that the resulting discrete adjoint is consistent with the continuous dual problem. These so-called adjoint or dual consistent methods have additional beneficial properties associated with the convergence of the functional as the mesh is refined, such as superconvergence [197, 198]. While it is possible to demonstrate dual consistency for many variational-based discretization schemes, for non-variational-based discretizations such as the finite-volume approach considered herein, proof of dual consistency can be more difficult to show. Nevertheless, in this thesis, the discrete adjoint formulation is considered.

Returning to Equation (2.37), the semi-discrete form of the spatially discretized differential equations of interest is given by

$$\frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{dt}} = -\mathbf{R}(\bar{\mathbf{U}}), \tag{3.1}$$

where $\mathbf{R}(\bar{\mathbf{U}})$ represents the discretized residual and $\bar{\mathbf{U}}$ represents the cell-averaged conserved solution. For a steady flow, the solution satisfies $\mathbf{R}(\bar{\mathbf{U}}) = 0$. 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 this coarse grid is given by $\mathbf{R}_H(\bar{\mathbf{U}}_H) = 0$, where $\bar{\mathbf{U}}_H$ is the discrete flow solution on the coarse grid. The so-called 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 denoted as $f_H(\bar{\mathbf{U}}_H)$. 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, $\bar{\mathbf{U}}_h$, is naturally much higher than that of the coarse grid solution, $\bar{\mathbf{U}}_H$. Calculation of the functional, $f_h(\bar{\mathbf{U}}_h)$, by first evaluating the actual fine mesh solution, $\bar{\mathbf{U}}_h$, in order to aid in guiding the mesh adaptation would not be desirable since the expensive computations would have been already performed.

The output-based refinement strategy considered here therefore instead makes use of 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, this estimate 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. This can be accomplished as follows. The coarse grid solution, $\bar{\mathbf{U}}_H$, is first prolonged to the fine grid as

$$\bar{\mathbf{U}}_h^H = \mathbf{I}_h^H \bar{\mathbf{U}}_H, \tag{3.2}$$

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

$$f_h(\bar{\mathbf{U}}_h) = f_h(\bar{\mathbf{U}}_h^H) + \frac{\partial f_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}(\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H) + \mathcal{O}((\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H)^2)\,. \tag{3.3}$$

For the steady problem, the residual on the fine grid is $\mathbf{R}_h(\bar{\mathbf{U}}_h) = 0$. This residual can also then be expanded using a Taylor expansion based on the prolonged solution, $\bar{\mathbf{U}}_h^H$, as

$$\mathbf{R}_h(\bar{\mathbf{U}}_h) = \mathbf{R}_h(\bar{\mathbf{U}}_h^H) + \frac{\partial \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}(\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H) + \mathcal{O}((\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H)^2)\,. \tag{3.4}$$

Note that $\mathbf{R}_h(\bar{\mathbf{U}}_h^H) \neq 0$ on the fine grid because $\bar{\mathbf{U}}_h^H$ is the coarse grid prolonged solution and not the actual fine grid solution, $\bar{\mathbf{U}}_h$, obtained by solving the governing equations on the fine grid. Using $\mathbf{R}_h(\bar{\mathbf{U}}_h) = 0$, the difference between the fine and prolonged grid solutions can be expressed as

$$(\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H) \approx - \left[\frac{\partial \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}\right]^{-1} \mathbf{R}_h(\bar{\mathbf{U}}_h^H)\,. \tag{3.5}$$

Neglecting higher-order terms and using Equation (3.5), Equation (3.3) can be then re-written as

$$f_h(\bar{\mathbf{U}}_h) \approx f_h(\bar{\mathbf{U}}_h^H) - \left(\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H), \tag{3.6}$$

where $\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)$ is defined as the discrete adjoint solution on the fine grid. As can be seen from Equation (3.6), this term can also be viewed as the sensitivity of the functional for the residual error on the fine grid. The second term in Equation (3.6), $\left(\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H)$, is the error estimate required to drive the mesh adaptation. The fine grid discrete adjoint, $\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)$, or so-called dual solution, satisfies the discrete adjoint equation

$$\left(\frac{\partial \mathbf{R}_h}{\partial \bar{\mathbf{U}}_h}\right)^T \boldsymbol{\psi}_h = \left(\frac{\partial f_h}{\partial \bar{\mathbf{U}}_h}\right)^T, \tag{3.7}$$

on the fine grid.

Equation (3.7) is not solved due to its high computational expense. The fine grid discrete adjoint, $\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)$, is instead approximated by an interpolated value, $\boldsymbol{\psi}_h^H$, given by

$$\boldsymbol{\psi}_h^H = \mathbf{J}_h^H \boldsymbol{\psi}_H, \tag{3.8}$$

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

$$\left(\frac{\partial \mathbf{R}_H}{\partial \bar{\mathbf{U}}_H}\right)^T \boldsymbol{\psi}_H = \left(\frac{\partial f_H}{\partial \bar{\mathbf{U}}_H}\right)^T. \tag{3.9}$$

Here, $\mathbf{J}_h^H$ is also a prolongation operator similar to $\mathbf{I}_h^H$. Note that a finer mesh is used here to evaluate the primal residual error, $\mathbf{R}_h(\bar{\mathbf{U}}_h^H)$. An alternate approach to explicit refinement of the mesh 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 as considered in other studies [83,92,146,196]. However, such an approach was not considered as part of this thesis.

3.1.1 Extensive Nature of Discrete Adjoint Variables

The discrete finite-volume residual for the conserved solution variables, $\mathbf{R}_{i,j,k}$, as defined by Equation (2.37), involves division by the cell volume, $V_{i,j,k}$. A consequence of this formulation is that the resulting discrete adjoint variables as defined by Equation (3.9) are extensive-like solution quantities that are proportional to the cell size. Moreover, the adjoint variables approach zero as the cell size approaches zero. This behaviour is somewhat different from other previous formulations of adjoint-based methods by other authors where, due to the definition of the discrete residual, the adjoint variables are intensive-like solution quantities that converge to a finite value as the mesh is refined. However, a simple division of the current adjoint variables as defined by Equation (3.9) results in an intensive quantity that converges to a finite value as the mesh in refined. Hence, for visualization, plotting, and comparison purposes, the adjoint variables have been divided by the local cell volume in all plots and figures given in this thesis.

3.1.2 Solving the Linear System

Equation (3.9) is a linear system of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$. As in the Newton iterative scheme considered in Chapter 2, this system is a large sparse linear system distributed across the processors participating in the simulations and is solved using preconditioned GMRES. The Trilinos software package is used in combination with MPI for this purpose. Trilinos [202–206] contains subpackages like Epetra [207] for the various matrix and vector classes for data distribution over multiple processors and AztecOO [208] for routines like GMRES and preconditioning for faster convergence of linear systems. The incomplete LU and multilevel [209] preconditioners have been used exclusively for this work.

Although the preceding iterative procedures proved very effective for solution of the linear system of Equation (3.9) for all of the gaseous and MHD flow cases presented in this thesis, in some other cases not reported here, the linear system could end up being poorly conditioned and not diagonally dominant. For instance, this situation could arise due to large non-diagonal source terms occuring in the linear system associated with rotational terms arising from solving the governing equations in a rotating frame of reference. For such cases, an implicit pseudo-time-marching scheme was developed and employed here wherein the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ was re-cast as an unsteady system of ordinary differential equations of the form

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau} = -(\mathbf{A}\mathbf{x} - \mathbf{b}), \tag{3.10}$$

where τ is a pseudo-time variable. Applying a first-order implicit Euler time discretization, the solution for $\mathbf{x}$ can be written as

$$\left(\mathbf{A} + \frac{\mathbf{I}}{\Delta\tau}\right)\mathbf{x}_f = \mathbf{b} + \frac{\mathbf{x}_i}{\Delta\tau}, \tag{3.11}$$

where $\mathbf{I}$ is an identity matrix, $\Delta\tau$ is the pseudo-time-step, and $\mathbf{x}_i$ and $\mathbf{x}_f$ are solutions at the beginning and end of the time-step, respectively. The adjoint solution is then obtained by iteratively solving Equation (3.11) until the norm $||\mathbf{A}\mathbf{x} - \mathbf{b}||_\infty$ converges to a predetermined tolerance level. In order to obtain the adjoint solution efficiently, the linear system shown in Equation (3.11) need not be solved until convergence at each iteration. Accordingly, a modest tolerance of 10^{-4} could be used here. For more rapid convergence, the time-step is increased after every iteration as

$$\Delta\tau_{k+1} = \Delta\tau_0 \left(\frac{||\mathbf{A}\mathbf{x}_0 - \mathbf{b}||_\infty}{||\mathbf{A}\mathbf{x}_k - \mathbf{b}||_\infty} \right)^{1/2}, \tag{3.12}$$

where $\Delta\tau_{k+1}$ is the time-step for the $(k+1)$th iteration, $\Delta\tau_0$ is an initially prescribed time-step, $\mathbf{x}_k$ is the solution at the end of the kth iteration and $\mathbf{x}_0$ is an initial guess for the solution vector $\mathbf{x}$. Typically, 100 to 200 iterations are needed for large meshes to obtain an accurate adjoint solution, using the proposed modified technique. While possibly not optimal, the procedure above ensured a diagonal dominant form for the linear equations for $\mathbf{x}$ and provided a relatively straightforward approach to obtaining converged solutions for problematic cases. However, it should again be emphasized that this implicit time marching technique was not required and has not been used for any of the cases reported in the thesis.

3.2 Evaluation of the Residual Jacobian

Solving Equation (3.9) requires evaluating the matrix $(\partial\mathbf{R}_H/\partial\bar{\mathbf{U}}_H)^T$ and this requires the calculation of Jacobians of residuals of each grid cell with respect to its own conserved state as well as conserved states of neighbouring cells. The semi-discrete form of the residual shown in Equation (3.1) for a generic cell (i, j, k) contains inviscid (hyperbolic) fluxes for all sets of governing equations considered in this thesis. Additionally, the Navier-Stokes equations also contain viscous (elliptic) fluxes while the ideal MHD equations contain source terms. Analytical approximations of the Jacobian matrix, $\mathbf{J}$, defined by

$$\mathbf{J} \approx \left(\frac{\partial\mathbf{R}_H}{\partial\bar{\mathbf{U}}_H} \right)^T, \tag{3.13}$$

are used here for calculating the adjoint solution. The exact second-order Jacobian matrices, $(\partial\mathbf{R}_H/\partial\bar{\mathbf{U}}_H)^T$, are in general difficult and computationally expensive to evaluate, particularly when the limiter is applied, and thus, only a first-order approximation is used here. A finite-difference Jacobian calculation method has also been implemented, wherein, in every cell, a small perturbation is given to each of the conserved quantities and the perturbed residual is used to obtain the residual Jacobian. For the purpose of the current work, a second-order central-difference scheme has been implemented. Due to its high computational cost, the finite-difference method was only used for validating the analytical Jacobian calculation method.

More accurate evaluation of the Jacobian can be obtained by using the complex-step method [201], wherein a complex perturbation is given to calculate the residual perturbation. Automatic differentiation libraries [199,200] can also be used for obtaining highly accurate values of the residual Jacobian. Such approaches, however, were deemed beyond the scope of this thesis. It was felt that, while the approximate expressions for the residual Jacobian would somewhat reduce the accuracy of the adjoint solution, the approach would be sufficient for use in guiding mesh adaptation in the block-based AMR method considered here. The following sections describe the approximations used to evaluate analytically each of the components that constitute the residual Jacobian.

3.2.1 Approximate Inviscid Flux Jacobian

The inviscid fluxes are calculated using an upwind Godunov-type scheme and make use of a Riemann-solver based flux function, namely the HLLE scheme. The reference frame is first rotated to a local reference coordinate frame wherein the x-direction is aligned with the normal to the cell face across which the flux is being calculated. After evaluating the flux, the frame is rotated back to the original orientation. This process can be written as

$$\mathbf{F}\cdot\mathbf{n} = \mathcal{F}(\mathbf{U}_L,\mathbf{U}_R,\mathbf{n}) = \mathbf{A}^{-1}\mathcal{F}(\mathbf{A}\mathbf{U}_L,\mathbf{A}\mathbf{U}_R), \tag{3.14}$$

where $\mathbf{A}$ is a rotation matrix that rotates the momentum and magnetic field vectors to the local frame, leaving mass and energy unchanged. The solution quantities, $\mathbf{U}_L$ and $\mathbf{U}_R$, are the left and right states, respectively while $\mathcal{F}$ is a flux function calculating the x-direction flux for the left and right conserved states passed to it in that order. As per convention, the left state $\mathbf{U}_L$ is $\bar{\mathbf{U}}_{ijk}$, the cell-averaged conserved state in the cell (i,j,k) and the face normal points away from (i,j,k) in the local x-direction.

We first consider the Jacobian of the inviscid component, $\mathbf{R}_I$, of the residual of the cell (i,j,k) with respect to the conserved variables in the cell given by

$$\mathbf{R}_I = -\frac{1}{V_{i,j,k}}\sum_{f=1}^{N_{f_{i,j,k}}}\left(\mathbf{F}_{face}\cdot\mathbf{n}_{face}\Delta A_{face}\right)_{i,j,k}. \tag{3.15}$$

Hence, the Jacobian is

$$\left(\frac{\partial\mathbf{R}_{I,ijk}}{\partial\mathbf{U}_{ijk}}\right) = \frac{-1}{V_{i,j,k}}\sum_{f=1}^{N_{f_{i,j,k}}}\left(\mathbf{A}^{-1}\frac{\partial\mathcal{F}(\mathbf{A}\mathbf{U}_L,\mathbf{A}\mathbf{U}_R)}{\partial(\mathbf{A}\mathbf{U}_L)}\mathbf{A}\Delta A_{face}\right)_{i,j,k}, \tag{3.16}$$

where the chain rule is used to write

$$\frac{\partial \mathcal{F}(\mathbf{AU}_L, \mathbf{AU}_R)}{\partial(\mathbf{U}_L)} = \frac{\partial \mathcal{F}(\mathbf{AU}_L, \mathbf{AU}_R)}{\partial(\mathbf{AU}_L)} \frac{\partial(\mathbf{AU}_L)}{\partial \mathbf{U}_L} = \frac{\partial \mathcal{F}(\mathbf{AU}_L, \mathbf{AU}_R)}{\partial(\mathbf{AU}_L)} \mathbf{A}. \tag{3.17}$$

Here, $\partial \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)/\partial \mathbf{U}_L$ is the Jacobian of the flux function, $\mathcal{F}$, with respect to the first argument, $\mathbf{U}_L$, or $\bar{\mathbf{U}}_{ijk}$, i.e. the "left" state of the Riemann problem. Since the finite-volume method is conservative, the flux leaving the cell (i, j, k) through the face under consideration is equal to the flux entering the neighbour sharing the same face. Hence, the Jacobians on the right-hand side of Equation (3.16) also give the residual Jacobians of the corresponding neighbouring cells with respect to $\mathbf{U}_{ijk}$ after reversing the sign. In this way, the Jacobians of the residual of any cell with respect to the conserved variables in the same cell as well as its neighbours can be found, and the entire matrix can be constructed. Thus, the first-order approximate Jacobian of the HLLE flux function given in Equation (2.47) becomes

$$\frac{\partial \mathcal{F}_{HLLE}(\mathbf{AU}_L, \mathbf{AU}_R)}{\partial \mathbf{U}_L} \approx \begin{cases} \frac{\partial \mathbf{F}_L}{\partial \mathbf{U}_L} & for\ \frac{x}{t} \leq \lambda^- \\ \frac{\lambda^+}{\lambda^+ - \lambda^-} \frac{\partial \mathbf{F}_L}{\partial \mathbf{U}_L} - \frac{\lambda^+ \lambda^-}{\lambda^+ - \lambda^-} \mathbf{I} & for\ \lambda^- < \frac{x}{t} < \lambda^+ \\ \mathbf{0} & for\ \lambda^+ \leq \frac{x}{t} \end{cases}, \tag{3.18}$$

where $\partial \mathbf{F}_L/\partial \mathbf{U}_L$ is a standard Jacobian of the x-direction flux vector with respect to the conserved solution vector and is readily available and $\mathbf{I}$ is an identity matrix.

3.2.2 Approximate Viscous Flux Jacobian

The Navier-Stokes equations contain viscous fluxes as shown in Equation (2.11). These fluxes depend not only on the values of the solution at the faces but also on their gradients. as shown in Equation (2.55). For a viscous flux, $\mathbf{F}_V$, calculated at a face $(i + 1/2, j, k)$, its Jacobian with respect to the cell-averaged conserved vector, $\bar{\mathbf{U}}_{ijk}$, in cell (i, j, k) can be calculated using a three-step process. Using the chain rule for differentiation, this process is written as

$$\frac{\partial \mathbf{F}_{V,ijk}}{\partial \bar{\mathbf{U}}_{ijk}} = \frac{\partial \mathbf{F}_{V,ijk}}{\partial \mathbf{E}_{i+1/2,j,k}} \frac{\partial \mathbf{E}_{i+1/2,j,k}}{\partial \mathbf{W}_{ijk}} \frac{\partial \mathbf{W}_{ijk}}{\partial \bar{\mathbf{U}}_{ijk}}, \tag{3.19}$$

where $\mathbf{W}_{ijk}$ is the vector of primitive variables for the cell (i, j, k) and $\mathbf{E}_{i+1/2,j,k}$ is the extended solution vector containing primitive variables and their gradients at the face

$(i+1/2,j,k)$. The vector, $\mathbf{E}_{i+1/2,j,k}$, is given by

$$\mathbf{E}_{i+1/2,j,k} = \left[\rho, u, v, w, p, \frac{\partial \rho}{\partial x}, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}, \frac{\partial w}{\partial x}, \frac{\partial p}{\partial x}, \frac{\partial \rho}{\partial y}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial y}, \frac{\partial w}{\partial y}, \frac{\partial p}{\partial y}, \frac{\partial \rho}{\partial z}, \frac{\partial u}{\partial z}, \frac{\partial v}{\partial z}, \frac{\partial w}{\partial z}, \frac{\partial p}{\partial z}\right]^T . \tag{3.20}$$

Here, $\partial \mathbf{W}_{ijk}/\partial \bar{\mathbf{U}}_{ijk}$ is a standard Jacobian of the primitive solution vector with respect to the conserved solution vector for the cell (i,j,k) and is readily available. Similarly, $\partial \mathbf{F}_{V,ijk}/\partial \mathbf{E}_{i+1/2,j,k}$ can also be easily calculated from Equation (3.20) and the definition of viscous fluxes in terms of primitive variables and their gradients shown in the function in Equation (2.55). The calculation of the term, $\partial \mathbf{E}_{i+1/2,j,k}/\partial \mathbf{W}_{ijk}$, can be obtained by differentiating the individual components of $\mathbf{E}_{i+1/2,j,k}$ as follows. The vector, $\mathbf{E}_{i+1/2,j,k}$, is composed of the primitive variables, $\mathbf{W}_{i+1/2,j,k}$, and their gradients, $\nabla \mathbf{W}|_{i+1/2,j,k}$, at the face $(i+1/2,j,k)$. As shown in Equation (2.56), $\mathbf{W}_{i+1/2,j,k}$ is given by

$$\mathbf{W}_{i+1/2,j,k} = \frac{\mathbf{W}_L + \mathbf{W}_R}{2}. \tag{3.21}$$

where, as per convention, $\mathbf{W}_L$ is $\mathbf{W}_{ijk}$, i.e. the "left" primitive state and $\mathbf{W}_R$ is the "right" primitive state. Thus, the first-order approximation for $\partial \mathbf{W}_{i+1/2,j,k}/\partial \mathbf{W}_{ijk}$ is given by

$$\frac{\partial \mathbf{W}_{i+1/2,j,k}}{\partial \mathbf{W}_{ijk}} \approx \frac{1}{2}\mathbf{I}, \tag{3.22}$$

where $\mathbf{I}$ is an identity matrix. As shown in Equation (2.57), $\nabla \mathbf{W}|_{i+1/2,j,k}$ is given by

$$\nabla \mathbf{W}|_{i+1/2,j,k} = \frac{\mathbf{W}_{i+1,j,k} - \mathbf{W}_{i,j,k}}{\Delta s}\frac{\mathbf{n}}{\mathbf{n}\cdot\mathbf{e}_s} + \left(\overline{\nabla \mathbf{W}} - \overline{\nabla \mathbf{W}}\cdot\mathbf{e}_s\frac{\mathbf{n}}{\mathbf{n}\cdot\mathbf{e}_s}\right), \tag{3.23}$$

where $\overline{\nabla \mathbf{W}}$ is the weighted average of the cell centered gradient at the face $(i+1/2,j,k)$, $\mathbf{n}$ is the unit normal vector at the cell interface, Δs is the distance between the centroids of the cells sharing the face and $\mathbf{e}_s$ is the unit normal vector between the centroids. Hence, neglecting higher-order terms, the first-order approximation for $\partial \nabla \mathbf{W}|_{i+1/2,j,k}/\partial \mathbf{W}_{ijk}$ is given by

$$\frac{\partial \nabla \mathbf{W}|_{i+1/2,j,k}}{\partial \mathbf{W}_{ijk}} \approx -\frac{1}{\Delta s}\frac{\mathbf{n}}{\mathbf{n}\cdot\mathbf{e}_s}\mathbf{I}, \tag{3.24}$$

where $\mathbf{I}$ is an identity matrix. Values for $\partial \mathbf{E}_{i+1/2,j,k}/\partial \mathbf{W}_{ijk}$ can be calculated using Equations (3.22) and (3.24).

3.2.3 Approximate Source Term Jacobian

The ideal MHD equations contain the rotational source term, $\mathbf{Q}$, and the Powell source term, $\mathbf{S}$, as shown in Equation (2.24). The term, $\mathbf{Q}$, for a cell (i,j,k) depends only on the solution variables in the cell (i,j,k). Hence, its Jacobian with respect to $\bar{\mathbf{U}}_{ijk}$ can be simply calculated as

$$\frac{\partial \mathbf{Q}_{ijk}}{\partial \bar{\mathbf{U}}_{ijk}}. \tag{3.25}$$

The Jacobian for the Powell source term can be simplified using differentiation by parts and written as

$$\frac{\partial \mathbf{S}_{ijk}}{\partial \bar{\mathbf{U}}_{ijk}} = -(\nabla\cdot\mathbf{B})_{ijk}\frac{\partial}{\partial \bar{\mathbf{U}}_{ijk}}\begin{bmatrix}0\\ \mathbf{B}\\ \mathbf{u}\cdot\mathbf{B}\\ \mathbf{u}\end{bmatrix}_{ijk} - \begin{bmatrix}0\\ \mathbf{B}\\ \mathbf{u}\cdot\mathbf{B}\\ \mathbf{u}\end{bmatrix}_{ijk}\frac{\partial}{\partial \bar{\mathbf{U}}_{ijk}}(\nabla\cdot\mathbf{B})_{ijk}. \tag{3.26}$$

Here, $(\nabla\cdot\mathbf{B})_{ijk}$ is the discrete divergence of the magnetic field which should be zero upto machine accuracy for steady ideal MHD flows. Hence, the Jacobian for the Powell source term given in Equation (3.26) further simplifies as

$$\begin{aligned}\frac{\partial \mathbf{S}_{ijk}}{\partial \bar{\mathbf{U}}_{ijk}} &= -\begin{bmatrix}0\\ \mathbf{B}\\ \mathbf{u}\cdot\mathbf{B}\\ \mathbf{u}\end{bmatrix}_{ijk}\frac{\partial}{\partial \bar{\mathbf{U}}_{ijk}}(\nabla\cdot\mathbf{B})_{ijk}\\ &= -\begin{bmatrix}0\\ \mathbf{B}\\ \mathbf{u}\cdot\mathbf{B}\\ \mathbf{u}\end{bmatrix}_{ijk}\frac{1}{V_{i,j,k}}\frac{\partial}{\partial \bar{\mathbf{U}}_{ijk}}\sum_{f=1}^{N_{f_{i,j,k}}}\left(\mathbf{B}_{face}\cdot\mathbf{n}_{face}\Delta A_{face}\right)_{i,j,k},\end{aligned} \tag{3.27}$$

where $V_{i,j,k}$, $\mathbf{B}_{face}$, $\mathbf{n}_{face}$ and ΔA_{face} denote the cell volume, magnetic field vector at the face centroid, outward pointing unit normal vector and the area of the cell face, respectively, and $N_{f_{i,j,k}}$ represents the number of faces for cell (i,j,k). The magnetic field vector at the face, $\mathbf{B}_{face}$, is evaluated as

$$\mathbf{B}_{face} = \frac{\mathbf{B}_L + \mathbf{B}_R}{2}, \tag{3.28}$$

where $\mathbf{B}_L$ and $\mathbf{B}_R$ are reconstructed magnetic fields at the face obtained from the two cells sharing the face. Again, the "left" state is $\bar{\mathbf{U}}_{ijk}$. A first-order approximation for the above Jacobian can be simply obtained by using the relation

$$\frac{\partial \mathbf{B}_{face}}{\partial \bar{\mathbf{U}}_{ijk}} \approx \frac{1}{2}\frac{\partial \mathbf{B}_L}{\partial \mathbf{U}_L}, \tag{3.29}$$

where $\partial \mathbf{B}_L/\partial \mathbf{U}_L$ is a standard Jacobian and is readily available. Similarly, the residual with respect to the neighbouring state can be calculated by differentiating $\mathbf{B}_{face}$ with respect to $\mathbf{U}_R$ instead of $\mathbf{U}_L$.

3.3 Influence of Boundary Conditions on the Adjoint Solution

The matrix $\mathbf{A} = [\partial \mathbf{R}_H/\partial \bar{\mathbf{U}}_H]^T$ in Equation (3.9) contains the Jacobians of residuals of cells adjacent to the boundaries of the computational domain. The associated residuals contain fluxes evaluated at the cell faces lying on the boundaries of the computational domain. In general, these boundary fluxes depend on the conserved solution in the interior cell adjacent to the boundary as well the conserved solution in the boundary ghost cell sharing a face with the interior cell. The boundary flux can be denoted by $\mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)$ where $\mathbf{U}_L = \bar{\mathbf{U}}_{ijk}$ is the conserved solution in the interior cell (i, j, k) and $\mathbf{U}_R = \bar{\mathbf{U}}_{B,ijk}$ is the conserved solution in the boundary ghost cell sharing a face with the interior cell (i, j, k). The calculation of the Jacobians of such fluxes is not as simple as described in the previous section and requires some special treatment.

The Jacobian of the boundary flux with respect to $\mathbf{U}_L$ does not simply involve the differentiation of the flux with respect to its first argument, $\mathbf{U}_L$, because the second argument, $\mathbf{U}_R$, may also be a function of $\mathbf{U}_L$ depending on how the boundary condition is imposed. Using differentiation by parts and the chain rule, the Jacobian can be calculated as

$$\frac{\partial \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)}{\partial (\mathbf{U}_L)} = \left.\frac{\partial \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)}{\partial (\mathbf{U}_L)}\right|_{\mathbf{U}_R} + \left.\frac{\partial \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)}{\partial (\mathbf{U}_R)}\right|_{\mathbf{U}_L} \underbrace{\frac{\partial \mathbf{U}_R}{\partial \mathbf{U}_L}}_{\mathbf{D}}. \tag{3.30}$$

The Jacobians $\partial \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)/\partial \mathbf{U}_L$ and $\partial \mathcal{F}(\mathbf{U}_L, \mathbf{U}_R)/\partial \mathbf{U}_R$ can be evaluated as described in the previous section. The matrix $\mathbf{D} = \partial \mathbf{U}_R/\partial \mathbf{U}_L$ depends on the specific condition that is imposed at the boundary. The calculation of such matrices will now be discussed for the various boundary conditions that have been used in this thesis.

For a fixed or constant flow boundary condition, such as for an incoming supersonic flow, $\mathbf{U}_R$ is independent of $\mathbf{U}_L$ and the matrix $\mathbf{D}$ is a null matrix with zero values throughout. For a boundary where constant extrapolation is imposed, such as for an outgoing supersonic flow, $\mathbf{U}_R$ is equal to $\mathbf{U}_L$ and the matrix $\mathbf{D}$ is an identity matrix, $\mathbf{D} = \mathbf{I}$. For more complex boundary conditions, the vector $\mathbf{U}_L$ is first expressed in terms of the primitive variables and written as

$$\mathbf{U}_L = \begin{bmatrix} U_{1,L} \\ U_{2,L} \\ U_{3,L} \\ U_{4,L} \\ U_{5,L} \\ U_{6,L} \\ U_{7,L} \\ U_{8,L} \end{bmatrix} = \begin{bmatrix} \rho_L \\ \rho_L u_L \\ \rho_L v_L \\ \rho_L w_L \\ e_L \\ B_{x,L} \\ B_{y,L} \\ B_{z,L} \end{bmatrix}, \tag{3.31}$$

where ρ_L is the density, u_L, v_L, w_L are the x-, y-, z-components of the velocity $\mathbf{u}_L$ and $B_{x,L}, B_{y,L}, B_{z,L}$ are the x-, y-, z-components of the magnetic field $\mathbf{B}_L$. The specific total energy is given by

$$e_L = \frac{\rho_L |\mathbf{u}_L|^2}{2} + \frac{p_L}{\gamma - 1} + \frac{|\mathbf{B}_L|^2}{2}, \tag{3.32}$$

where p_L is the thermal pressure. For the non-conducting solutions of the Euler and Navier-Stokes equations, $\mathbf{B}$ can be taken to be zero.

At a reflective or symmetry plane boundary condition, the density and thermal pressure from the interior cell are extrapolated such that the gradient of these quantities is zero in a direction normal to the boundary while the velocity and magnetic field vectors in the ghost cell are reflections of the corresponding interior cell quantities across the plane containing the boundary face such that the normal components of the two vectors are zero at the boundary. For a unit normal of the face given by $\hat{k} = (\hat{k}_x, \hat{k}_y, \hat{k}_z)$, the reflection of $\mathbf{u}_L$ is given by $\mathbf{u}_R = \mathbf{u}_L - 2(\mathbf{u}_L \cdot \hat{k})\hat{k}$. Similarly, reflection of $\mathbf{B}_L$ is given by $\mathbf{B}_R = \mathbf{B}_L - 2(\mathbf{B}_L \cdot \hat{k})\hat{k}$. The magnitudes of the velocity and magnetic field vectors do not change after reflection, keeping the total energy in the boundary ghost cell the same as that in the interior cell. Thus, $\mathbf{U}_R$ is given by

$$\mathbf{U}_R = \begin{bmatrix} \rho_L \\ \rho_L(u_L - 2(\mathbf{u}_L \cdot \hat{k})\hat{k}_x) \\ \rho_L(v_L - 2(\mathbf{u}_L \cdot \hat{k})\hat{k}_y) \\ \rho_L(w_L - 2(\mathbf{u}_L \cdot \hat{k})\hat{k}_z) \\ e_L \\ B_{x,L} - 2(\mathbf{B}_L \cdot \hat{k})\hat{k}_x \\ B_{y,L} - 2(\mathbf{B}_L \cdot \hat{k})\hat{k}_y \\ B_{z,L} - 2(\mathbf{B}_L \cdot \hat{k})\hat{k}_z \end{bmatrix} = \begin{bmatrix} U_{1,L} \\ U_{2,L} - 2(U_{2,L}\hat{k}_x + U_{3,L}\hat{k}_y + U_{4,L}\hat{k}_z)\hat{k}_x \\ U_{3,L} - 2(U_{2,L}\hat{k}_x + U_{3,L}\hat{k}_y + U_{4,L}\hat{k}_z)\hat{k}_y \\ U_{4,L} - 2(U_{2,L}\hat{k}_x + U_{3,L}\hat{k}_y + U_{4,L}\hat{k}_z)\hat{k}_z \\ U_{5,L} \\ U_{6,L} - 2(U_{6,L}\hat{k}_x + U_{7,L}\hat{k}_y + U_{8,L}\hat{k}_z)\hat{k}_x \\ U_{7,L} - 2(U_{6,L}\hat{k}_x + U_{7,L}\hat{k}_y + U_{8,L}\hat{k}_z)\hat{k}_y \\ U_{8,L} - 2(U_{6,L}\hat{k}_x + U_{7,L}\hat{k}_y + U_{8,L}\hat{k}_z)\hat{k}_z \end{bmatrix}. \qquad (3.33)$$

Thus, $\mathbf{D} = \partial \mathbf{U}_R / \partial \mathbf{U}_L$ for a reflective boundary condition can be obtained by differentiating Equation (3.33) with respect to $\mathbf{U}_L$ to obtain

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-2\hat{k}_x^2 & -2\hat{k}_x\hat{k}_y & -2\hat{k}_x\hat{k}_z & 0 & 0 & 0 & 0 \\ 0 & -2\hat{k}_y\hat{k}_x & 1-2\hat{k}_y^2 & -2\hat{k}_y\hat{k}_z & 0 & 0 & 0 & 0 \\ 0 & -2\hat{k}_z\hat{k}_x & -2\hat{k}_z\hat{k}_y & 1-2\hat{k}_z^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\hat{k}_x^2 & -2\hat{k}_x\hat{k}_y & -2\hat{k}_x\hat{k}_z \\ 0 & 0 & 0 & 0 & 0 & -2\hat{k}_y\hat{k}_x & 1-2\hat{k}_y^2 & -2\hat{k}_y\hat{k}_z \\ 0 & 0 & 0 & 0 & 0 & -2\hat{k}_z\hat{k}_x & -2\hat{k}_z\hat{k}_y & 1-2\hat{k}_z^2 \end{bmatrix}. \qquad (3.34)$$

The corresponding matrices for reflective, fixed and constant extrapolation boundary conditions for Euler and Navier-Stokes types of flows can be obtained by simply disregarding the magnetic field in the above procedure.

For an adiabatic, no-slip boundary condition encountered in viscous aerodynamic flows, the flow velocity is assumed to be zero at the wall. For a first-order approximation, the velocity vector in the boundary ghost cell is taken to be equal in magnitude and opposite in direction to the velocity vector in the corresponding interior cell. The density and pressure in the ghost cell are assumed to be equal to those in the interior cell, again corresponding to zero gradients for these quantities in the boundary normal direction.

Hence, $\mathbf{U}_R$ and $\mathbf{D} = \partial \mathbf{U}_R / \partial \mathbf{U}_L$ are given by

$$\mathbf{U}_R = \begin{bmatrix} U_{1,L} \\ -U_{2,L} \\ -U_{3,L} \\ -U_{4,L} \\ U_{5,L} \end{bmatrix}, \; \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.35}$$

For an incoming subsonic flow, the thermal pressure in the boundary ghost cell is assigned to be equal to the thermal pressure p_L in the interior cell, whereas the density and velocity are held constant. For a constant density ρ_0 and constant velocity $\mathbf{u}_0 = (u_0, v_0, w_0)$, $\mathbf{U}_R$ is given by

$$\mathbf{U}_R = \begin{bmatrix} \rho_0 \\ \rho_0 u_0 \\ \rho_0 v_0 \\ \rho_0 w_0 \\ \frac{p_L}{\gamma-1} + \frac{\rho_0}{2}|\mathbf{u}_0|^2 \end{bmatrix} = \begin{bmatrix} \rho_0 \\ \rho_0 u_0 \\ \rho_0 v_0 \\ \rho_0 w_0 \\ U_{5,L} - \frac{U_{2,L}^2 + U_{3,L}^2 + U_{4,L}^2}{2U_1} + \frac{\rho_0}{2}|\mathbf{u}_0|^2 \end{bmatrix}. \tag{3.36}$$

Hence, $\mathbf{D} = \partial \mathbf{U}_R / \partial \mathbf{U}_L$ is given by

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{U_2^2 + U_3^2 + U_4^2}{2U_1^2} & -\frac{U_2}{U_1} & -\frac{U_3}{U_1} & -\frac{U_4}{U_1} & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{|\mathbf{u}_L|^2}{2} & -u_L & -v_L & -w_L & 1 \end{bmatrix}. \tag{3.37}$$

For an outgoing subsonic flow, the thermal pressure in the boundary ghost cell is kept fixed whereas the density and velocity are assigned to be equal to those in the interior cell. For a constant pressure p_0, $\mathbf{U}_R$ is given by

$$\mathbf{U}_R = \begin{bmatrix} \rho_L \\ \rho_L u_L \\ \rho_L v_L \\ \rho_L w_L \\ \frac{p_0}{\gamma-1} + \frac{\rho_L}{2}(u_L^2 + v_L^2 + w_L^2) \end{bmatrix} = \begin{bmatrix} U_{1,L} \\ U_{2,L} \\ U_{3,L} \\ U_{4,L} \\ \frac{p_0}{\gamma-1} + \frac{U_{2,L}^2 + U_{3,L}^2 + U_{4,L}^2}{2U_{1,L}} \end{bmatrix}, \tag{3.38}$$

and in this case, $\mathbf{D} = \partial \mathbf{U}_R / \partial \mathbf{U}_L$ is given by

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{U_2^2+U_3^2+U_4^2}{2U_1^2} & \frac{U_2}{U_1} & \frac{U_3}{U_1} & \frac{U_4}{U_1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{|\mathbf{u}_L|^2}{2} & u_L & v_L & w_L & 0 \end{bmatrix}. \tag{3.39}$$

It should be noted that accounting for the influence of primal boundary conditions is essential for accurate values of computable corrections, especially for functionals calculated by integrating primitive solution variables over sections of domain boundaries. This is usually the case for engineering functionals such as forces and moments over airfoil configurations. Incorrect implementation of boundary conditions can lead to less accurate and also incorrect values of computable corrections. This would also lead to incorrect values of the calculated adjoint solution, which can cause the solution method to refine in unwanted regions when applying output-based refinement methods.

3.4 Output-Based Criteria for Mesh Adaptation

One way to drive the grid refinement in output-based adaptive methods is to directly use the local contribution of the computable correction in the functional, $\left(\boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H)$, calculated as described in Section 3.1, as the refinement criterion. The interpolated adjoint described in Equation (3.8) is used for this purpose. As described by Venditti and Darmofal [85–87], this approach can however 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. In order to understand these two variants, the higher-order terms neglected in Equation (3.6) must be reconsidered. Equation (3.6) can be written as

$$f_h(\bar{\mathbf{U}}_h) - f_h(\bar{\mathbf{U}}_h^H) \approx \underbrace{-\left(\boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}_{\text{Computable Correction}} - \underbrace{\left(\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H) - \boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}_{\text{Error in Computable Correction}}, \tag{3.40}$$

where $\boldsymbol{\psi}_h^H$ is the interpolated coarse-grid adjoint shown in Equation (3.8) whereas $\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)$ is the more accurate value of the adjoint calculated on the fine grid. As mentioned earlier, values for $\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H)$ are not calculated due to the relatively high computational cost. Hence, the term $\left(\boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H) - \boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H)$ can be identified as the error in the computable correction. Equation (3.40) can also be written as

$$f_h(\bar{\mathbf{U}}_h) - f_h(\bar{\mathbf{U}}_h^H) \approx \underbrace{-\left(\boldsymbol{\psi}_h^H\right)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}_{\text{Computable Correction}} + \underbrace{\mathbf{R}_h^{\boldsymbol{\psi}}(\boldsymbol{\psi}_h^H)^T \left[\frac{\partial \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}\right]^{-1} \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}_{\text{Error in Computable Correction}}, \tag{3.41}$$

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

$$\mathbf{R}_h^{\boldsymbol{\psi}}(\boldsymbol{\psi}) = \left[\frac{\partial \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}\right]^T \boldsymbol{\psi} - \left(\frac{\partial f_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}\right)^T . \tag{3.42}$$

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

$$\left[\frac{\partial \mathbf{R}_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}\right]^T \boldsymbol{\psi}_h(\bar{\mathbf{U}}_h^H) = \left(\frac{\partial f_h(\bar{\mathbf{U}}_h^H)}{\partial \bar{\mathbf{U}}_h}\right)^T , \tag{3.43}$$

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

Hence, as indicated by Equation (3.41), 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. Thus, Venditti and Darmofal [85–87] proposed using a combination of 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 [85–87] for a particular coarse grid cell, i, is given by

$$\begin{aligned} \boldsymbol{\varepsilon}_i =& \frac{1}{2} \sum_{l(i)} \Big(\left| [Q_h^H \boldsymbol{\psi}_H - L_h^H \boldsymbol{\psi}_H]_{l(i)}^T [\mathbf{R}_h(L_h^H \bar{\mathbf{U}}_H)]_{l(i)} \right| \\ &+ \left| [Q_h^H \bar{\mathbf{U}}_H - L_h^H \bar{\mathbf{U}}_H]_{l(k)}^T [\mathbf{R}_h^{\boldsymbol{\psi}}(L_h^H \boldsymbol{\psi}_H)]_{l(i)} \right| \Big). \end{aligned} \tag{3.44}$$

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.

Henceforth, the output-based adaptation parameters used for flagging blocks are either the computable correction (CC), as represented by Equation (3.6), or the approximate error in the computable correction (ECC), as given by Equation (3.44). Appropriate norms of these error estimate contributions from each cell of the block are calculated and used to flag individual blocks for mesh adaptation. In order to provide a more linear scale for the resulting grid refinement criteria when flagging the grid blocks for refinement and coarsening, logarithmic values of the error estimates were used.

3.5 Error Estimation and *h*-Refinement Procedure

The proposed adjoint-based error estimation procedure and its use for driving the anisotropic mesh refinement can be summarized by the following steps:

1) An initially coarse body-fitted mesh consisting of one or more blocks is generated and the appropriate initial and boundary conditions are imposed.

2) The governing equations are solved on this base mesh using the inexact Newton's method for steady flows described in Section 2.2. The L_2 norm of the residual of one of the conserved quantities is monitored to determine whether a converged solution has been obtained. Once a solution is determined with a norm of the residual below a very small predefined tolerance level, a steady state solution is available on the current mesh and the value of the functional can then be calculated.

3) At this point, the mesh can be adapted in order to obtain a more accurate value of the functional. At first, the adjoint solution is calculated on the current mesh by solving Equation (3.9) using the GMRES algorithm. A limited least-squares piecewise linear reconstruction is performed on the coarse mesh in order to calculate profiles of both the primal and dual/adjoint solutions in each computational cell.

4) The entire mesh is isotropically and uniformly refined and the solution residuals are evaluated on this fine mesh. The total computable correction, as given in Equation (3.6), is then calculated by weighting these residuals with the prolonged adjoint solution. The mesh is coarsened back to the original mesh after obtaining the computable correction.

5) The entire mesh is uniformly refined in each of the three logical directions and the primal, conserved and adjoint variables are prolonged to the fine mesh using the second-order reconstruction performed earlier in step 3. These finer meshes are required in order to calculate the solution residual errors required for the estimation of the errors

in the functional considered.

6) The solution residuals are evaluated on the finer meshes. These residuals, as expected, are generally non-zero within the defined tolerance as the prolonged solution on the finer mesh does not correspond to the converged steady state solution. It is this difference between the fine and coarse grid residuals that is multiplied with the adjoint as shown in Equation (3.6), which acts as the sensitivity of the functional to the residual error.

7) The coarse computational mesh is then recovered and the primal flow quantities and the functional errors corresponding to the logical direction of refinement are projected back to the coarse grid. In this way, the error estimates in the value of the functional, associated with each grid cell for refinement in each of the preferred directions of the computational domain, are obtained.

8) Output-based error measures for each computational block are then obtained by calculating either the L_1, L_2 or L_∞ norm of the functional error contributions from the grid cells comprising the block. Note that these error measures for the block are just for the logical direction under consideration and measures for each of the three directions must be calculated similarly before the blocks can be flagged for refinement/coarsening.

9) For three-dimensional anisotropic refinement, steps 5-8 are carried out sequentially for all three logical directions to obtain error measures for all directions within each computational block. At this point, we have all the required information required to perform mesh adaptation on the base mesh. The blocks with error measures above a predefined threshold are flagged for refinement in the corresponding logical direction. Conversely, the blocks with error measures below a predefined threshold are flagged for coarsening in the corresponding logical direction. The mesh is adapted accordingly and a new mesh with a higher refinement level is generated.

10) This newly created mesh then becomes the base mesh and the entire process from steps 2-9 is carried out repeatedly until the desired maximum level of refinement is reached or the error estimate is reduced below a user-specified threshold.

The preceding procedure for the output-based anisotropic AMR is also illustrated in the form of a flowchart given in Figure 3.1.

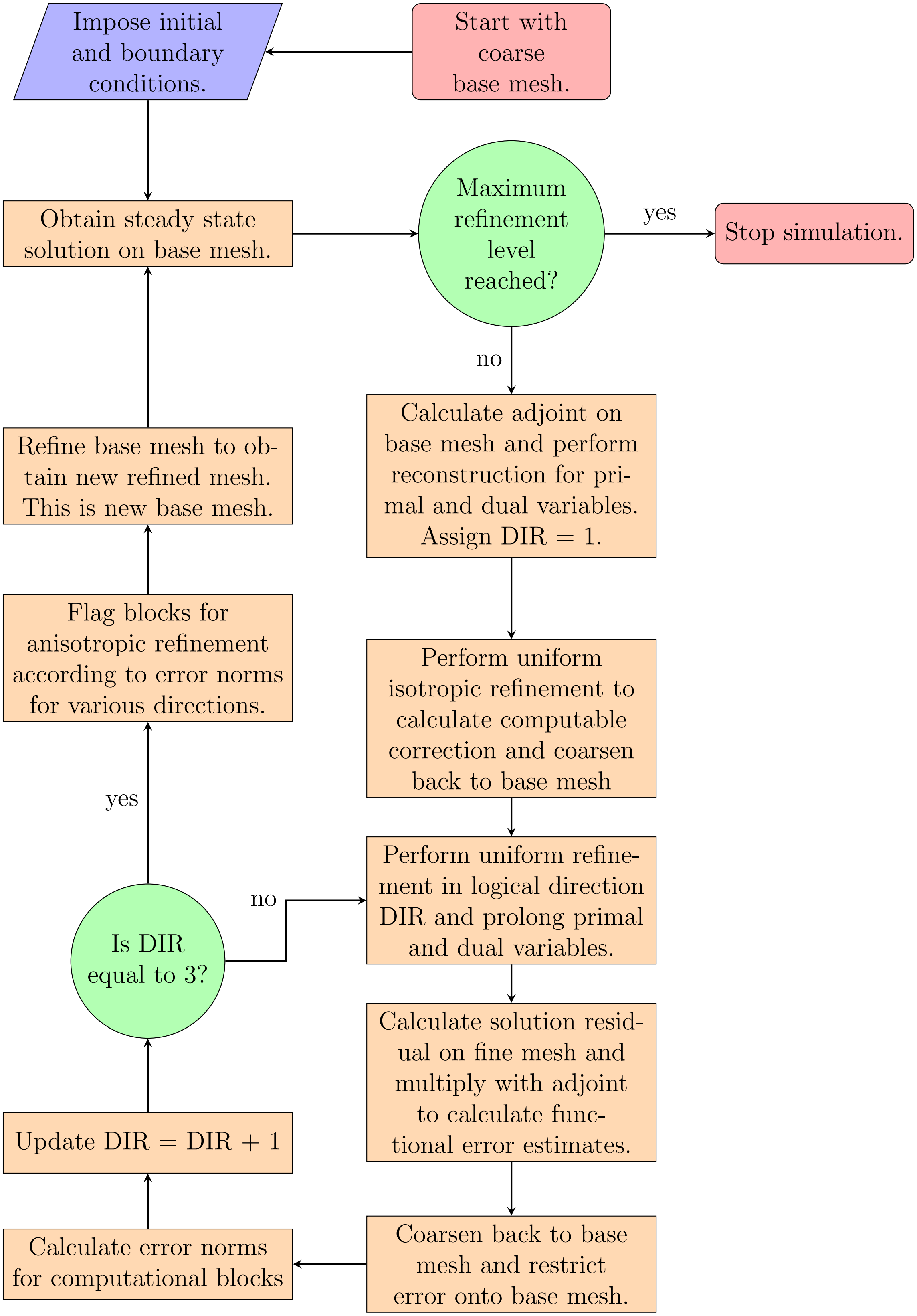

Impose initial and boundary conditions.
Start with coarse base mesh.
Obtain steady state solution on base mesh.
Maximum refinement level reached?
yes
Stop simulation.
no
Calculate adjoint on base mesh and perform reconstruction for primal and dual variables. Assign DIR = 1.
Refine base mesh to obtain new refined mesh. This is new base mesh.
Flag blocks for anisotropic refinement according to error norms for various directions.
Perform uniform isotropic refinement to calculate computable correction and coarsen back to base mesh
yes
no
Is DIR equal to 3?
Perform uniform refinement in logical direction DIR and prolong primal and dual variables.
Calculate solution residual on fine mesh and multiply with adjoint to calculate functional error estimates.
Update DIR = DIR + 1
Calculate error norms for computational blocks
Coarsen back to base mesh and restrict error onto base mesh.

Figure 3.1: Flowchart depicting procedure used for output-based error estimation and h-refinement.

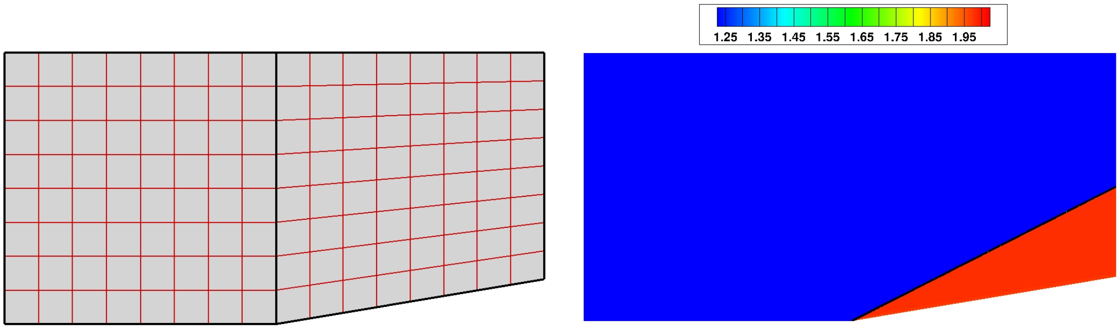
Chapter 4

Output-Based AMR for Steady Non-Conducting Aerodynamic Flows

In this chapter, the application of the proposed output-based anisotropic AMR scheme to several steady non-conducting aerodynamic flows is considered. Comparisons are made between the results obtained using the proposed output-based refinement approach to those obtained using both uniform mesh refinement and traditional gradient-based AMR approaches. The mesh savings obtained using the proposed output-based scheme for given levels of accuracy are investigated. The qualitative differences in the meshes obtained using the various approaches are also noted. Section 4.1 describes the application of the proposed scheme to inviscid flows governed by the Euler equations. Section 4.2 describes the application of the proposed scheme to viscous flows governed by the Navier-Stokes equations.

4.1 Inviscid Flows

Prior to investigating non-conducting viscous laminar flow cases, the application of the proposed output-based strategy to a number of inviscid cases is considered. The cases considered include inviscid supersonic flow past a wedge, inviscid supersonic flow past a diamond-shaped airfoil, inviscid supersonic flow over a smooth sinusoidal bump in a channel, and inviscid supersonic spherical outflow. The flow problems considered present a range of challenges for the proposed output-based AMR scheme ranging from the treatment of continuous supersonic outflows to supersonic flows with both isolated and complex interacting shock structures.



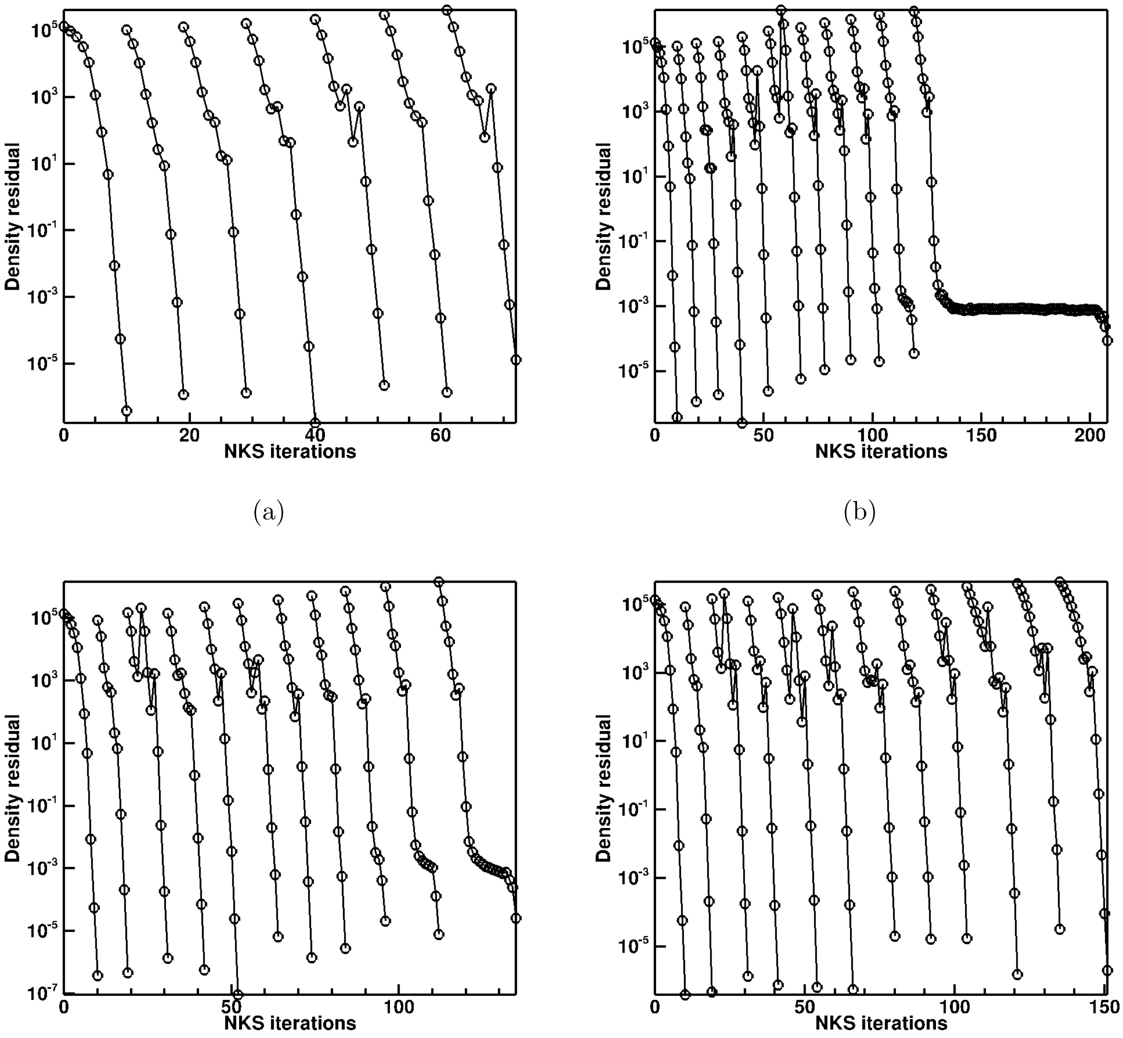
(a)

(b)

Figure 4.1: (a) Initial computational mesh for steady inviscid $M = 3$ supersonic flow past a wedge. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 2, the total number of cells per block is 8×8×4=256 and the total number of cells is 512. (b) Predicted density contours for this case after 6 uniform refinements. Blocks: 8,192. Cells: 2,097,152.

4.1.1 Supersonic Flow Past a Wedge

The first inviscid flow case considered here is that studied previously by Hartmann and Houston [210]. In particular, prediction of an oblique shock that forms when an inviscid supersonic flow of air enters a channel and is subsequently deflected by an inclined wedge is examined. An incoming supersonic flow of air with Mach number $M = 3$, standard atmospheric density $\rho = 1.225$ kg/m^3 and standard atmospheric pressure $p = 101,325$ Pa is considered. The length of the wedge is $l = 1$ m, the width is $w = 0.25$ m and the wedge angle α is taken to be 9.5°. The initial mesh, as shown in Figure 4.1(a), is composed of just two blocks along the length of the channel and the number of cells per block is $8 \times 8 \times 4 = 256$, with 4 cells in the out-of-plane direction. Reflective boundary conditions are imposed at the top and bottom boundaries. At the inlet of the channel, the flow variables for the incoming supersonic flow are kept fixed whereas at the outlet, constant extrapolation is imposed for the outgoing supersonic flow. The ratio of specific heats, γ, is assumed to be 1.4. Using the Rankine-Hugoniot relations, the angle of the shock and the analytical solution can be determined for a given wedge angle, α, and Mach number, M. For $\alpha = 9.5°$, the angle of the shock is given by $\beta = 26.9308°$. The analytical solution at the upstream and downstream sides of the shock in terms of ρ, p



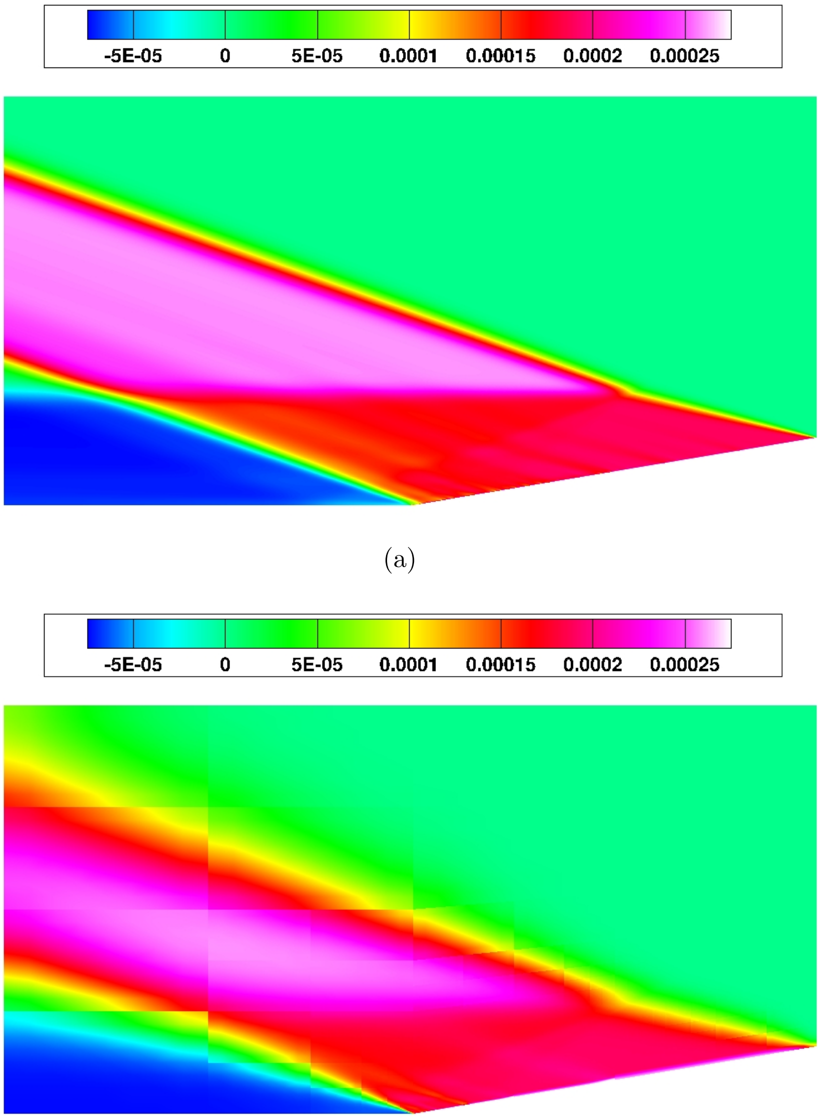
(a)

(b)

(c)

(d)

Figure 4.2: Density residual convergence history on the complete set of adaptive refined AMR meshes for steady inviscid $M = 3$ supersonic flow past a wedge: (a) uniform refinement; (b) gradient-based refinement; (c) output-based refinement using CC; (d) output-based refinement using ECC.



(a)

(b)

Figure 4.3: Energy adjoint distribution for steady inviscid $M = 3$ supersonic flow past a wedge : (a) after 7 uniform refinements. Blocks: 32,768. Cells: 8,388,608 (b) after 10 output-based refinements using CC as refinement criterion. Blocks: 1,323. Cells: 823,808.

and M is given by

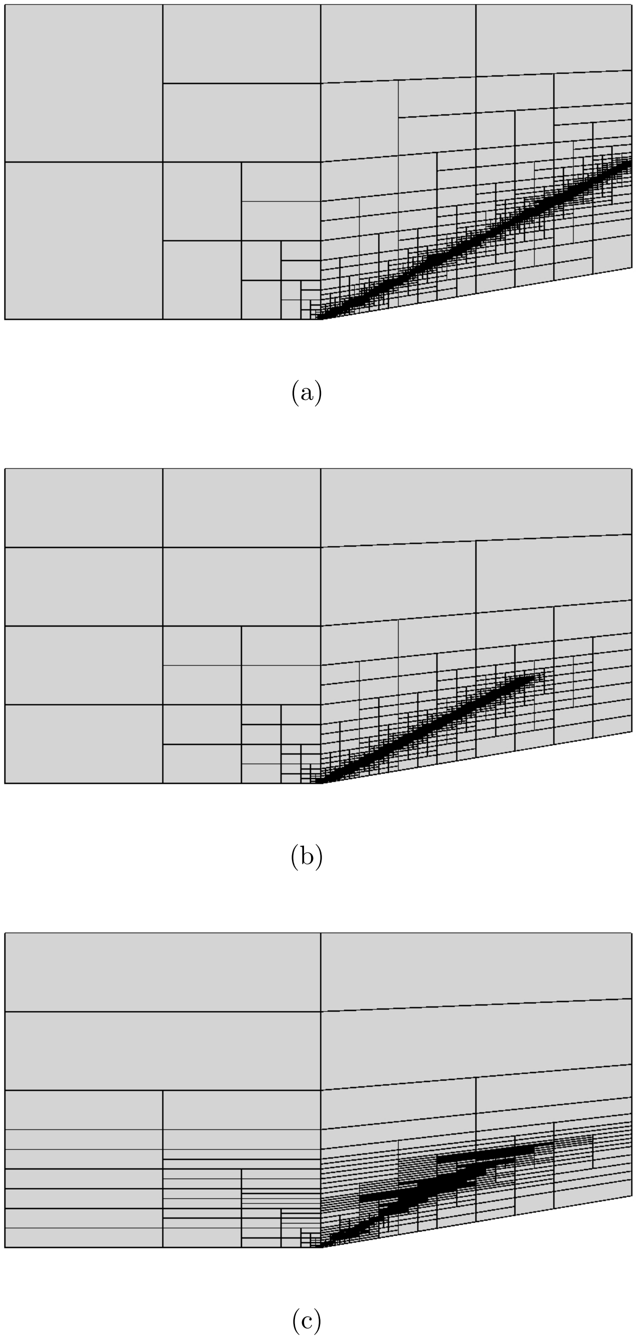
$$\begin{bmatrix}\rho\\ p\\ M\end{bmatrix}_{\text{left}} = \begin{bmatrix}1.225\ \text{kg/m}^3\\ 101325\ \text{Pa}\\ 3\end{bmatrix}, \begin{bmatrix}\rho\\ p\\ M\end{bmatrix}_{\text{right}} = \begin{bmatrix}1.98\ \text{kg/m}^3\\ 201354.51\ \text{Pa}\\ 2.53\end{bmatrix}. \tag{4.1}$$

Figure 4.1(b) shows the predicted density distribution for this case after a converged solution has been obtained after 6 uniform refinements on the initial mesh. The convergence of the steady-state solution residual for this case for density is shown in Figure 4.2 for the complete set of refined meshes for the four refinement techniques considered (i.e., uniform, gradient-based, as well as CC and ECC output-based refinement of the mesh). From Figure 4.2, it is evident that rapid convergence of the steady-state solution is achieved using the proposed NKS scheme in less than 40 Newton steps on all grids with a residual reduction of more than eight orders of magnitude. Similar convergence behaviour was achieved for all of the non-conducting flow cases considered in the thesis.

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 $l = 1$ m and width $w = 0.25$ m, the analytical value of the drag force is given by $D_a = p_{\text{right}} \cdot w \cdot l \cdot \sin(\alpha) = 8308.27$ N. Figure 4.3 shows the distribution for the 5^{th} component of the adjoint solution vector after a converged solution has been obtained after 7 uniform refinements on the initial mesh and after 10 refinements using CC as refinement criterion. Additionally, Figure 4.4 shows a comparison between meshes having approximately the same functional error, obtained after several adaptive refinements on the initial mesh, using various mesh refinement criteria. For comparison purposes, Figure 4.4(a) shows the mesh obtained using density gradients to drive the mesh adaptation whereas Figures 4.4(b) and 4.4(c) show the meshes obtained using the two proposed output-based refinement criteria considered here, the CC and ECC methods, respectively.

It is evident from Figure 4.4 that the gradient-based method refines the entire region containing the oblique shock from the start of the ramp all the way to the outflow boundary where the flow exits the domain. In contrast, the output-based method using the CC as the refinement criteria refines only the portion of the shock lying in the region where the adjoint-variable has a non-zero value. The output-based method using the ECC approach as the refinement criteria also refines the portion of the shock lying in the region where the adjoint-variable has a non-zero value. However, as can be seen in Figure 4.4(c), the shock is not highly refined for the latter as compared to the CC results of Figure 4.4(b), with the highest refinement level only observed in a very few blocks located in the vicinity of the compression corner where the shock originates and where the functional of interest is evaluated.

The difference between the numerically computed value of the drag force on the wedge and the analytical value calculated above provides the error in the drag force. Figures 4.5(a) and 4.5(b) show plots of this percentage error and the actual value of the drag force against the number of degrees of freedom per direction for this effectively

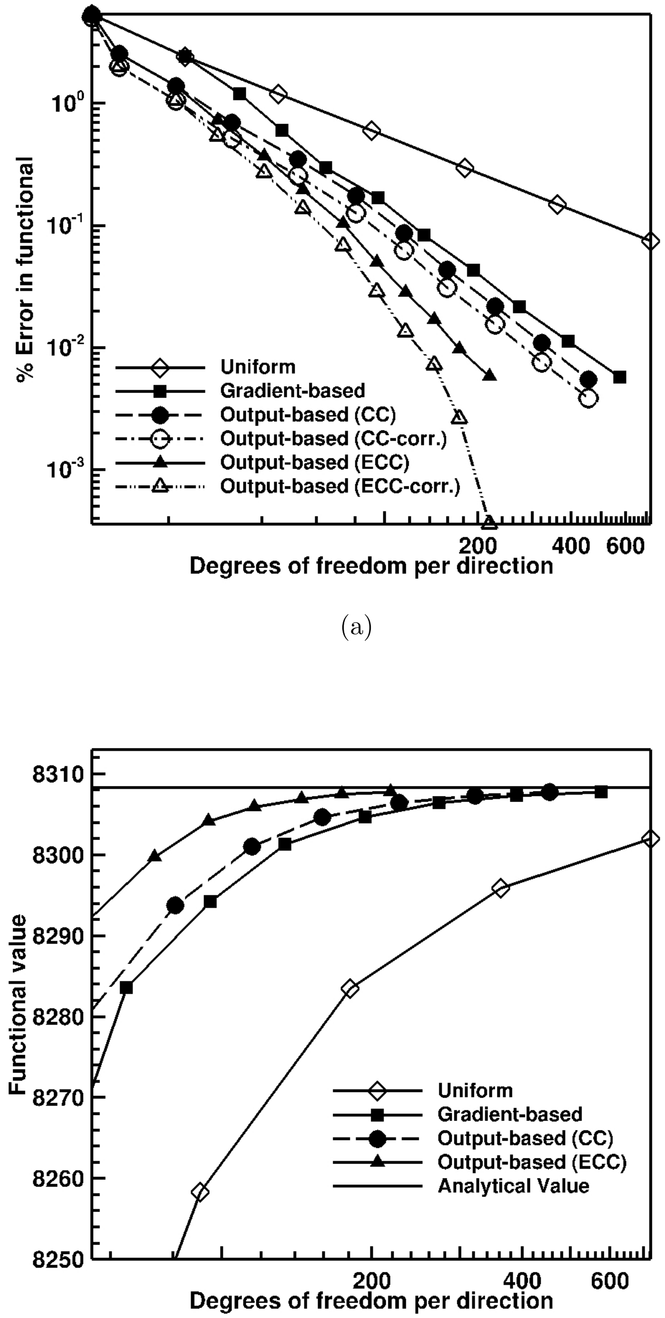


(a)

(b)

(c)

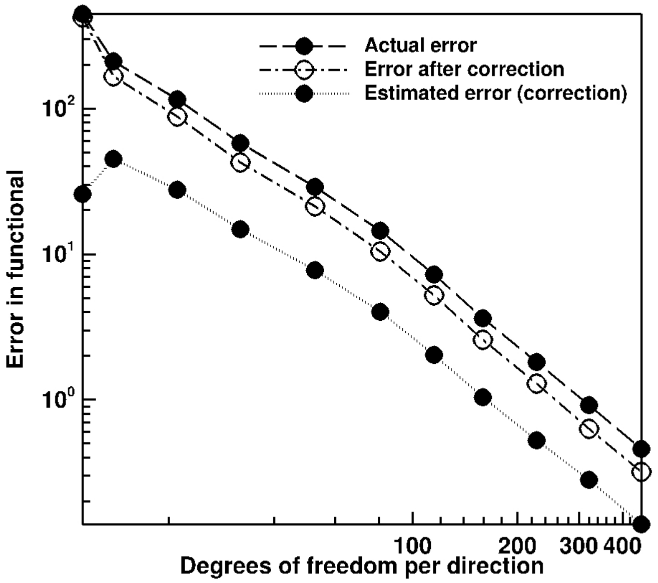
Figure 4.4: Adapted meshes showing the grid block boundaries (not individual cells) having almost the same functional error, for steady inviscid $M = 3$ supersonic flow past a wedge: (a) After 10 refinements using density gradient as refinement criterion. Blocks: 5,171. Cells: 1,323,776. (b) After 10 refinements using CC as refinement criterion. Blocks: 3,218. Cells: 823,808. (c) After 11 refinements using ECC as refinement criterion. Blocks: 743. Cells: 190,208.



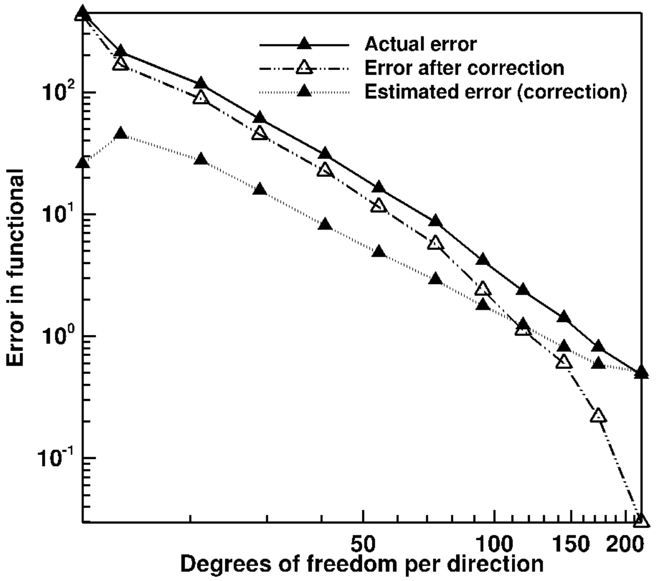
(a)

(b)

Figure 4.5: (a) Convergence of the percentage error in the predicted drag force and (b) convergence of the predicted drag force, as a function of the mesh size (number of degrees of freedom per direction) for various refinement techniques for steady inviscid $M = 3$ supersonic flow past a wedge.



(a)



(b)

Figure 4.6: Convergence of the actual error, the error after adding the computable correction and the error estimate (absolute computable correction) as a function of the mesh size (number of degrees of freedom per direction) (a) for the CC method and (b) for the ECC method, for steady inviscid $M = 3$ supersonic flow past a wedge.

two-dimensional flow problem, i.e. $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells in the computational mesh, respectively. The convergence of the error is shown for uniform as well as the gradient-based and both forms of the output-based mesh refinement techniques. For the output-based techniques, the error obtained after adding the computable correction to the functional is also shown in Figure 4.5(a) whereas the solid horizontal line in the closeup of Figure 4.5(b) represents the exact value of the functional. From these figures it is evident that the output-based method using CC achieves a mesh size reduction of 38% in terms of the total number of computational cells at the finest refinement level as compared to the gradient-based method for the same accuracy level (∼0.006%). A mesh size reduction of 86% is achieved by using the ECC to drive the mesh refinement. Adding the computable correction, as illustrated in Figure 4.5(a), further reduces the error in the functional. While the gradient-based approach provides a more accurate global flow solution compared to uniform refinement for the same mesh density, it focuses too much attention to the reflected shock, excessively refining the upper shock which does not improve the calculation of the pressure drag on the wedge as efficiently as the output-based approaches. This behaviour clearly illustrates the benefits of the output-based AMR.

In order to illustrate more clearly the convergence and behaviour of the output-based methods, the actual error in the functional, the corrected error, and the estimated error (computable correction) are all shown in Figure 4.6 as a function of the number of degrees of freedom for both the CC and ECC methods. For both these output-based refinement strategies, the error measures all continuously decrease as the mesh is refined and the estimated error after the computable correction is applied is significantly smaller than the uncorrected result. For both the CC and ECC methods, it can also be observed that the values of the actual error and estimated error based on the computable correction approach each other as the mesh is refined, as should be expected.

4.1.2 Supersonic Flow Past a Diamond-Shaped Airfoil

In the second inviscid flow example case considered here, inviscid supersonic flow is intercepted by a symmetrical diamond-shaped airfoil at a non-zero angle of attack. A Mach number $M = 2$ flow of air at standard atmospheric density $\rho = 1.225$ kg/m^3 and standard atmospheric pressure $p = 101,325$ Pa is considered. The chord length of the airfoil is assumed to be unity and the far-field boundary is located 32 chord lengths away from the airfoil in a radially outward direction. The initial mesh, as shown in Figure 4.7(a), is composed of 8 blocks and the number of cells per block is $10 \times 10 \times 4 = 400$, with 10

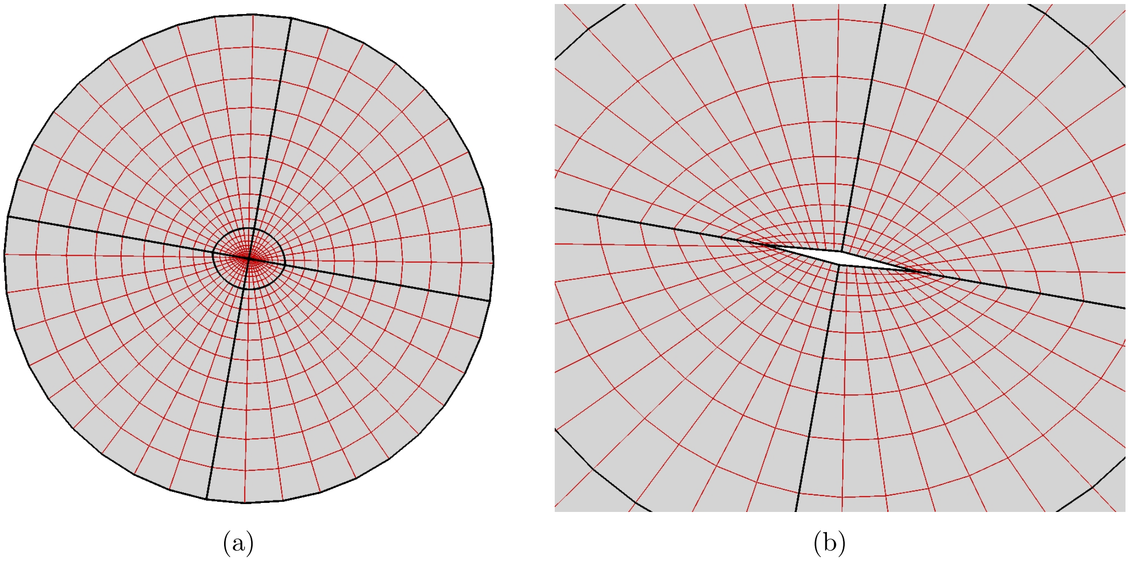


Figure 4.7: (a) Initial computational mesh for steady inviscid $M = 2$ supersonic flow past a diamond-shaped airfoil showing: (a) entire view of the mesh and (b) close-up view of the mesh. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 8, the total number of cells per block is 10×10×4=400 and the total number of cells is 3,200.

cells in the direction along the airfoil surface, 10 cells in the direction radially outward from the airfoil surface and 4 cells in the out-of-plane direction. Mesh stretching in the radial direction is used such that the grid cells are more clustered near the airfoil surface, as shown in a close-up view of the initial mesh in Figure 4.7(b). 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. Reflective boundary conditions are imposed at the airfoil surface. At the outer boundary, fixed boundary conditions are imposed on the left half of the boundary for the incoming supersonic flow while constant extrapolation is imposed on the right half. The ratio of specific heats, γ, is assumed to be 1.4. 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. Figure 4.8 shows the predicted density distribution around the diamond-shaped airfoil after a converged solution has been obtained after 6 uniform refinements on the initial mesh.

The functional used in this case is the pressure drag force on the airfoil in the direction of the free-stream flow. The flow solution at the airfoil surface can be calculated using

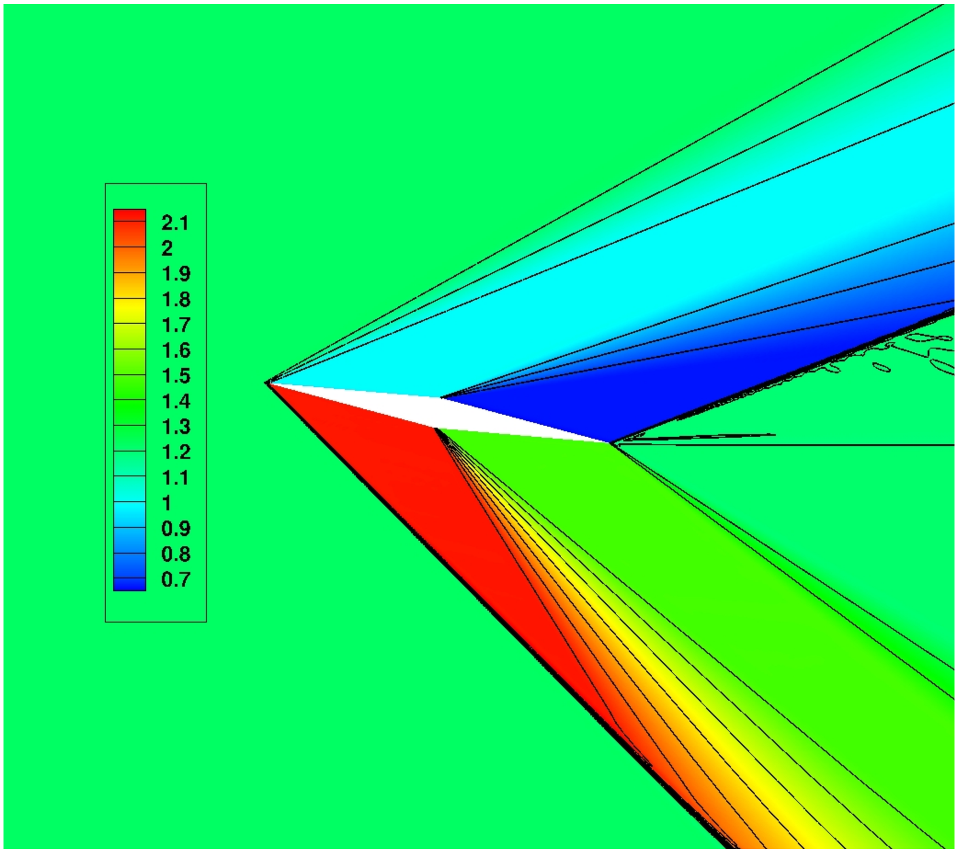


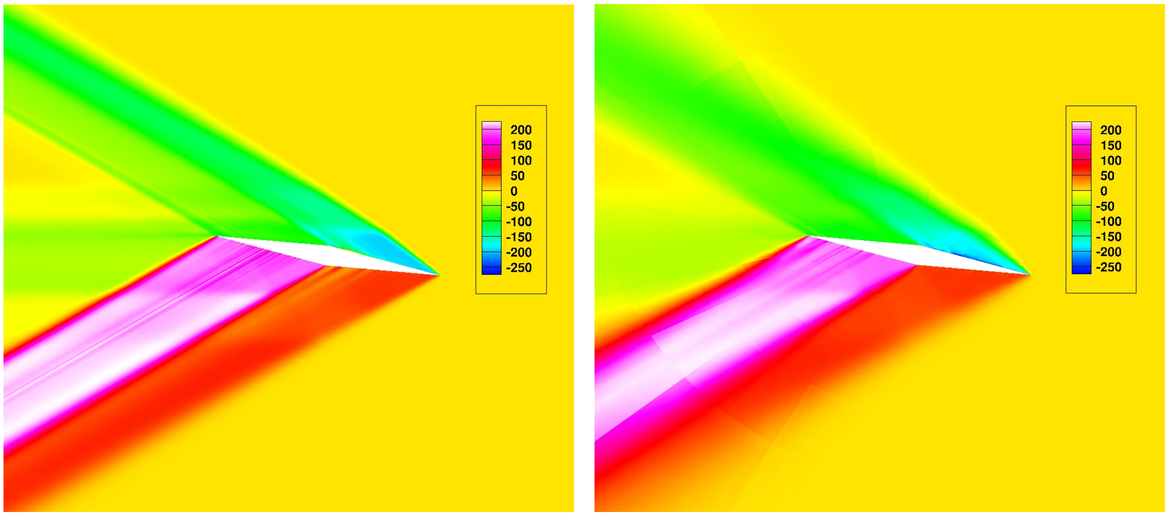
Figure 4.8: Predicted density contours for steady inviscid $M = 2$ supersonic flow past a diamond-shaped airfoil after 6 uniform refinements. Blocks: 32,768. Cells: 13,107,200.

the Rankine-Hugoniot relations across the oblique shocks and isentropic flow relations across the Prandtl-Meyer expansion fans. Thus, an analytical value of the drag force can be calculated. For a unit chord length, the drag coefficient is given by

$$C_D = \frac{2D}{\rho_\infty V_\infty^2}, \tag{4.2}$$

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

Figure 4.9 illustrates the distributions of the 1^{st} component of the adjoint solution vector after converged solutions have been obtained after 6 uniform refinements of the initial mesh as well as following 8 refinements based on the CC refinement criterion, respectively. Additionally, Figure 4.10 shows the resulting computational meshes obtained after several adaptive refinements on the initial mesh, using the various mesh refine-



(a) (b)

Figure 4.9: Density adjoint distribution for steady inviscid $M = 2$ supersonic flow past a diamond-shaped airfoil : (a) after 6 uniform refinements. Blocks: 32,768. Cells: 13,107,200. (b) after 8 refinements using CC as refinement criterion. Blocks: 2,324. Cells: 929,600.

ment strategies considered herein. In particular, Figures 4.10(a) and 4.10(b) show the meshes obtained after 7 and 8 refinements using the density gradient to drive the mesh adaptation, respectively. Figures 4.10(c) and 4.10(d) show the meshes obtained after 8 refinements using the CC refinement criteria and after 7 refinements using ECC criteria, respectively. As for the previous wedge flow case examined having a strong shock, the gradient-based method can be seen to produce extensive refinement of the mesh in the vicinities of the leading and trailing edge shocks. Additionally, the gradient-based method also refines the regions containing the four expansion fans just above the leading edge, just below the trailing edge and at the central portion of the airfoil just above and below the surface near the sharp corners of the diamond airfoil. Since the airfoil surfaces abruptly change direction as opposed to a smooth airfoil, the flow properties also change abruptly across the expansion fans, leading to very sharp gradients. The output-based refinement method employing the CC strategy as the refinement criteria generously refines the expansion fans formed at the leading edge and above and below the airfoil. The leading edge shock is also refined but only in the region where the adjoint variable has a non-zero value. The output-based refinement method employing the ECC as the refinement criteria refines in the same regions, however, in a relatively milder manner. Also, the refinement in the direction along the airfoil surface is similar to that using the

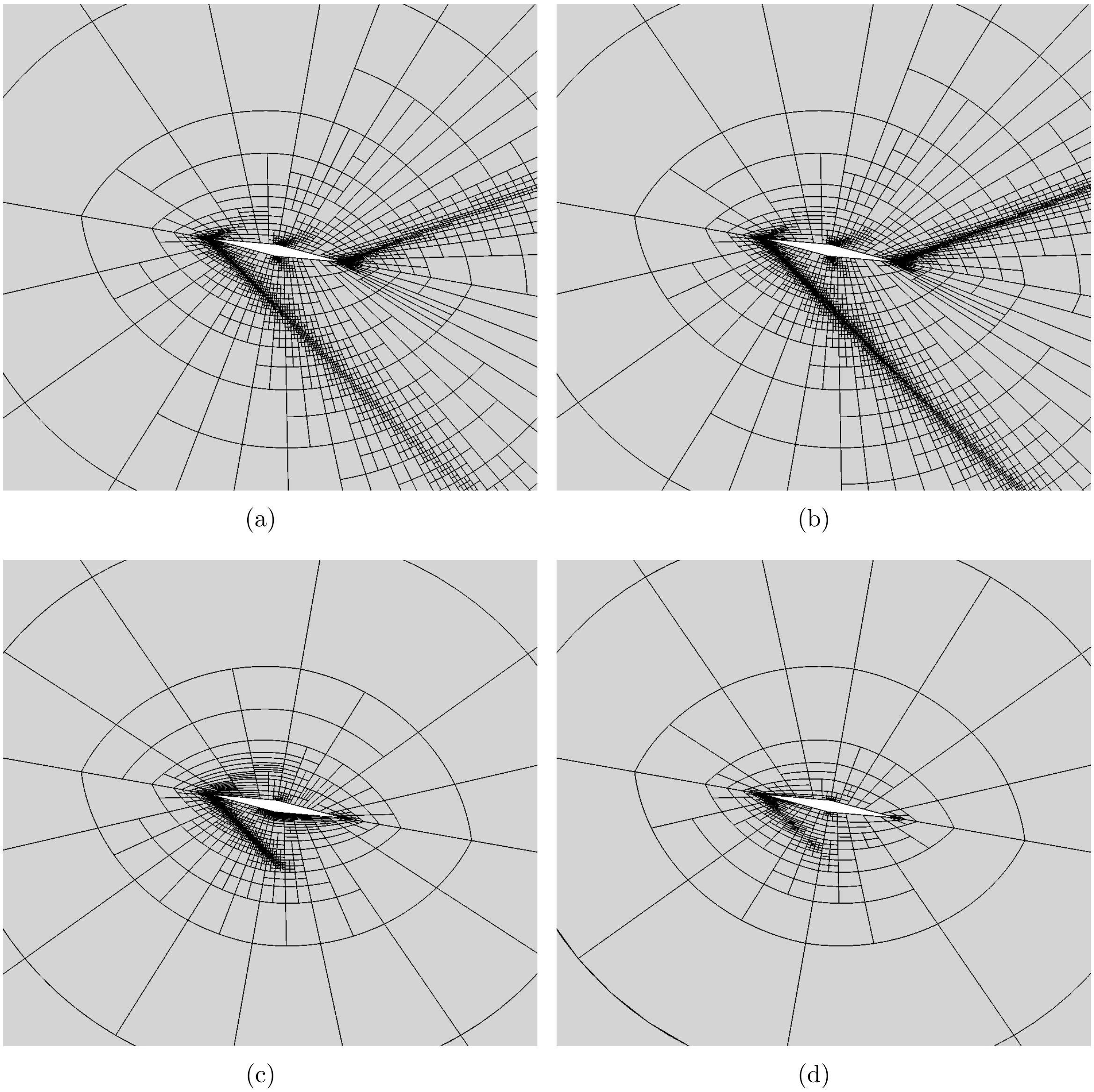
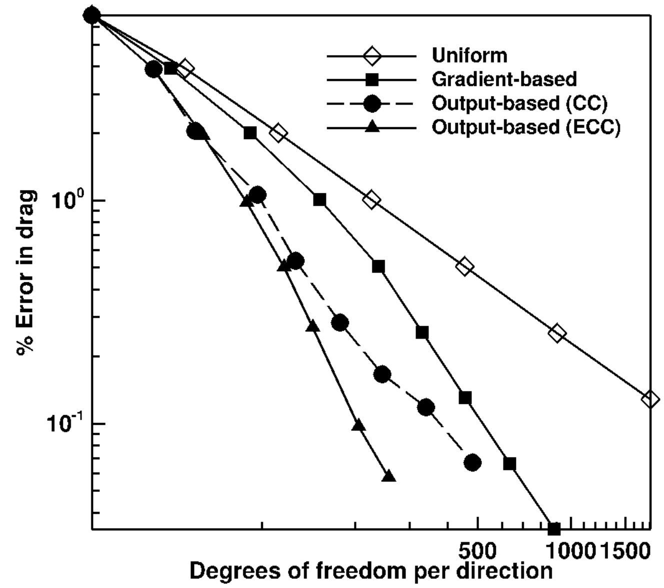


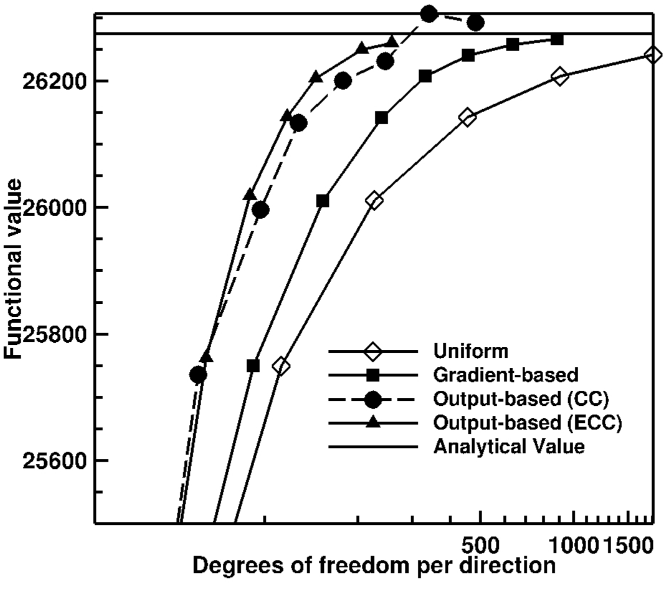
Figure 4.10: Adapted meshes showing the grid block boundaries (not individual cells) for steady inviscid $M = 2$ supersonic flow past a diamond-shaped airfoil: (a) After 7 refinements using density gradient as refinement criterion. Blocks: 4,012. Cells: 1,604,800. (b) After 8 refinements using density gradient as refinement criterion. Blocks: 7,774. Cells: 3,109,600. (c) After 8 refinements using CC as refinement criterion. Blocks: 2,324. Cells: 929,600. (d) After 7 refinements using ECC as refinement criterion. Blocks: 661. Cells: 264,400. Meshes shown in (a), (c) and (d) have approximately the same functional error.

CC method, but the refinement in the outward direction normal to the airfoil is more sparse.

The difference between the numerically computed value of the drag coefficient and the



(a)



(b)

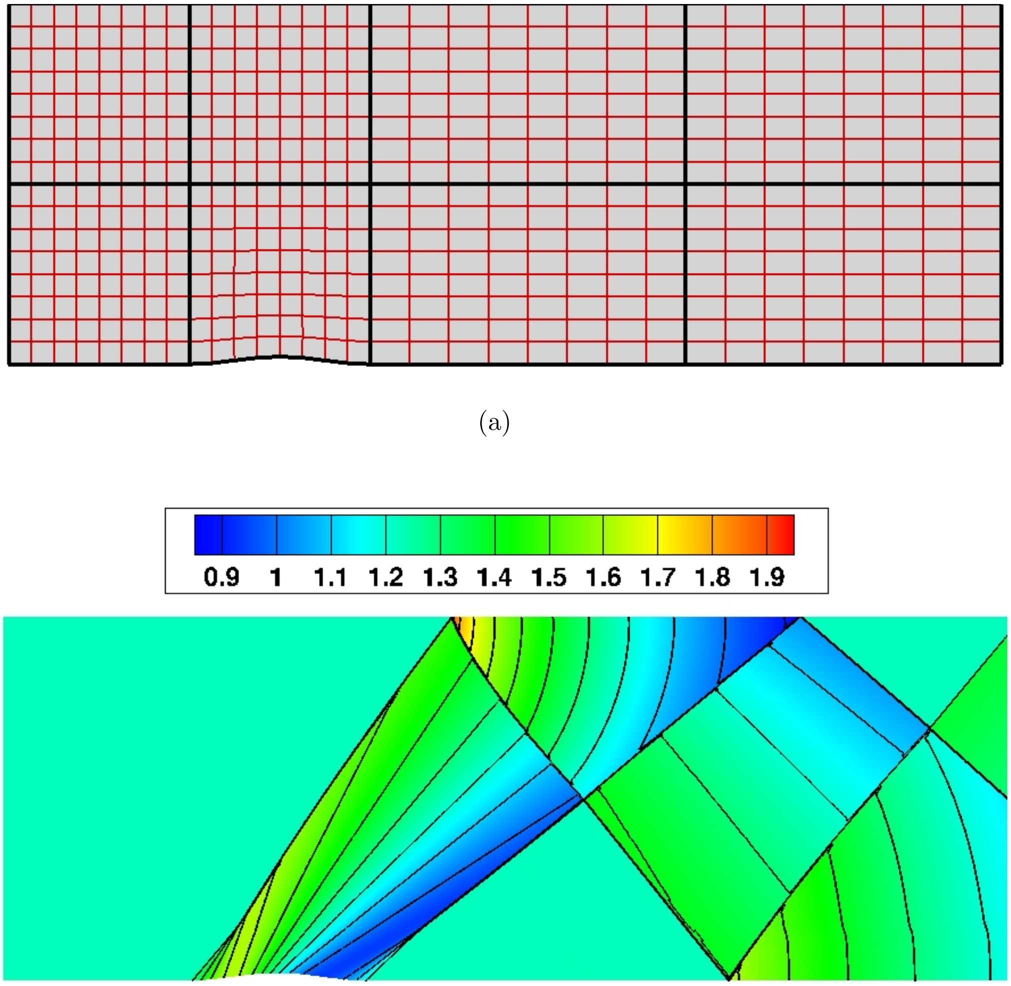
Figure 4.11: (a) Convergence of the percentage error in the predicted drag force and (b) convergence of the predicted drag force, as a function of the mesh size (number of degrees of freedom per direction) for various refinement techniques for steady inviscid $M = 2$ supersonic flow past a diamond-shaped airfoil.

analytical value, 0.0926, is the error in the drag coefficient. Figures 4.11(a) and 4.11(b) show plots of this error as a function of the total number of degrees of freedom per direction for this two-dimensional flow problem, i.e. $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells in the computational mesh, respectively. The convergence of the functional error is shown for uniform as well as the gradient-based and both output-based mesh refinement techniques. As for the previous wedge flow case, it is evident that all four of these approaches tend to an almost constant rate of convergence in the error after the first few refinements, with the output-based methods showing the steepest descent. While it is clear that the density gradient-based method performs better than the uniform approach, use of the CC refinement criterion achieves a 42% reduction in the size of the computational mesh as compared to the density gradient-based approach for the same accuracy ($\sim$0.07%), whereas with the ECC approach provides an 83% reduction in mesh size for the same accuracy, both of which are quite considerable.

4.1.3 Supersonic Flow Over a Bump in a Channel

In the next inviscid flow problem, a Mach number $M = 1.4$ supersonic flow of air with standard atmospheric density $\rho = 1.225$ kg/m^3 and standard atmospheric pressure $p = 101,325$ Pa enters a rectangular channel and is intercepted by a sinusoidal bump. The profile equation of the sinusoidal bump is $y_1(x) = 0.042 \sin^2(\pi x)$ m for $x \in [0, 1]$ m, the height of the bump thus being 4.2 cm. The length of the channel is 5.5 m and its height is 2 m. The bump is located at a distance of 1 m from the inlet along the length of the channel. The initial mesh, as shown in Figure 4.12(a), is composed of 4 blocks along the length of the channel and 2 blocks along its height and the number of cells per block is $8 \times 8 \times 4 = 256$, with 4 cells in the out-of-plane direction. Reflective boundary conditions are imposed on the top and bottom boundaries of the channel including the bump. At the inlet, flow variables for the incoming supersonic flow are kept fixed whereas at the outlet, constant extrapolation is imposed for the outgoing supersonic flow. The ratio of specific heats, γ, is assumed to be 1.4. The presence of the bump results in the formation of compression waves in front of the bump that coalesce to form a strong oblique shock. Similarly, a strong oblique shock is also formed near the trailing edge of the bump. These shocks subsequently reflect from both the upper and lower boundaries. This case was studied previously by Freret and Groth [1]. Figure 4.12(b) shows the predicted density distribution for this case after a converged solution has been obtained after 6 uniform refinements were applied to the initial mesh.

The functional used here is the pressure-induced drag force on the bump in the di-

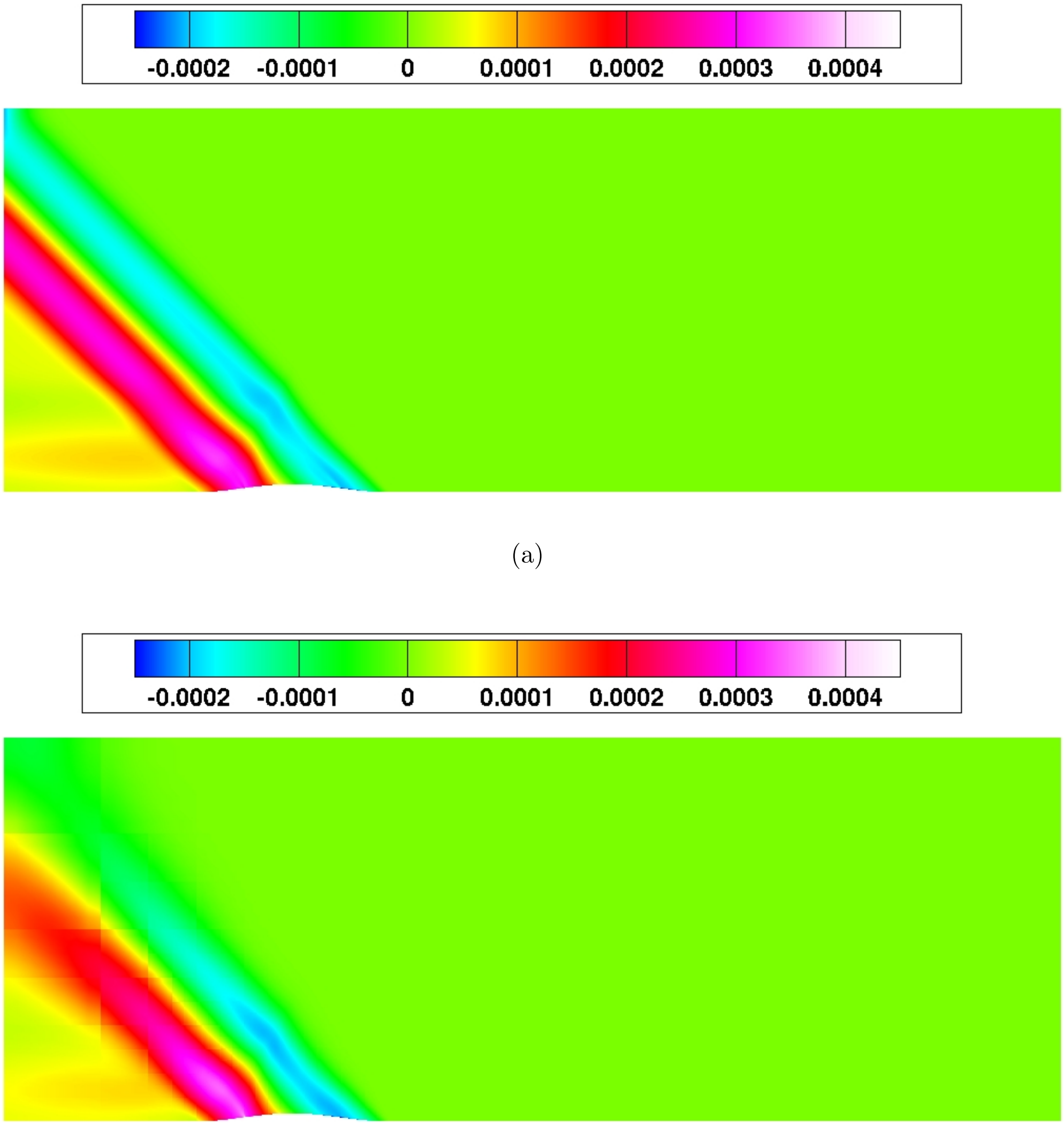


(a)

(b)

Figure 4.12: (a) Initial computational mesh for steady inviscid $M = 1.4$ supersonic flow over a bump. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 8, the total number of cells per block is 8×8×4=256 and the total number of cells is 2,048. (b) Predicted density contours for this case after 6 uniform refinements. Blocks: 32,768. Cells: 8,388,608.

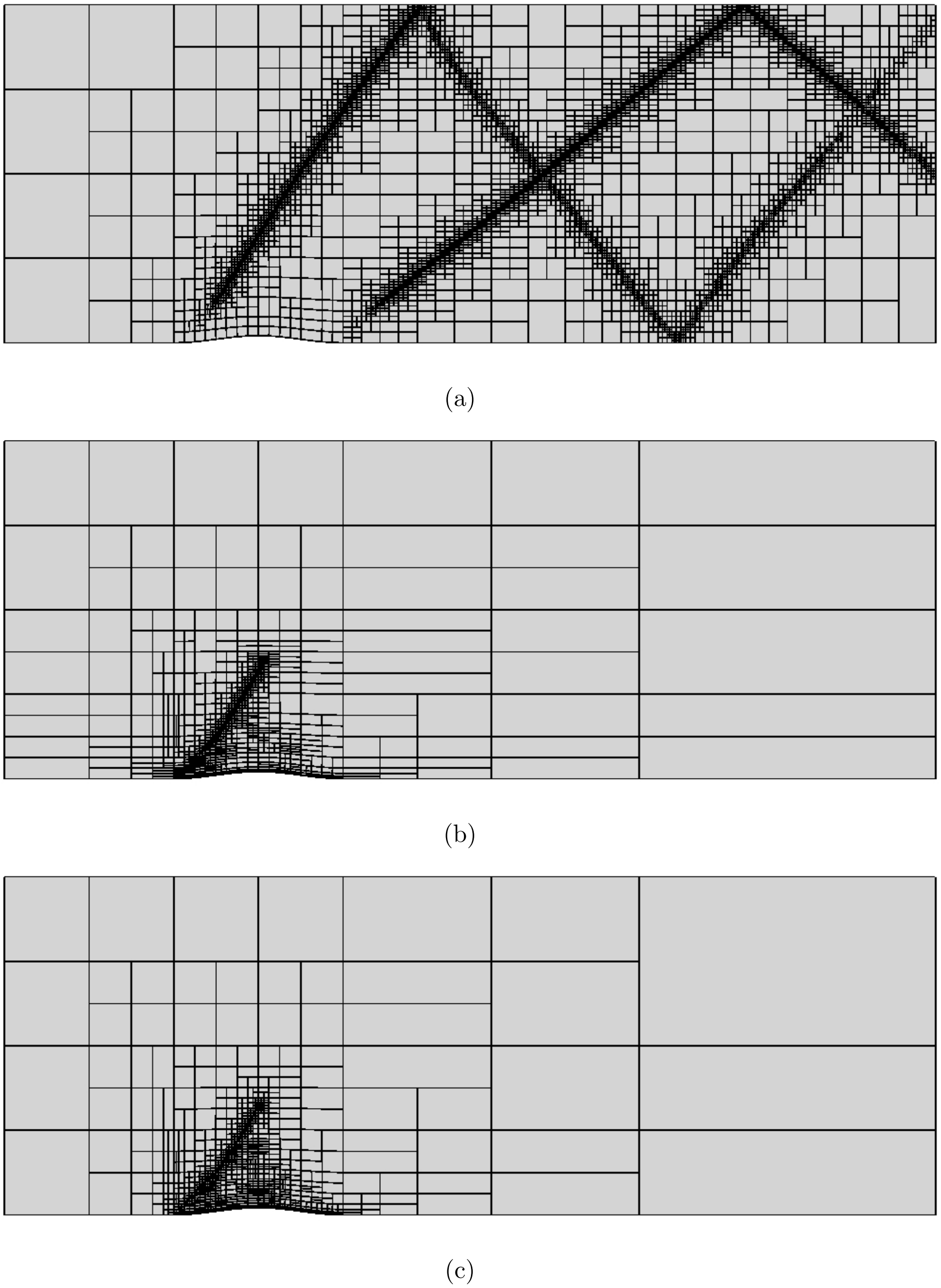
rection of the incoming flow. Figure 4.13 shows the computed distributions for the 5^{th} component of the adjoint solution vector for this functional after a converged solution has been obtained following 6 uniform refinements of the initial mesh and after 7 refinements baed on the CC refinement criterion. Again for comparison purposes, Figure 4.14 depicts the meshes obtained after 7 adaptive refinements of the initial mesh, using the various mesh refinement criteria. Figure 4.14(a) shows the mesh obtained using density gradients to drive the mesh adaptation. Figures 4.14(b) and 4.14(c) show the meshes



(b)

Figure 4.13: Energy adjoint distribution for steady inviscid $M = 1.4$ supersonic flow over a bump : (a) after 6 uniform refinements. Blocks: 32,768. Cells: 8,388,608. (b) after 7 refinements using CC as refinement criterion. Blocks: 971. Cells: 248,576.

obtained using the two output-based refinement criteria, the CC and ECC approaches, respectively. The gradient-based method refines both the shocks emanating from the leading edge as well as the trailing edge of the bump, as well as their reflections from the top and bottom boundaries of the channel. The output-based method employing the CC method as the refinement criteria refines only a small portion of the leading shock lying in

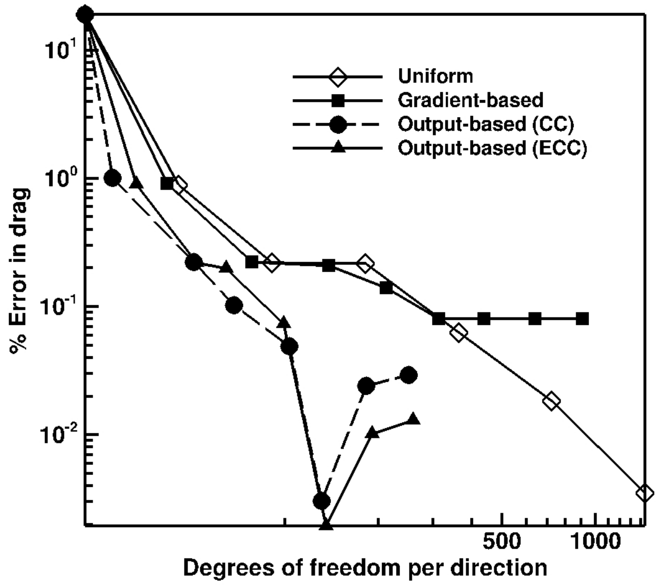


(a)

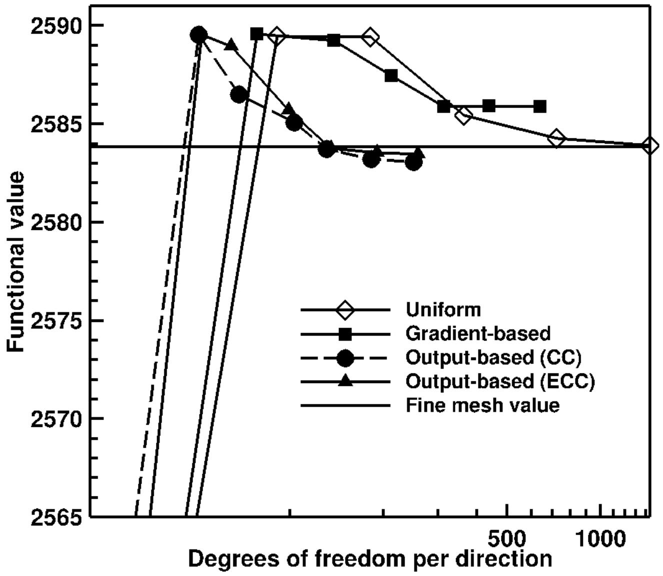
(b)

(c)

Figure 4.14: Adapted meshes showing the grid block boundaries (not individual cells) for steady inviscid $M = 1.4$ supersonic flow over a bump: (a) After 7 refinements using density gradient as refinement criterion. Blocks: 6,392. Cells: 1,636,352. (b) After 7 refinements using CC as refinement criterion. Blocks: 971. Cells: 248,576. (c) After 7 refinements using ECC as refinement criterion. Blocks: 1,038. Cells: 265,728.



(a)



(b)

Figure 4.15: (a) Convergence of the estimated percentage error in the predicted drag force as a function of the mesh size (number of degrees of freedom per direction) for steady inviscid $M = 1.4$ supersonic flow over a bump. (b) Convergence of the value of the predicted drag force for various refinement techniques for the bump flow.

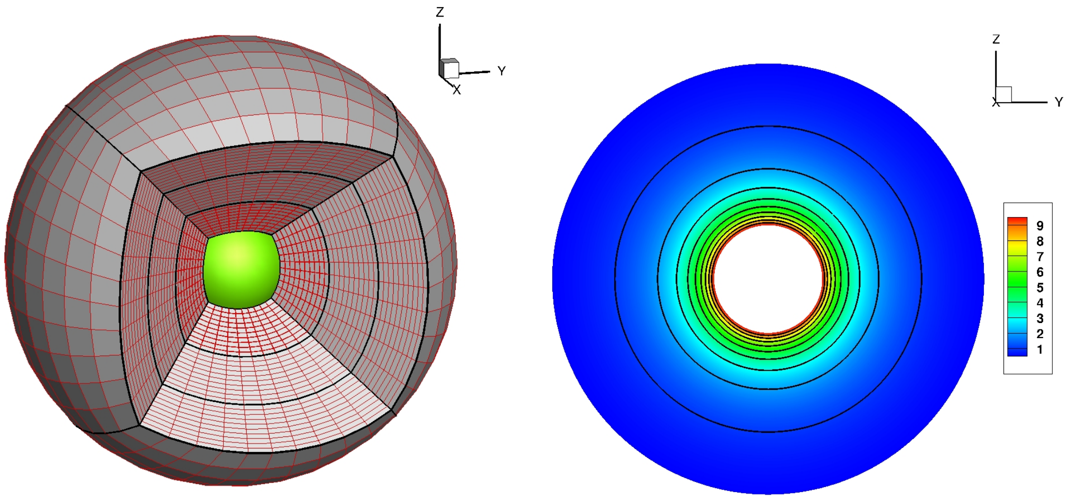
the region where the adjoint variable has a non-zero value. None of the shock reflections or the shock formed near the trailing edge of the bump undergo any refinement. The output-based method using the ECC criteria also exhibits a similar behaviour, although the amount of refinement is relatively more sparse.

An analytical value for the drag force is unavailable for this case. Thus, a value of the functional calculated on a uniformly refined mesh to one level higher than the highest mesh refinement level considered here is taken as the reference value of the pressure drag for the purposes of evaluating directly the solution error. Figure 4.15(a) shows a plot of the corresponding error in the predicted drag force value again as a function of the total number of degrees of freedom per direction, i.e. $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells in the computational mesh. For the output-based techniques, the corrected functional obtained after adding the computable correction to the functional is also shown. Additionally, a closeup of the convergence of the predicted drag force value for the various mesh refinement techniques is shown in Figure 4.15(b). For the latter, the solid horizontal line indicates the converged reference fine mesh value of the functional.

The gradient-based refinement strategy initially outperforms the uniform refinement strategy by continuously reducing the functional error, but after 4 refinements, it reaches a plateau and the uniform refinement strategy produces more accurate results. This is because this technique focuses on refining the shocks and their reflections, mostly in the region downstream from the bump. This region has no influence on the accuracy of the force on the bump, as can be seen from the distribution for the adjoint variable shown in Figure 4.13. Thus, the gradient-based approach converged to a rather inaccurate value of the functional, as can be seen in Figure 4.15(a). However, as for the supersonic wedge and diamond airfoil flow cases, the output-based refinement approaches continue to produce more accurate predictions of the functional as the mesh is further refined. In terms of mesh savings, these represent a more than 95% reduction in mesh size, as compared to a uniformly refined mesh having equivalent accuracy (∼0.01%).

4.1.4 Supersonic Spherical Outflow

The last inviscid flow case of interest here is that of steady supersonic spherical outflow. Air at a supersonic speed with flow density $\rho_i = 10$ kg/m^3, radial velocity $V_{r,i} = 4.5$ m/s, and pressure $p_i = 26$ Pa 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. A spherical computational domain is used for this case with a cubed-sphere multi-block mesh as described by Ivan *et al.* [67,69]. The initial mesh, as



(a)

(b)

Figure 4.16: (a) A portion of the initial computational mesh for steady spherical supersonic outflow. The thicker black lines are the block edges and the thinner red lines are the cell edges. The shaded green surface is the inner boundary. The total number of blocks is 18, the total number of cells per block is 10×10×10=1,000 and the total number of cells is 18,000. (b) Predicted density contours for this case on a sliced cross-section after 3 uniform refinements. Blocks: 9,216. Cells: 9,216,000.

partially shown in Figure 4.16(a), consists of eighteen blocks connected, forming an inner hollow sphere of radius $R_i = 1$ m and an outer spherical shell of radius $R_o = 4$ m. The number of cells per block is $10 \times 10 \times 10 = 1,000$. The flow variables for the incoming supersonic flow are fixed at the inner boundary with radius, R_i. At the outer boundary with radius, R_o, constant extrapolation is imposed for the exiting supersonic flow. The ratio of specific heats, γ, is assumed to be 1.4.

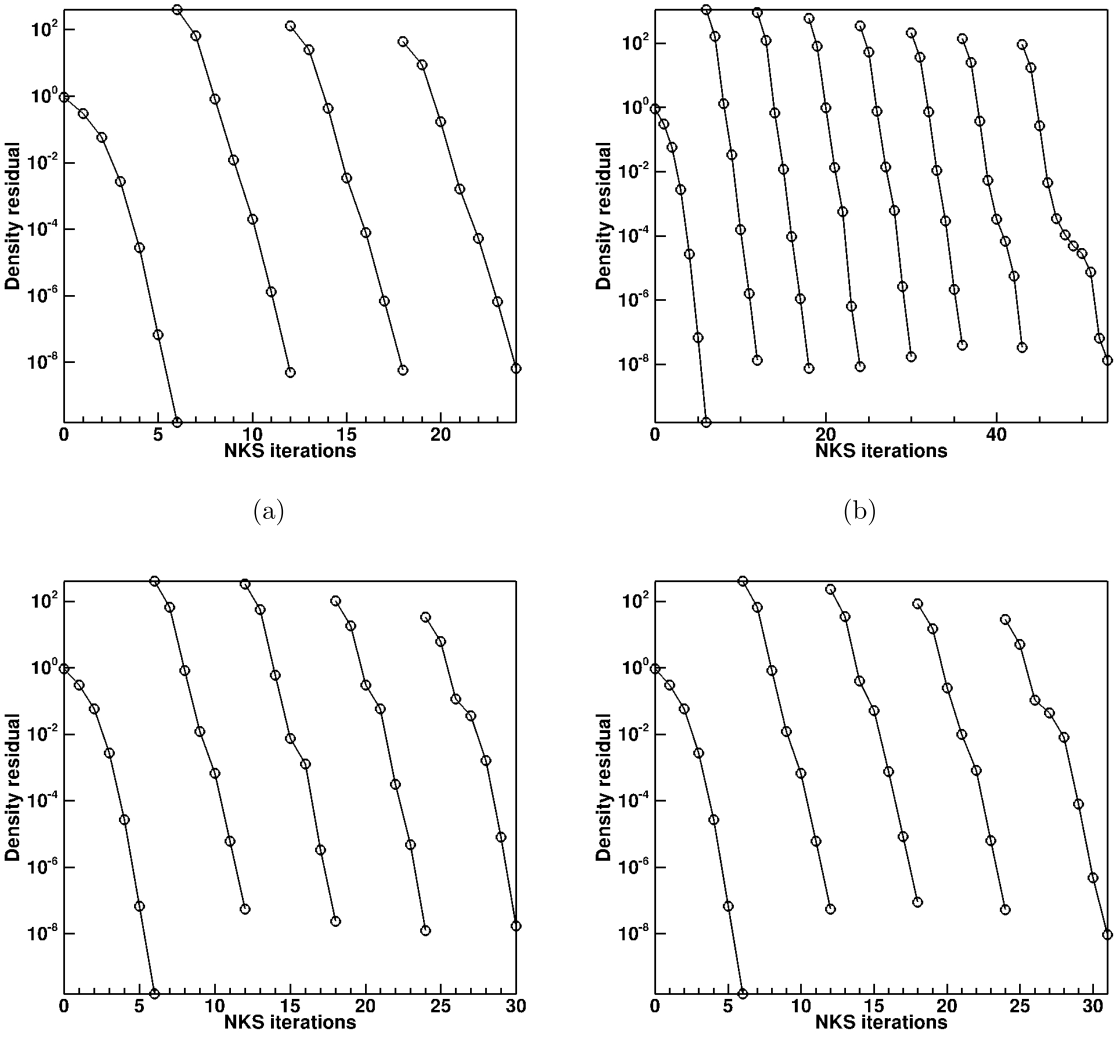
The analytical solution to the preceding supersonic outflow problem, which is also described by Ivan *et al.* [69], can be defined at any radial location, r, using

$$C_3 - \frac{1}{r^2 V_r \left[(C_2 - V_r^2)^{\frac{1}{\gamma-1}} \right]} = 0, \tag{4.3}$$

and

$$C_3 = \frac{1}{\left(\frac{2\gamma}{\gamma-1} \frac{p_i}{\rho_i}^{\frac{1}{\gamma-1}} \right) R_i^2 V_{r,i}}, \quad C_2 = \frac{2\gamma}{\gamma-1} \frac{p_i}{\rho_i} + V_{r,i}^2, \tag{4.4}$$

where C_2 and C_3 are constants depending on the inflow conditions. Here, p_i, ρ_i and $V_{r,i}$



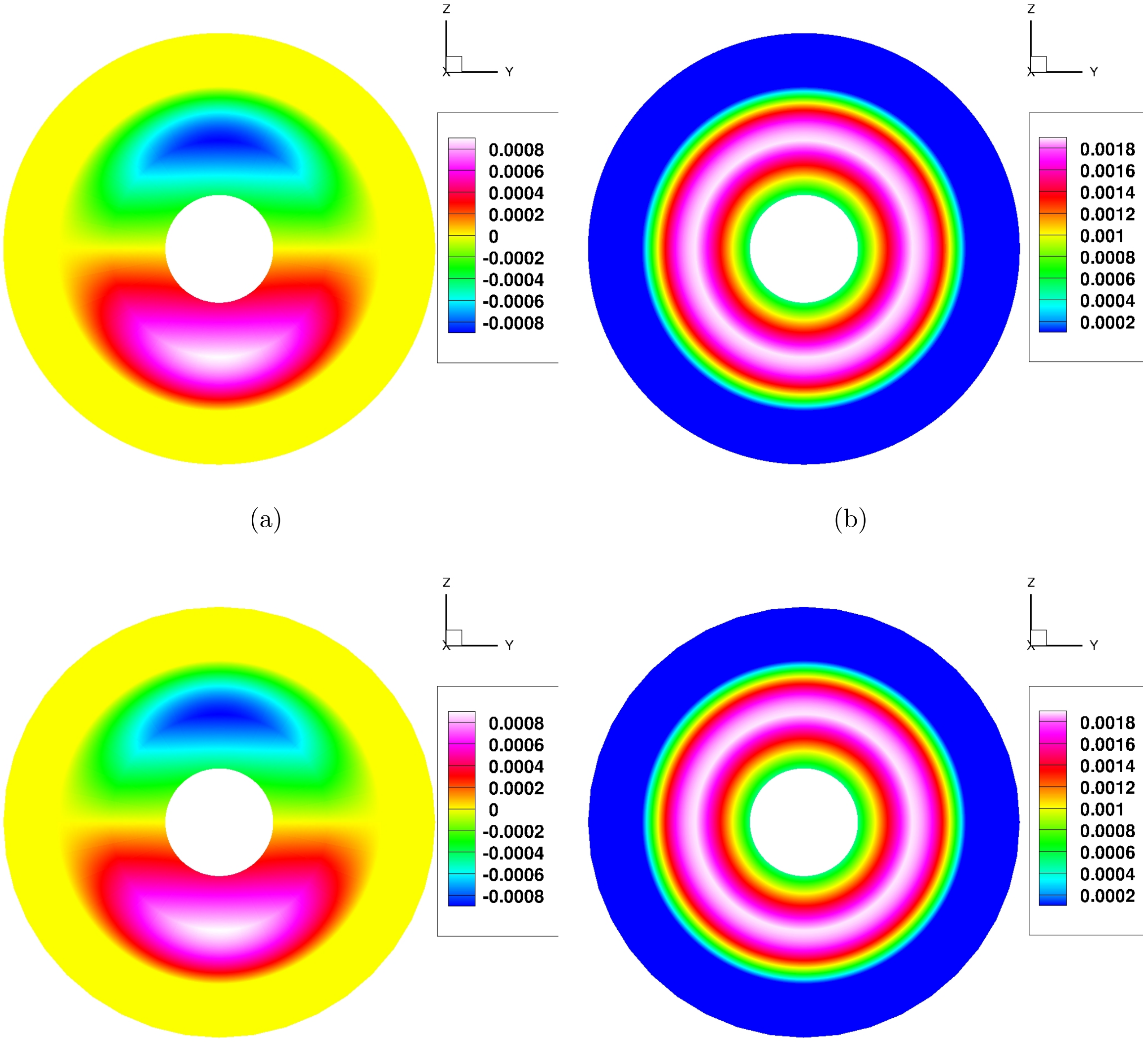
(a)

(b)

(c)

(d)

Figure 4.17: Density residual convergence history on the complete set of adaptive refined AMR meshes for steady inviscid supersonic spherical outflow: (a) uniform refinement; (b) gradient-based refinement; (c) output-based refinement using CC; (d) output-based refinement using ECC.



(a)

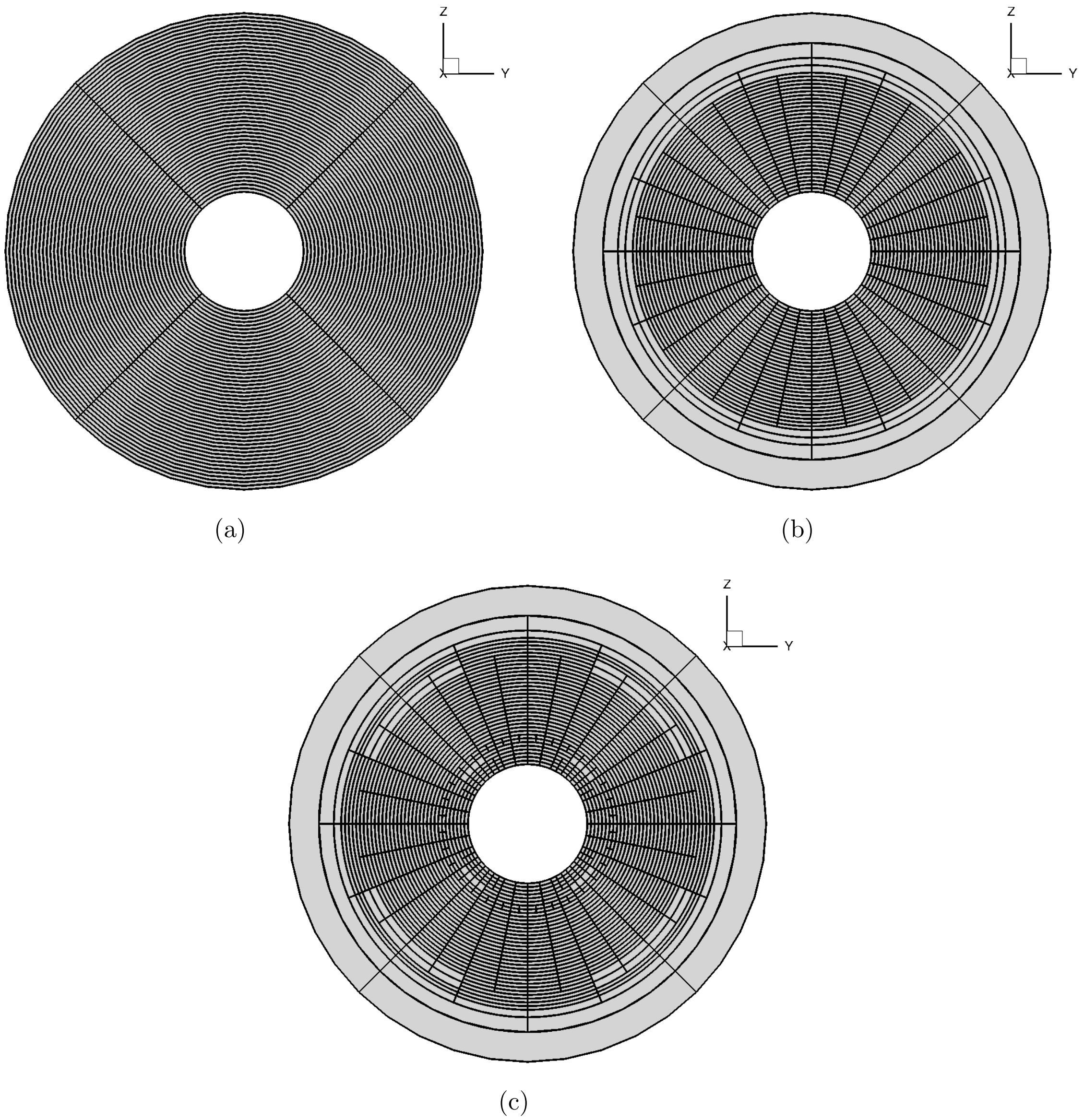
(b)

(c)

(d)

Figure 4.18: (a) The z-momentum and (b) density adjoint distributions for steady spherical supersonic outflow on a sliced cross-section after 3 uniform refinements. Blocks: 9,216. Cells: 9,216,000. (c) The z-momentum and (d) density adjoint distributions for steady spherical supersonic outflow on a sliced cross-section after 4 refinements using CC as refinement criterion. Blocks: 12,126. Cells: 12,126,000.

are the pressure, density and radial velocity at the inner boundary while V_r is the velocity at an arbitrary radius, r. For illustrative purposes, Figure 4.16(b) shows the predicted density distribution for this case for a portion of the spherical domain after a converged solution has been obtained following 3 uniform refinements on the initial mesh. As for the wedge flow case, the convergence of the steady-state solution density residual for this

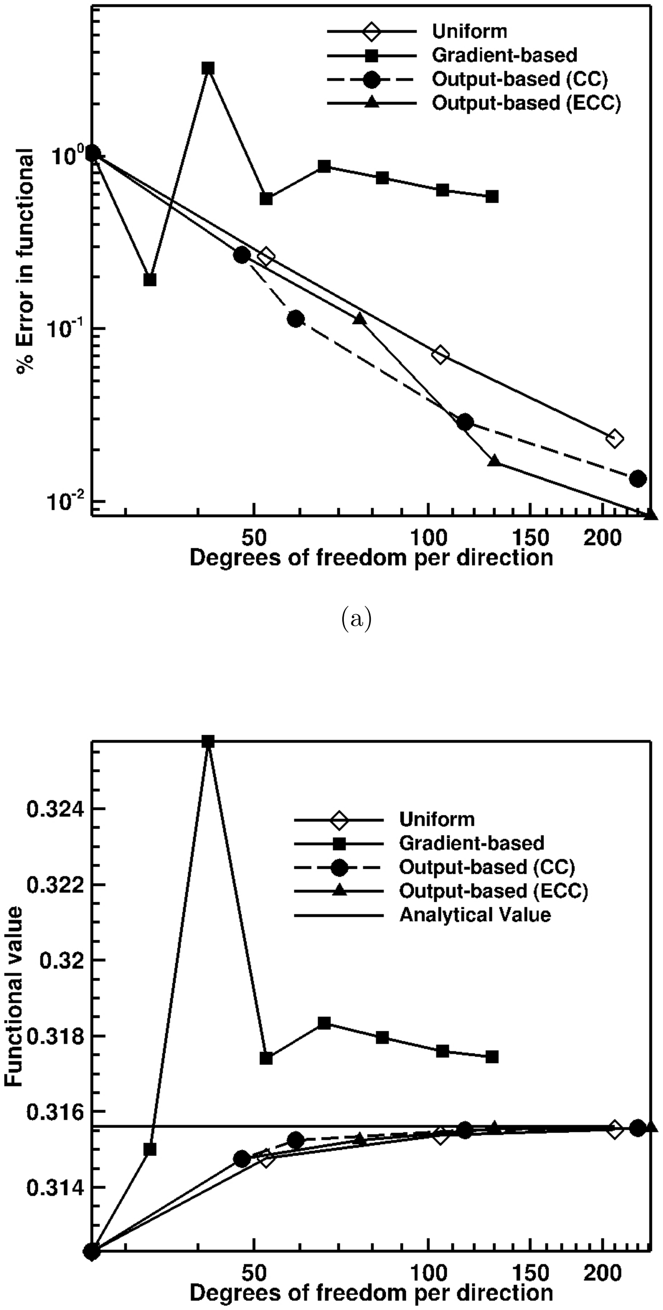


(a)

(b)

(c)

Figure 4.19: Sliced cross-sections of adapted meshes showing the grid block boundaries (not individual cells) for steady spherical supersonic outflow: (a) After 4 refinements using density gradient as refinement criterion. Blocks: 288. Cells: 288,000. (b) After 4 refinements using CC as refinement criterion. Blocks: 12,126. Cells: 12,126,000. (c) After 4 refinements using ECC as refinement criterion. Blocks: 14,216. Cells: 14,216,000.



(a)

(b)

Figure 4.20: (a) Convergence of the percentage error in the functional and (b) convergence of the functional, as a function of the mesh size (number of degrees of freedom per direction) for various refinement techniques for steady inviscid supersonic spherical outflow.

case is also given in Figure 4.17 for the complete set of refined meshes obtained using the four refinement techniques considered herein. As for the wedge flow case, it is evident from Figure 4.17 that rapid convergence of the steady-state solution is achieved using the proposed NKS scheme in less than 20 Newton steps on all grids with a residual reduction of more than eight orders of magnitude.

The functional adopted for the supersonic outflow problem is the volume-integrated temperature of the air contained in the portion of the spherical domain contained between the radii of 2 m and 3 m. The analytical value of the functional, i.e. the temperature integrated over this volume can be calculated analytically and is 0.31560688 m^{3}K. Figure 4.18 shows a cross-sectional view of the distributions of the 4^{th} and 1^{st} components of the adjoint solution vector after a converged solution has been obtained after 3 uniform refinements on the initial mesh and following 4 output-based refinements using the CC refinement criterion. Additionally, Figure 4.19 shows a comparison of meshes obtained after several adaptive refinements on the initial mesh, using gradient-based and output-based refinement criteria. Figure 4.19(a) shows a slice of the mesh obtained after 6 adaptive refinements using density gradients as the adaptation criteria. This technique mainly detects gradients in the radial direction and hence the anisotropic refinement is restricted to that direction. Figures 4.19(b) and 4.19(c) show a slice of the mesh obtained after 4 output-based adaptive refinements using the CC and ECC strategies, respectively as the adaptation criteria. These methods are able to detect the errors in the functional arising in the azimuthal direction and therefore refine the mesh in both radial and azimuthal directions in the region of the functional calculation, including directions upwind from this region.

The difference between the numerically computed value of the functional and the analytical value, 0.31560688 m^{3}K, provides the true error in the functional. Figures 4.20(a) and 4.20(b) show a plot of this percentage error against the degrees of freedom per direction for this now three-dimensional problem, i.e. $\sqrt[3]{N_{\text{cells}}}$, where N_{cells} is the total number of grid cells in the computational mesh. In particular, Figure 4.20(b) shows a plot of the convergence in the value of the functional to the analytical value. The convergence of the error is shown for uniform as well as the gradient-based and both output-based mesh refinement techniques. The uniform refinement approach leads to a constant rate of reduction in error with a slope 2, as expected for smooth flows simulated with second-order schemes. It is quite evident that the gradient-based approach is not able to detect the dominant errors in the azimuthal directions and, thus, is not able to reduce the functional error. However, significant reductions in mesh size ($> 80\%$) are achieved using the output-based methods, as compared to the uniform approach for the same order of

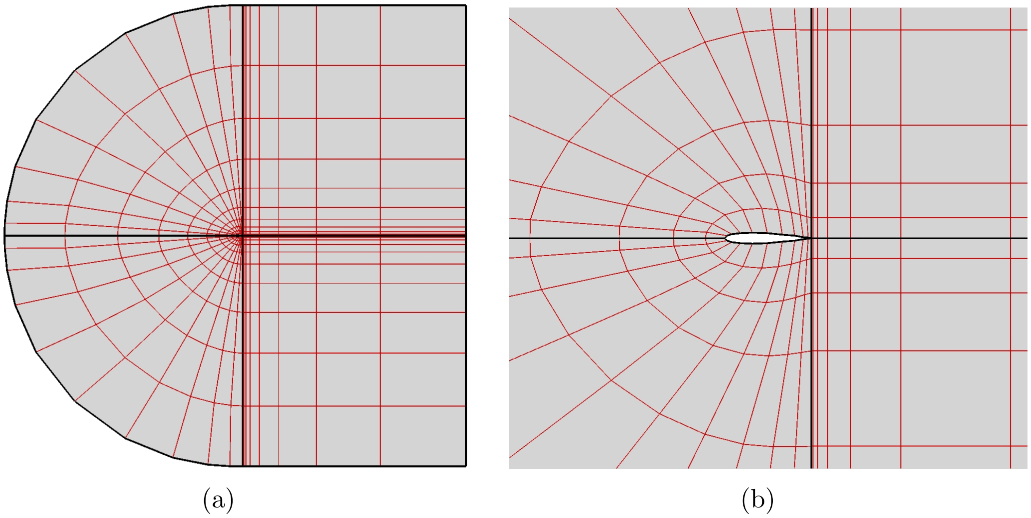


Figure 4.21: Initial computational mesh for steady subsonic viscous flow past a NACA0012 airfoil showing: (a) entire view of the mesh and (b) close-up view of the mesh. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 4, the total number of cells per block is 10×10×4=400 and the total number of cells is 1,600.

accuracy (∼0.03%). Among the two output-based techniques, the ECC method shows more isotropic refinement near the inner boundary.

4.2 Viscous Flows

The application of the proposed output-based strategy to non-conducting viscous laminar flow cases is considered next. The two cases considered are the subsonic and supersonic laminar flows of air past a NACA0012 airfoil at a zero angle of attack. The solution for the former is smooth and continuous whereas the latter has a bow shock that is detached from the leading edge of the airfoil and will present additional challenges to the proposed output-based AMR method.

4.2.1 Subsonic Flow Past a NACA0012 Airfoil

In the first viscous flow case considered here, a horizontal viscous subsonic flow with a Mach number $M = 0.5$ and a Reynolds number $Re = 5,000$ is intercepted by a NACA0012 airfoil at an angle of attack given by $\alpha = 0°$. A C-shaped computational grid with stretching initially composed of 4 blocks, as shown in Figure 4.21, is used here. The stretching is required for efficient capturing of the viscous boundary layer formed

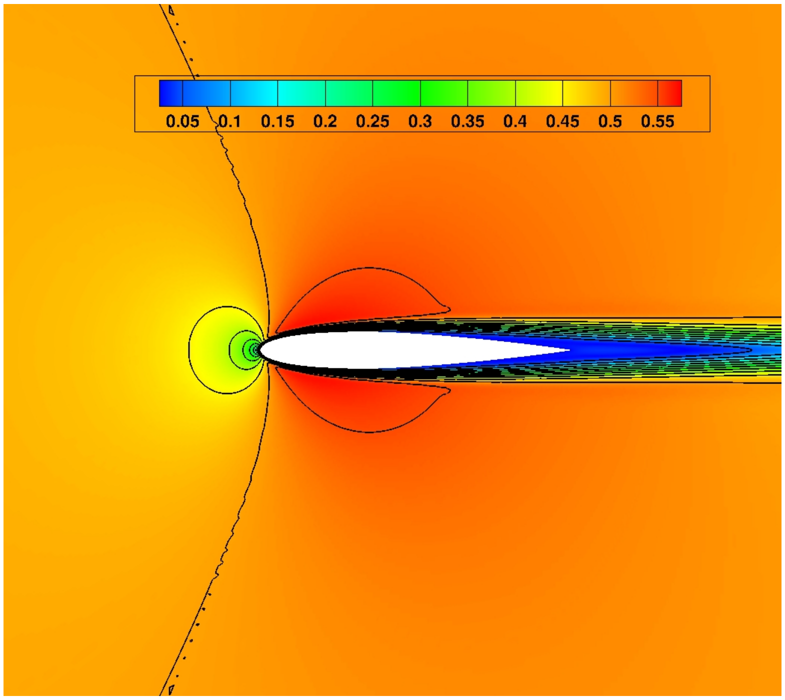
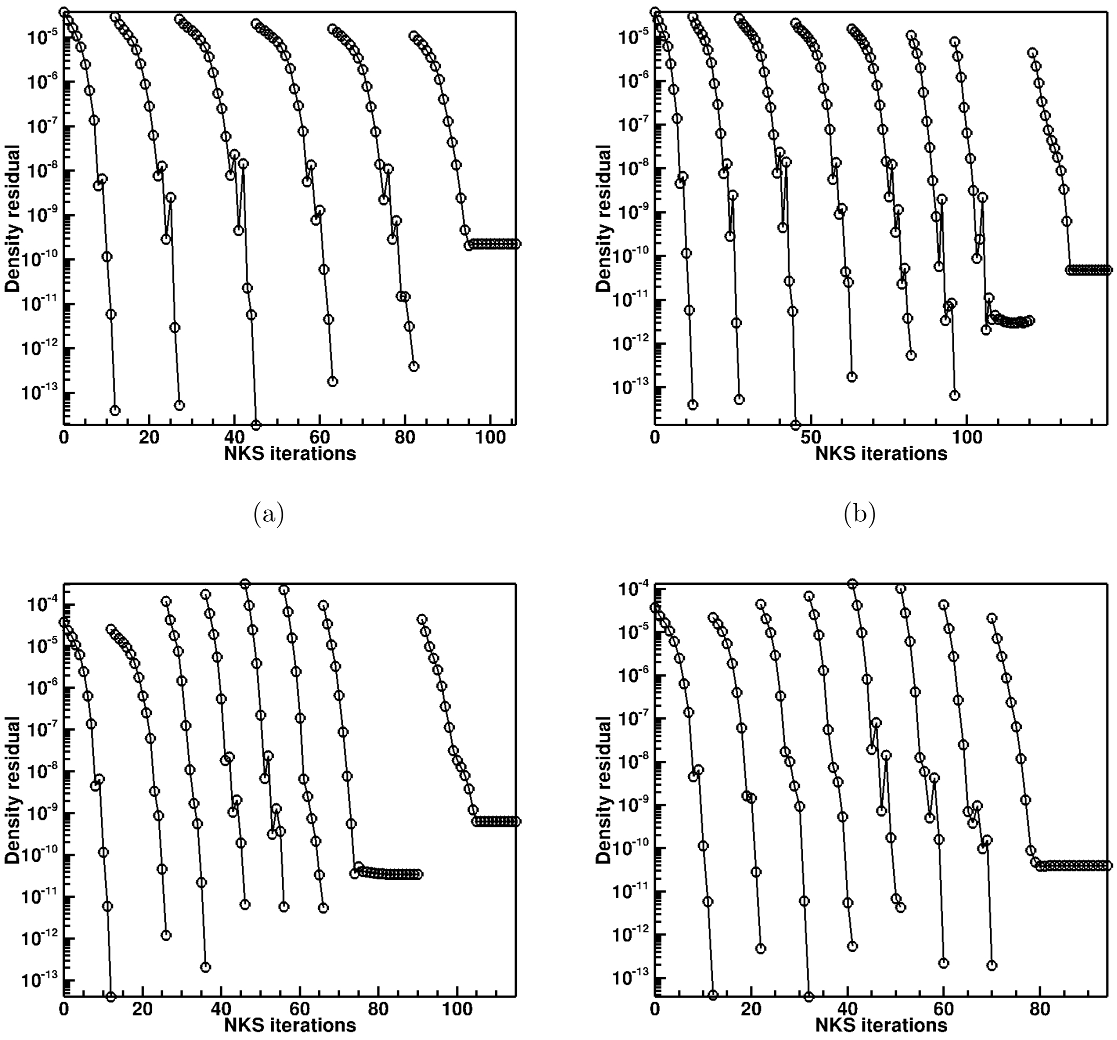


Figure 4.22: Predicted Mach number contours for steady $M = 0.5$ subsonic viscous flow past a NACA0012 airfoil after 5 uniform refinements. Blocks: 4,096. Cells: 1,638,400.

on the airfoil surface. The number of cells per block is $10 \times 10 \times 4 = 400$, with 4 cells in the out-of-plane direction. The chord length of the airfoil is taken to be unity and the far-field boundary is located at a distance of 32 chord lengths from the airfoil boundary in the radially outward direction. An adiabatic no-slip boundary condition is imposed at the airfoil boundary. At the top, bottom and upstream/left farfield boundaries, the flow variables except pressure are assumed to be fixed at subsonic inflow conditions, while the pressure is linearly extrapolated from the interior domain. At the downstream/right farfield boundary, the pressure is assumed to have a constant value while all other flow variables are linearly extrapolated for the outgoing subsonic flow. The ratio of specific heats, γ, is assumed to be 1.4. The main features of this subsonic flow are the formation of the viscous boundary layer, the flow acceleration and subsequent deceleration over the airfoil surface. This case has also been examined in several previous studies [72,211–214]. Figure 4.22 shows Mach number distribution for this case after a converged solution has been obtained after 5 uniform refinements of the initial mesh. The convergence of the steady-state density solution residual for this subsonic viscous flow case is shown in Figure 4.23 for the complete set of refined meshes for the four refinement techniques considered here (i.e., for uniform, gradient-based, CC-based, and ECC-based refinment strategies), again confirming the fully-converged nature of the predicted steady-state



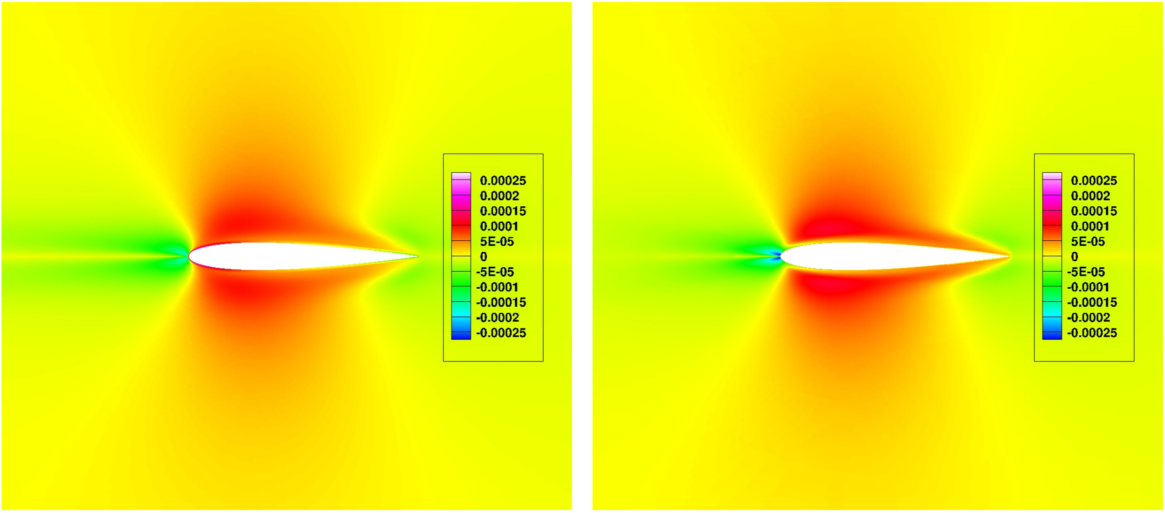
(a)

(b)

(c)

(d)

Figure 4.23: Density residual convergence history on the complete set of adaptive refined AMR meshes for steady $M = 0.5$ subsonic viscous flow past a NACA0012 airfoil: (a) uniform refinement; (b) gradient-based refinement; (c) output-based refinement using CC; (d) output-based refinement using ECC.

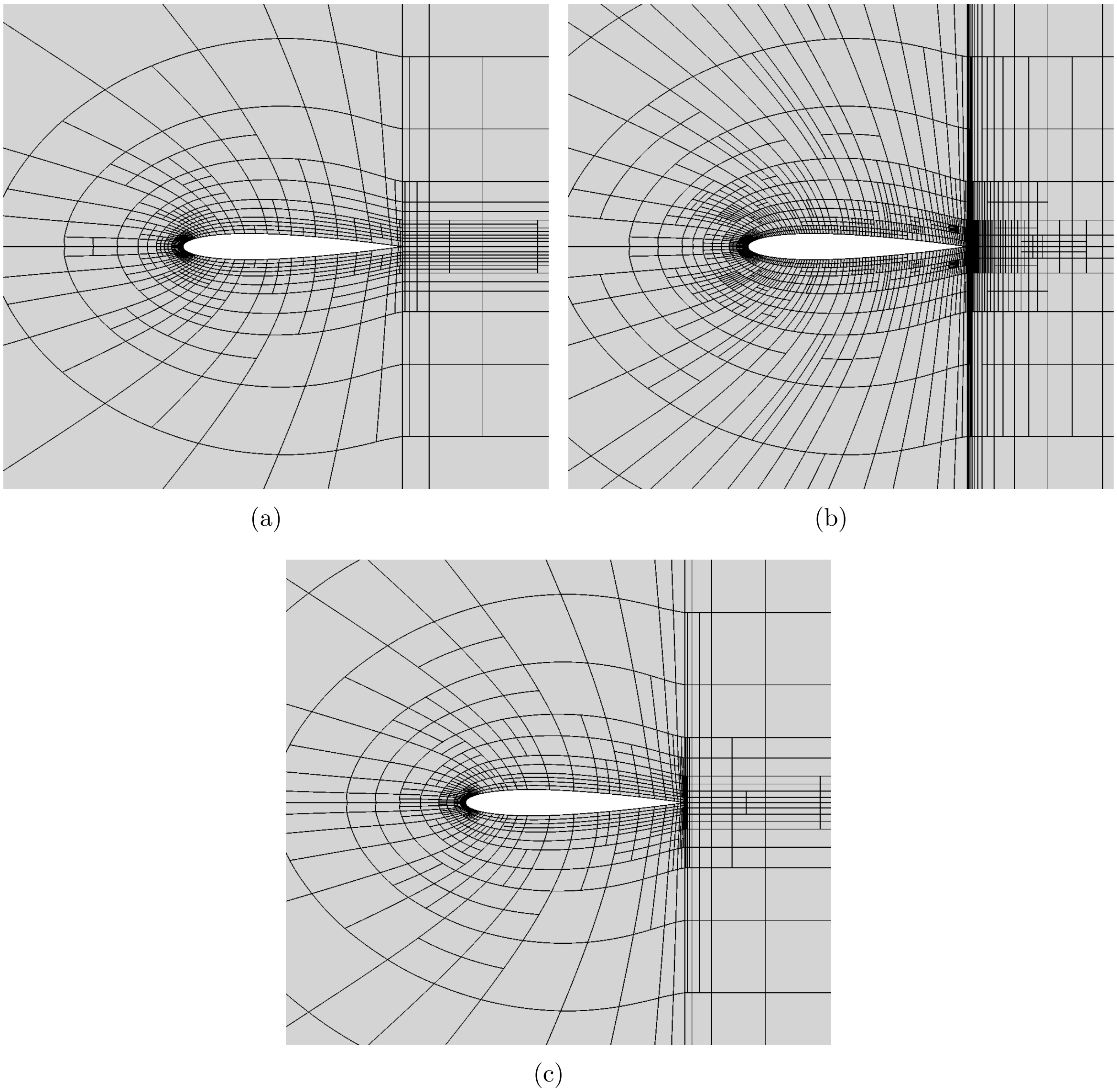


(a) (b)

Figure 4.24: Energy adjoint distribution for steady $M = 0.5$ subsonic viscous flow past a NACA0012 airfoil : (a) after 5 uniform refinements. Blocks: 4,096. Cells: 1,638,400. (b) after 7 refinements using CC as refinement criterion. Blocks: 2,126. Cells: 850,400.

solutions.

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. Figure 4.24 shows the distribution of the 5^{th} component of the adjoint solution vector on a refined mesh obtained after 5 uniform refinements of the initial mesh and after 7 anisotropic refinements using the CC refinement criterion. Figure 4.25 shows a comparison between meshes obtained after several adaptive refinements on the initial mesh, using various gradient-based and output-based refinement criteria. In particular, Figure 4.25(a) shows the mesh obtained after 7 refinements using density and velocity gradients to drive the mesh adaptation. This approach leads to excessive refinement of the stagnation region near the leading edge of the airfoil. This approach also leads to significant refinement in the region very close to the airfoil surface where the velocity gradients are high due to the presence of the viscous boundary layer. Alternatively, Figure 4.25(b) shows the mesh obtained after 7 refinements using the CC method to drive the mesh adaptation. There is substantial refinement near the leading edge of the airfoil where most of the pressure drag comes from, as well as in the region containing the boundary layer. Substantial refinement can also be observed in the viscous wake in the direction orthogonal to the incoming flow, just as observed for the density gradient-based approach. Finally, the mesh obtained after 7 refinements using the ECC criterion to drive the mesh adaptation,



(a)

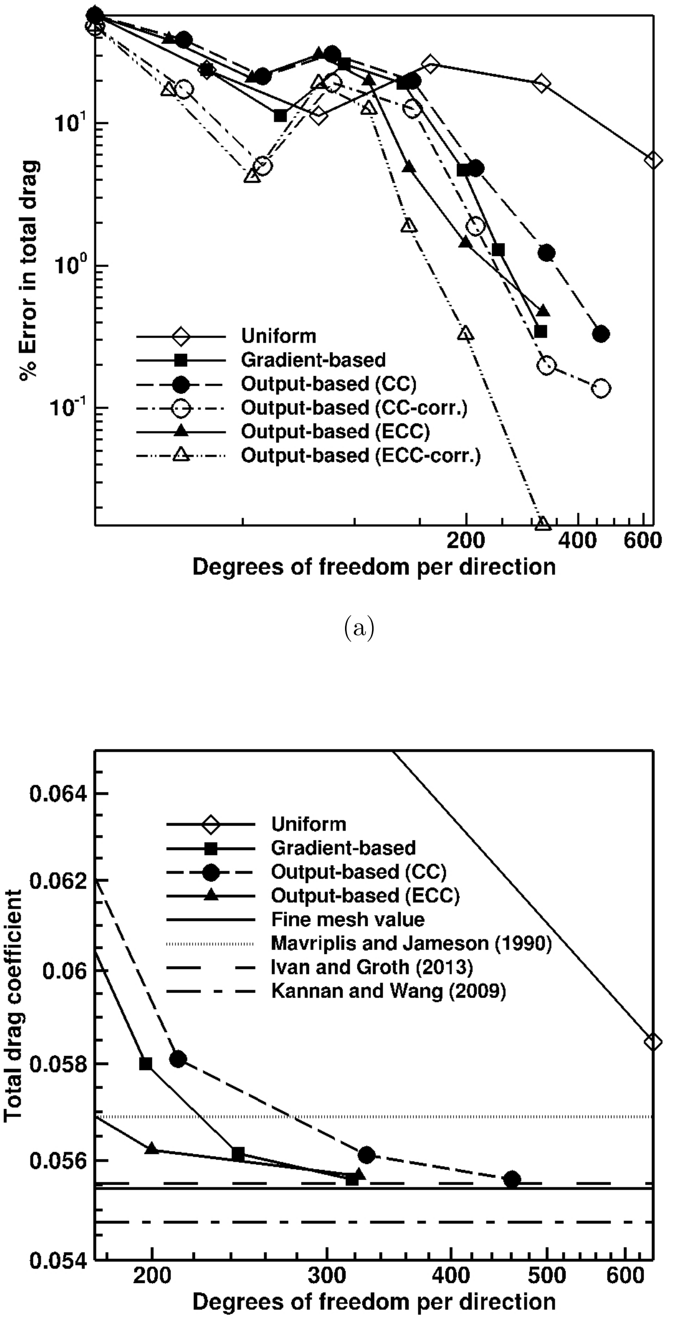
(b)

(c)

Figure 4.25: Adapted meshes showing the grid block boundaries (not individual cells) for steady $M = 0.5$ subsonic viscous flow past a NACA0012 airfoil: (a) After 7 refinements using density and velocity gradients as refinement criteria. Blocks: 1,012. Cells: 404,800. (b) After 7 refinements using CC as refinement criterion. Blocks: 2,126. Cells: 850,400. (c) After 7 refinements using ECC as refinement criterion. Blocks: 1,042. Cells: 416,800.

as shown in Figure 4.25(c), shows a similar trend to that obtained with the CC refinement strategy, however with a relatively sparser level of refinement.

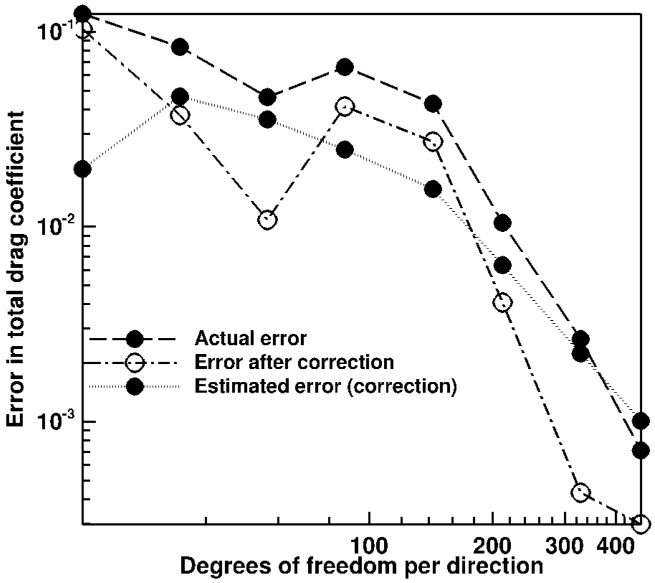
Figures 4.26(a) and 4.26(b) show plots of the percentage error in the computed total drag coefficient and the actual value of the computed total drag coefficient against the total number of degrees of freedom per direction for this two-dimensional problem,



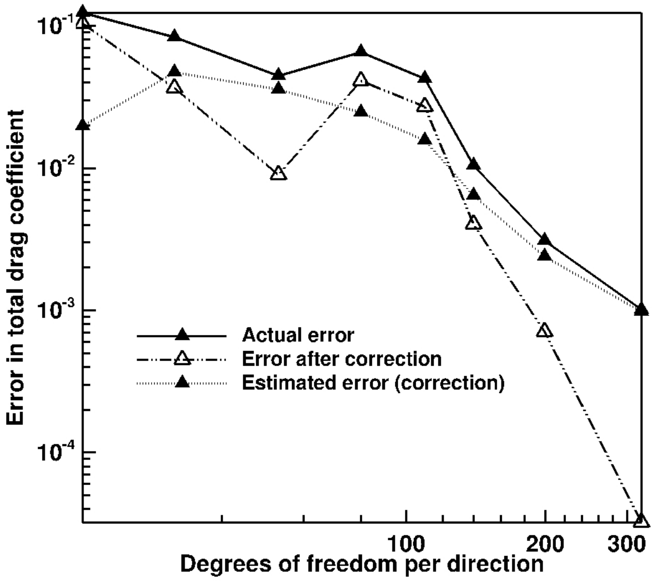
(a)

(b)

Figure 4.26: (a) Convergence of the percentage error in the predicted total drag coefficient and (b) convergence of the predicted total drag coefficient, as a function of the mesh size (number of degrees of freedom per direction) for steady viscous $M = 0.5$ subsonic flow past a NACA0012 airfoil.



(a)



(b)

Figure 4.27: Convergence of the actual error, the error after adding the computable correction and the error estimate (absolute computable correction) as a function of the mesh size (number of degrees of freedom per direction) (a) for the CC method and (b) for the ECC method, for steady viscous $M = 0.5$ subsonic flow past a NACA0012 airfoil.

i.e. $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells in the computational mesh, respectively. For the output-based techniques, the error in the corrected drag coefficient obtained after adding the computable correction to the functional is also shown in Figure 4.26(a). A fine mesh value, from a converged flow solution obtained after 7 uniform refinements on the initial mesh, is shown for reference in Figure 4.26(b). Values of the drag coefficient calculated by previous researchers [72, 211, 212, 214] are also depicted in Figure 4.26(b) for comparison with this fine mesh value. The latter is used as the reference value of the functional. From Figures 4.26(a) and 4.26(b), it is evident that all of the adaptive methods are eventually able to reduce the error in the drag coefficient to the same order. Among these, the output-based method employing CC as the refinement criterion provides the worst performance. Employing the gradient-based approach leads to 52% mesh savings as compared to the CC method for the same accuracy (∼0.3%). However, employing the ECC method to drive the mesh refinement shows a similar performance as the gradient-based approach. Adding the correction for the ECC method shows the best performance with 61% mesh savings with respect to the the gradient-based approach.

As for the supersonic inviscid wedge flow, the actual error in the functional, the corrected error, and the estimated error (computable correction) are shown in Figure 4.27 for the CC and ECC output-based methods. As for the previous inviscid case, it can be observed that the actual and estimated errors approach each other as the computational mesh is refined as should be expected. Furthermore, the computable correction can be applied to provide a significantly improved numerical estimate of the drag for the airfoil.

4.2.2 Supersonic Flow Past a NACA0012 Airfoil

In the second viscous flow case, horizontal viscous supersonic flow with a Mach number $M = 1.2$ and a Reynolds Number $Re = 1,000$ is intercepted by a NACA0012 airfoil at an angle of attack given by $\alpha = 0^\circ$. The stretched C-shaped grid shown in Figure 4.28 comprising 4 blocks is used for this case, with the stretching essential for efficiently capturing the viscous boundary layer. The number of cells per block is $16 \times 16 \times 4 = 1,024$, with 4 cells in the out-of-plane direction. Again, 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. An adiabatic no-slip boundary condition is imposed on the airfoil boundary. At the top, bottom and upstream/left farfield boundaries, the flow variables are kept fixed at the supersonic flow conditions, whereas at the downstream/right farfield boundary, constant extrapolation is imposed for the outgoing flow. The ratio of specific

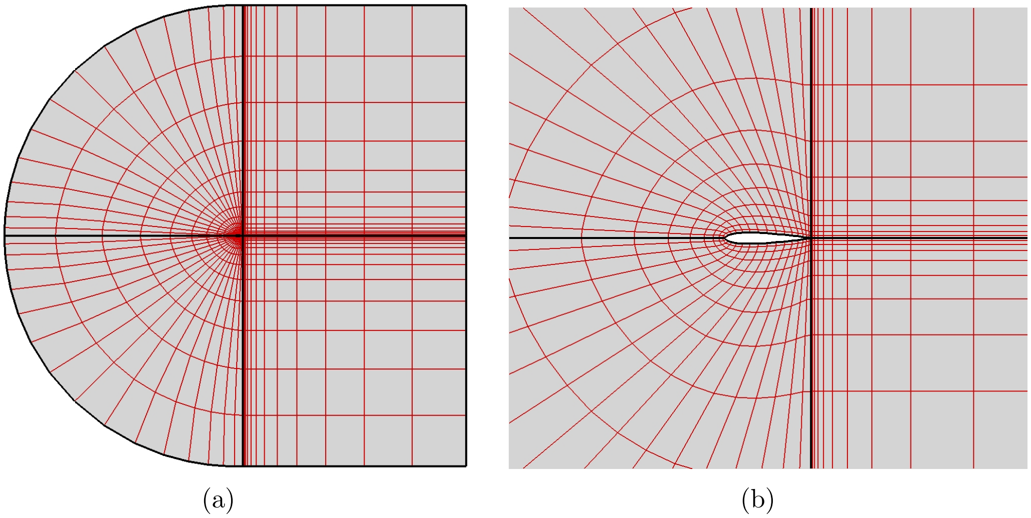


Figure 4.28: Initial computational mesh for steady supersonic viscous flow past a NACA0012 airfoil showing: (a) entire view of the mesh and (b) close-up view of the mesh. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 4, the total number of cells per block is 16×16×4=1,024 and the total number of cells is 4,096.

heats, γ, is assumed to be 1.4. A bow shock is formed in front of the leading edge of the airfoil. Since the flow is just above supersonic, the bow shock is detached and located at some distance upstream 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. This case has been previously studied by Hartmann [215]. Figure 4.29 shows the predicted density distribution for this case after a converged solution has been obtained following 5 uniform refinements of the initial mesh.

The functional used for the viscous supersonic airfoil flow is again the total drag force on the airfoil in the direction of the free-stream flow, which comprises both the pressure and viscous drag components. Figure 4.30 shows the predicted distribution of the 5^{th} component of the adjoint solution vector corresponding to the energy for this case after a converged solution has been obtained following 5 uniform refinements of the initial mesh as well as following 6 anisotropic refinements using the CC refinement criterion. Figure 4.31 shows a comparison between the meshes obtained after several adaptive refinements on the initial mesh, using various gradient-based and output-based refinement criteria. Figure 4.31(b) shows the mesh obtained after 9 refinements using the density gradient to drive the mesh adaptation. The gradient-based approach extensively refines the bow shock in front of the leading edge of the airfoil. The two weaker shocks

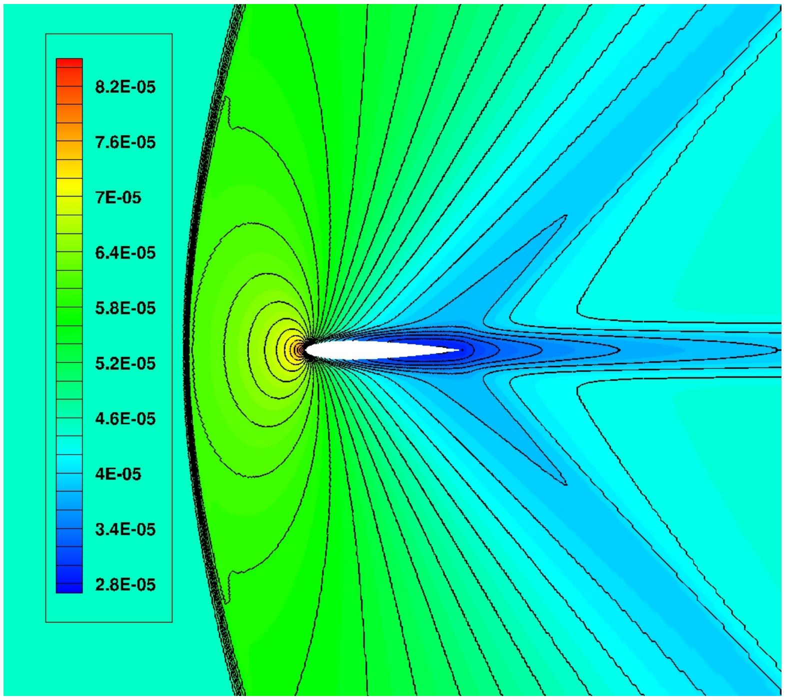
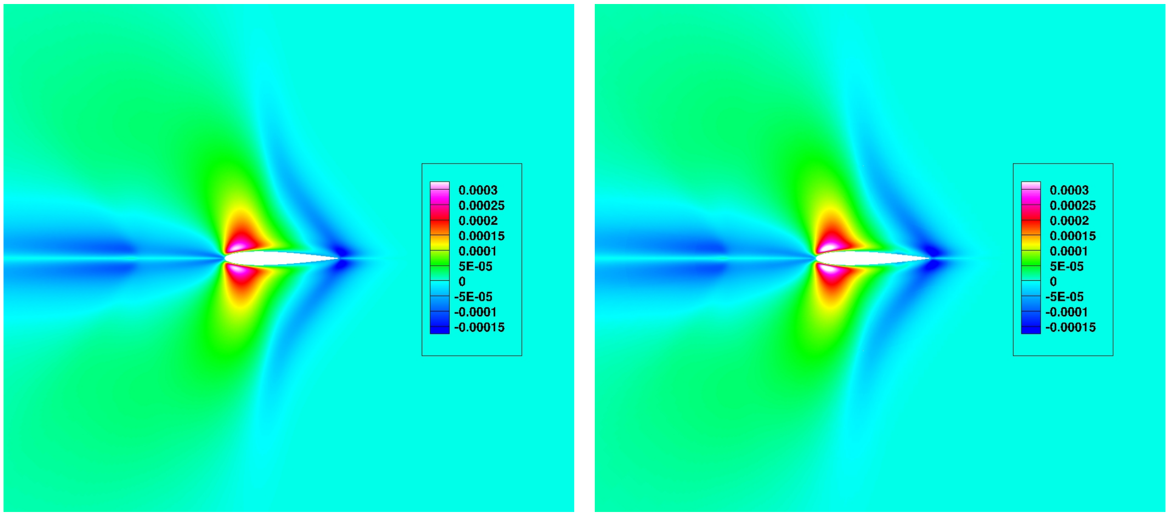


Figure 4.29: Predicted density contours for steady $M = 1.2$ supersonic viscous flow past a NACA0012 airfoil after 5 uniform refinements. Blocks: 4,096. Cells: 4,194,304.

originating from the trailing edge, the viscous boundary layer and the viscous wake also receive a considerable amount of refinement. Figures 4.31(c) and 4.31(d) show the mesh obtained after 6 output-based adaptive refinements using the CC and ECC methods as the refinement criteria, respectively. On both of these, only a portion of the bow shock directly in front of the leading edge is refined along with the viscous boundary layer. The viscous wake region shows a small amount of refinement while the trailing edge shocks are not captured at all. The mesh obtained using ECC also shows refinement in the region upstream from the airfoil between the bow shock and the leading edge.

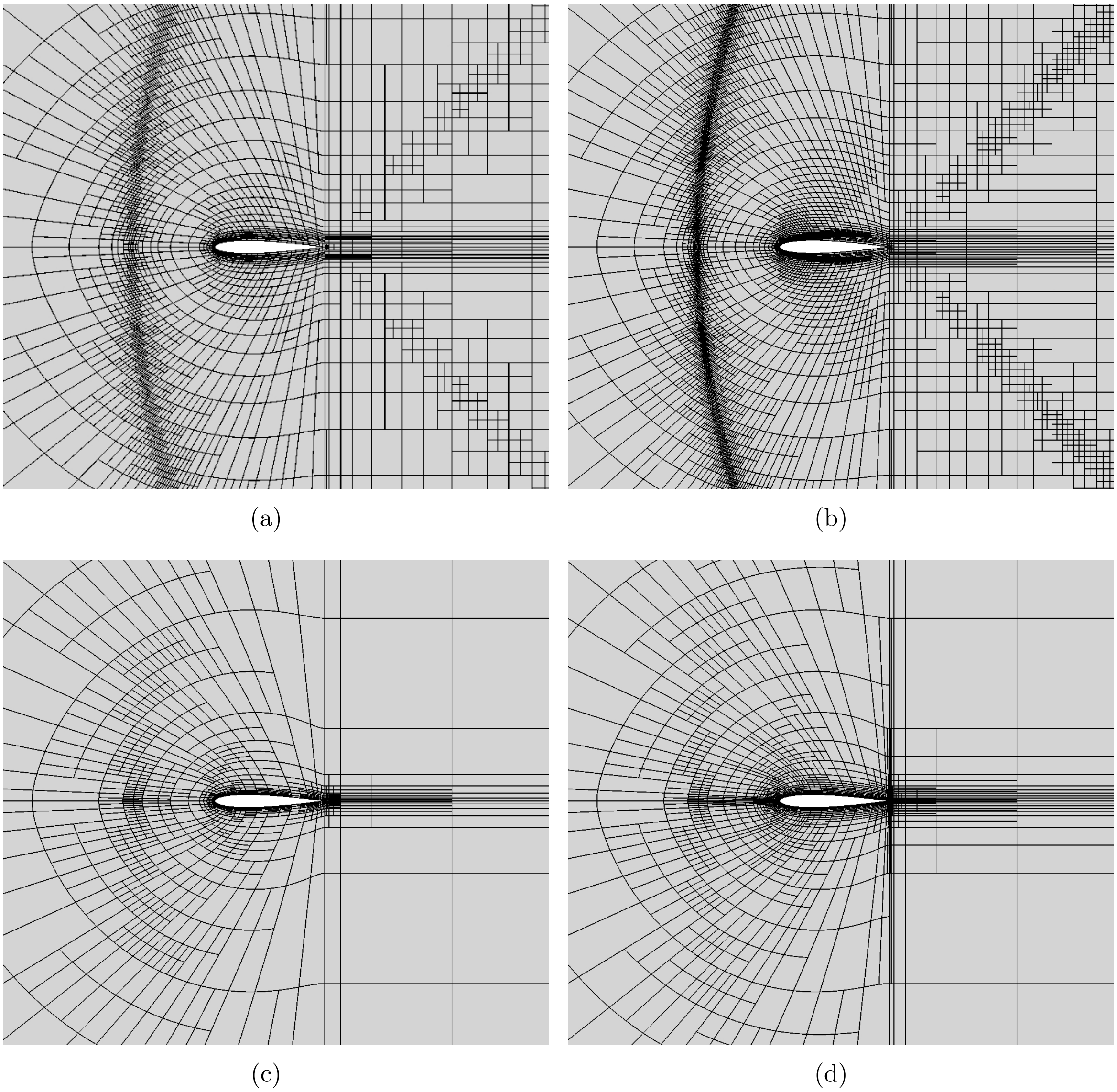
Figures 4.32(a) and 4.32(b) show plots of the percentage error in the computed total drag coefficient based on a fine mesh result and the actual value of the computed total drag coefficient against the total number of degrees of freedom per direction, i.e. $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells in the computational mesh, respectively. The reference fine mesh value for the drag coefficient of 0.21813 was obtained from a converged flow solution following 7 uniform refinements of the initial mesh and is shown in Figure 4.32(b) for reference. The value of the drag coefficient obtained previously by Hartmann [215], 0.20882, is also shown in the figure for comparison with this fine mesh result. It can be seen that all of the adaptive techniques can successfully converge to the same order of error in the drag coefficient after a number of refinements. However, the



(a) (b)

Figure 4.30: Energy adjoint distribution for steady $M = 1.2$ supersonic viscous flow past a NACA0012 airfoil : (a) after 5 uniform refinements. Blocks: 4,096. Cells: 4,194,304. (b) after 6 refinements using CC as refinement criterion. Blocks: 1,202. Cells: 1,230,848.

output-based approaches perform significantly better than the gradient-based approach in terms of savings in the computational mesh for the same order of error. Employing the ECC method as the refinement method achieves a 73% reduction in mesh size as compared to the gradient-based method. Employing the CC to drive the refinement leads to 82% reduction in mesh size with respect to the gradient-based method for the same accuracy (∼0.04%), again showing the value of the proposed output-based anisotropic AMR schemes.



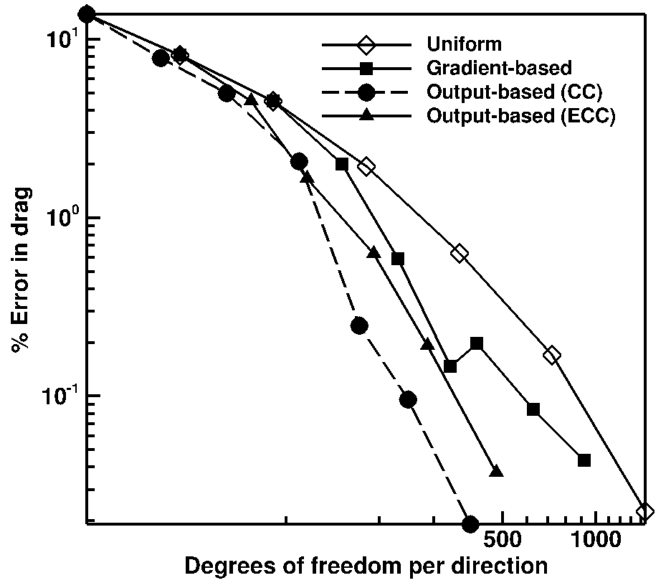
(a)

(b)

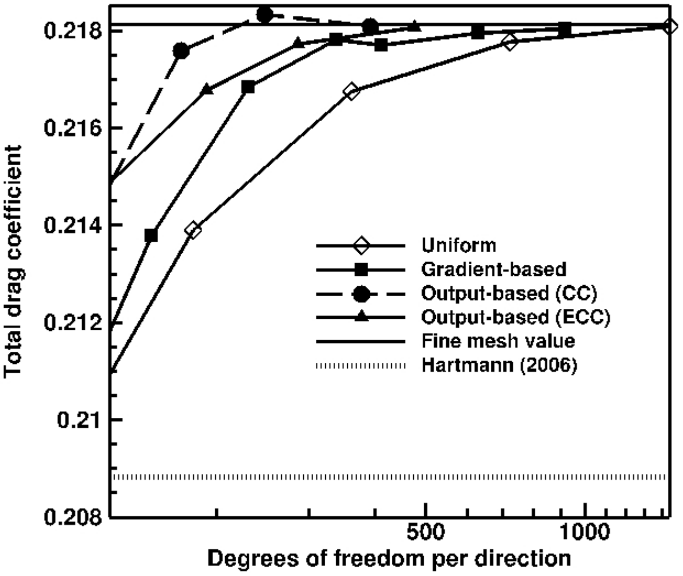
(c)

(d)

Figure 4.31: Adapted meshes showing the grid block boundaries (not individual cells) for steady $M = 1.2$ supersonic viscous flow past a NACA0012 airfoil: (a) After 8 refinements using density gradient as refinement criterion. Blocks: 6,602. Cells: 6,760,448. (b) After 9 refinements using density gradient as refinement criterion. Blocks: 14,520. Cells: 14,868,480. (c) After 6 refinements using CC as refinement criterion. Blocks: 1,202. Cells: 1,230,848. (d) After 6 refinements using ECC as refinement criterion. Blocks: 1,772. Cells: 1,814,528. Meshes shown in (a), (c) and (d) have approximately the same error in the functional.



(a)



(b)

Figure 4.32: (a) Convergence of the percentage error in the predicted total drag coefficient and (b) convergence of the predicted total drag force, as a function of the mesh size (number of degrees of freedom per direction) for steady viscous $M = 1.2$ supersonic flow past a NACA0012 airfoil.

Chapter 5

Integrated Solar Wind-MHD Framework

This chapter describes the new integrated solar wind-MHD framework developed as part of this thesis for possible eventual use in operational space-weather forecasting. An overview of existing methods for numerical modelling of the solar wind is given in Section 5.1. Section 5.2 discusses the combined model used in this work, comprising the physics-based solar wind models employed for coronal calculations and the parallel adaptive finite-volume MHD framework for the heliospheric calculation. Section 5.3 discusses numerical results obtained for a steady-state calculation of the ambient solar wind and an unsteady calculation performed for forecasting solar-wind properties at the Earth.

5.1 Overview and Motivation

In the lower corona, the magnetic pressure, p_B, is much higher than the plasma thermal pressure p_{th}, such that the plasma beta, β, given by

$$\beta = \frac{p_{th}}{p_B} = \frac{n k_b T}{|\mathbf{B}|^2/(2\mu_0)} \ll 1, \tag{5.1}$$

is low. Here, n is the particle number density, k_b is the Boltzmann constant, T is the temperature, $\mathbf{B}$ is the magnetic field and μ_0 is the magnetic permeability of vacuum. In the coronal region where the plasma beta is low ($\beta \ll 1$), the structure of the Sun's atmosphere and plasma dynamics are both defined by the topology of the magnetic field. As a result, models based on various approximations have been proposed to predict and determine the structure of the coronal magnetic field using solar magnetograms from observations as inputs [216]. Most of the coronal magnetic field models used in the solar wind forecast community are simple models based on the potential-field theory [217–221]. Potential-field models, such as the Potential-Field Source Surface (PFSS) model and the Schatten Current Sheet (SCS) model, offer significant advantages, as compared to MHD models especially in the coronal region of the Sun, particularly for space weather modelling. A summary of the primary advantages of such models is as follows:

1. Potential-field models are computationally less expensive than MHD models as they only involve the solution for the magnetic field. MHD models, on the other hand, also provide a solution for the plasma properties such as density, velocity and pressure. Riley *et al.* [222] and Owens *et al.* [223] have found that potential-field models can provide similar predictions of the global topology of the coronal magnetic field as MHD models at a lower computational cost.

2. The solar wind does not attain its asymptotic super-Alfvénic speed until it covers a radial distance of about 10-15R_{Sun} outward from the Sun's surface where R_{Sun} is the Sun's radius. Hence, in order to ensure well-posedness of the problem, boundary conditions at the Sun's surface for MHD models need to respect the characteristic equations of the governing PDEs. Thus, in MHD models, boundary conditions for density, velocity and pressure can be very difficult to formulate at the Sun's surface.

3. It is also difficult to model phenomena such as coronal heating and solar wind acceleration in the corona. These difficulties are circumvented by the potential-field models which only solve directly for the magnetic field.

4. In coronal MHD simulations, the length and time scales are much smaller as compared to those for outer heliospheric MHD simulations, leading to a higher computational cost.

As such, a numerical simulation framework employing potential-field models to predict the coronal magnetic field structure and also provide magnetic field and solar wind velocity forecasts at the Earth has been developed by Nikolic [224]. Nevertheless, it is recognized that for better understanding of the dynamic interactions between the heliospheric magnetic field and the solar wind plasma and transient phenomena such as CMEs [225, 226], potential-field models are certainly not sufficient and more sophisticated models are required. As described earlier in Chapter 1, various attempts have been made to model the corona and the solar wind using the equations of MHD [6–17, 19–22, 24–26, 28–33].

More recently, combined data-driven approaches exploiting the simplicity of potential-field models as well as the sophistication and more detailed physics of MHD models have been considered. In a combined approach, a potential-field model is used for the near-Sun inner corona and acts as an input to the MHD model for the heliosphere. The boundary conditions for the MHD model in this case would be very easy to implement as the solar wind would have attained its asymptotic super-Alfvénic speed by the time it enters the heliosphere. As such, all variables at the inner boundary of the heliosphere would have to be specified as inputs to the MHD model. Also, the difficulty of modelling coronal phenomena such as solar wind acceleration and coronal heating and implementing them in MHD models is circumvented.

Detman *et al.* [227] employed a PFSS-SCS model for the coronal magnetic field structure and used it as a driver for the heliospheric MHD model. They used a spherical coordinate grid covering 360° in ϕ and 90° in θ centered on the solar equatorial plane. ENLIL [228, 229], an operational space weather prediction tool, uses MHD modelling for the simulation of solar wind structures in a Cartesian or spherical geometry, using a fixed or an isotropic block-based adaptive mesh. The ENLIL model covers a latitudinal range of $+60°$ to $-60°$ and a full longitudinal range of 360° and uses the solar wind speed obtained from the Wang-Sheeley-Arge (WSA) [230–236] model and the magnetic field structure obtained from a PFSS model to represent the solar wind in the region $2.5R_{\text{Sun}}$ to $5R_{\text{Sun}}$. The WSA model correlates the solar wind speed to the photospheric magnetic field and the magnetic flux tube the solar wind passes through as it emerged out of the lower corona. If the flux tube widens, the solar wind is relatively slower, and if it remains narrow, the solar wind speed is higher. The expansion factor for the flux tube can be obtained by tracing the open magnetic field lines back to their origin in the photosphere. A general mathematical formulation for the WSA model is provided in Section 5.2.4. Baker *et al.* [237] employed the WSA-ENLIL code to reproduce solar wind observations from the MESSENGER spacecraft in orbit around Mercury. The ENLIL prediction tool also has the capability to model coronal mass ejections [238, 239]. EUHFORIA [240], a European heliospheric forecasting tool, also uses a PFSS-SCS model for the corona extending from the photosphere up to 0.1 AU, and an MHD model from 0.1 AU extending up to 2 AU so as to include the orbit of Mars. Like ENLIL, EUHFORIA also uses the WSA model to obtain solar wind speed and a spherical mesh spanning 120° in latitude and 360° in longitude. EUHFORIA also has the capability to model CMEs using a cone model with an intrinsic spheromak magnetic field [241–243]. In other research, Nakamizo *et al.* [244] used a PFSS model for the coronal calculation and performed the heliospheric MHD calculation on an angular triangular unstructured grid. Shiota *et al.* [245, 246] used a PFSS model for the coronal magnetic field calculation and used it to drive the heliospheric MHD calculation from $25R_{\text{Sun}}$ onwards on a spherical Yin-Yang grid [247].

Despite these efforts, the implementation based on global MHD models and simulation tools into operational solar wind prediction models is still in its nascent stage.

5.2 Data-Driven Solar Wind-MHD Model

In the proposed new data-driven solar-wind model for space weather prediction, a coupled coronal-heliospheric model is used in which PFSS, SCS and global ideal MHD models are all combined. A schematic diagram showing the numerical models used in the various regions of the Sun's atmosphere and heliosphere is shown in Figure 5.1. The prescription of the inner region of the corona and solar wind is provided by the PFSS-SCS model of Nikolic *et al.* [224,248]. The global MHD model and numerical solution method described in Chapter 2 is then coupled to this inner model and used to simulate and model the plasma flows in the outer solar-wind and heliosphere. The latter makes use of the parallel block-based anisotropic AMR framework and Godunov-type finite-volume scheme for multi-block hexahedral mesh [49, 50, 64, 68, 70] as well as the cubed-sphere meshing capabilities of the framework [67, 68] to represent the spherical-shaped computational domain of the solar outflows and heliosphere. The PFSS and SCS models are described in Section 5.2.1 and Section 5.2.2, respectively. The global MHD model is again briefly reviewed in Section 5.2.3 for completeness. Section 5.2.4 then describes the coupling of the three models and boundary conditions used for the MHD model.

5.2.1 Potential-Field Source Surface (PFSS) Model for Coronal Magnetic Field

A PFSS model [217–219] is used to derive the global 3D magnetic field in the solar corona from photospheric field observations. Global Oscillation Network Group (GONG) magnetograms [249] on a $180° \times 360°$ solid angle grid are processed and used as the input to the PFSS model. The PFSS model assumes the existence of a fixed spherical "source surface" where the coronal magnetic field is purely radial. A radius of $r = R_s = 2.5 R_{\text{Sun}}$ where R_s is the radius of the "source surface" and R_{Sun} is the Sun's radius, is widely accepted for modelling and is used here. The model is based on the assumption that there are no currents in the coronal region $R_{\text{Sun}} \leq r \leq R_s$. A current-free approximation of the magnetic field satisfies the condition that

$$\nabla \times \mathbf{B} = 0, \tag{5.2}$$

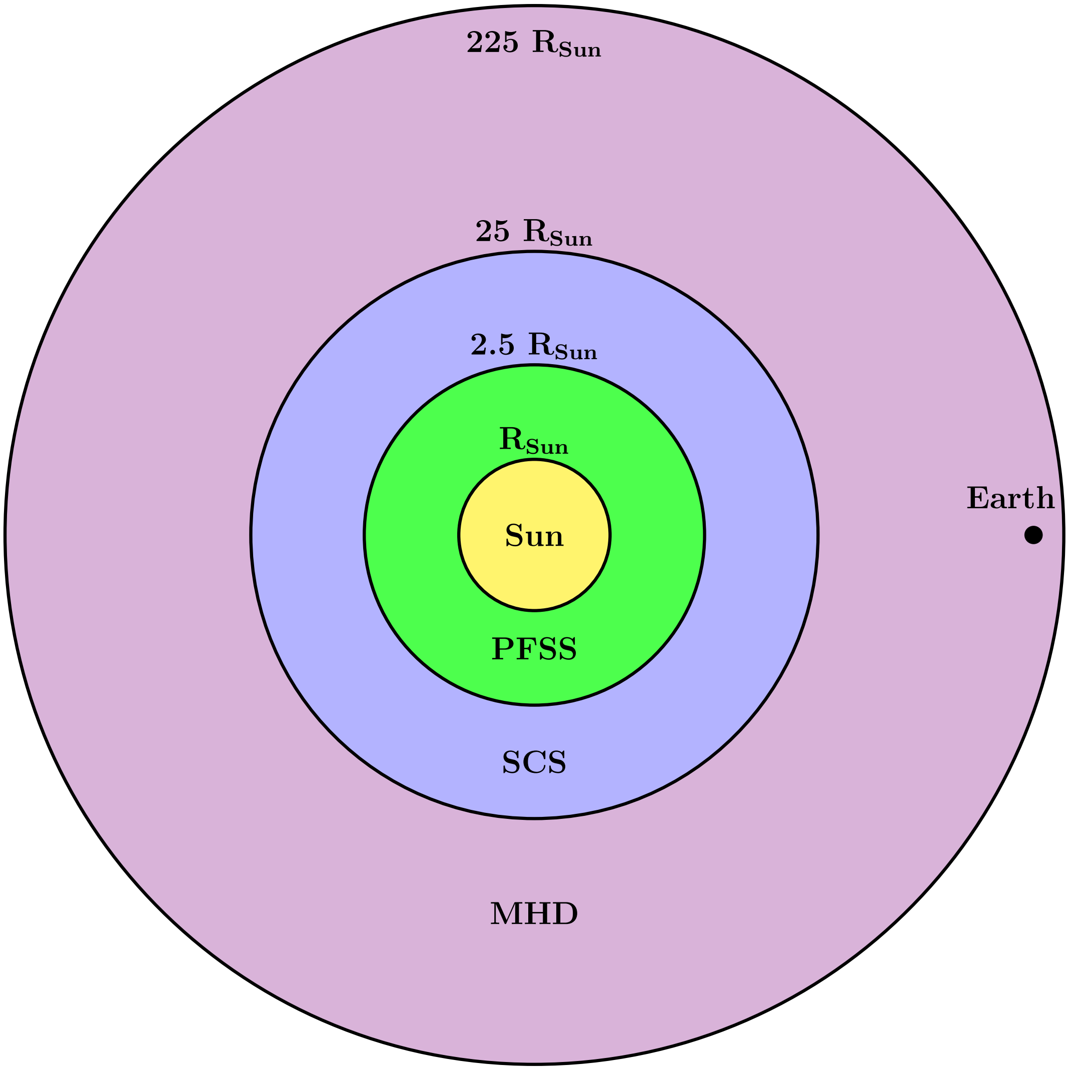


Figure 5.1: Schematic diagram showing the numerical models used in the various regions of the Sun's atmosphere.

which implies that the magnetic field can be expressed as the gradient of a scalar potential ψ and written as

$$\mathbf{B} = -\nabla\psi. \tag{5.3}$$

Along with the solenoidal property of the magnetic field $\nabla \cdot \mathbf{B} = 0$, Equation (5.3) can be expressed as a Laplace equation for the magnetic field potential, ψ, given by

$$\nabla^2\psi = 0. \tag{5.4}$$

Using a separation of variables such that $\psi(r, \theta, \phi) = R(r)\Theta(\theta)\Phi(\phi)$ in spherical coordinates, where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$, and assuming that at the source surface, $r = R_s$, the magnetic field is purely radial, i.e. $\psi(R_s, \theta, \phi)$ = constant, the solution of Equation (5.4) for the region $R_{\text{Sun}} \leq r \leq R_s$ can be expressed in terms of spherical harmonics and written as

$$\begin{aligned}\psi = &\sum_{n=1}^{\infty}\sum_{m=0}^{n} P_n^m(\cos(\theta))(g_{nm}\cos(m\phi) + h_{nm}\sin(m\phi)) \\ &\times \left[R_{\text{Sun}}\left(\frac{R_{\text{Sun}}}{r}\right)^{n+1} - R_s\left(\frac{R_{\text{Sun}}}{R_s}\right)^{n+2}\left(\frac{r}{R_s}\right)^{n}\right],\end{aligned} \tag{5.5}$$

where $P_n^m(\cos(\theta))$ represents the associated Legendre polynomial and g_{nm} and h_{nm} are coefficients depending on the summation indices n and m. Using Equations (5.3) and (5.5) and the orthogonality of Legendre polynomials, the magnetic field components can be expressed as a function of the radial component of the photospheric magnetic field $B_r(R_{\text{Sun}}, \theta, \phi)$. For a detailed analysis and derivation of the PFSS model used here, the reader is referred to the previous work by Nikolic [224].

5.2.2 Schatten Current Sheet (SCS) Model for Solar Wind Plasma Properties

The PFSS model does not take into account the effect of plasma transport on the solar corona. Although the approximation works very well for the lower corona, the condition $\beta \ll 1$ is violated at some distance from the Sun and the solution dictated by Equation (5.5) cannot represent the magnetic field and the solar outflows for all $r \geq R_{\text{Sun}}$. In order to include the effects of plasma currents on the structure of the solar corona, Schatten [220] proposed the introduction of a new spherical source surface at $r = R_{cp}$, which is called the cusp surface, where transverse currents are allowed [220]. These currents are assumed to be limited to thin sheets between regions of opposite polarity of the magnetic field where the Lorentz force $\mathbf{j} \times \mathbf{B}$ is small. Whereas the PFSS model forces the magnetic field lines to be radial at $r = R_s$, the SCS model allows a non-radial coronal field structure for $r \geq R_s$. The cusp surface is taken to be equal to R_s, i.e. $R_{cp} = R_s$, the outer boundary for the PFSS model. In the SCS model, the coronal magnetic field from the PFSS model at $r = R_{cp} = R_s$ is first reoriented to point outwards, i.e., if $B_r(R_{cp}) \geq 0$, no changes are needed, but if $B_r(R_{cp}) < 0$, the signs of $B_r(R_{cp}), B_\theta(R_{cp})$ and $B_\phi(R_{cp})$ are changed, where $B_r(R_{cp}), B_\theta(R_{cp})$ and $B_\phi(R_{cp})$ are the radial, latitudinal and longitudinal components of the magnetic field, respectively. The field beyond R_{cp} is obtained by matching the potential-field solution for $r \leq R_s$, i.e., the solution obtained using Equation (5.5) for $r \geq R_{cp}$, with the magnetic field components of the reoriented field at $r = R_{cp}$. For the SCS model, the $r/(R_s)$ terms in Equation (5.5) are discarded as the SCS model solution is not bounded by an outer surface. The effect of this step is to open the magnetic field lines, so that no closed magnetic field lines exist beyond R_{cp}. Finally, proper polarity is assigned to the magnetic field lines in the region $r \geq R_{cp}$ using the polarity before the field reorientation at R_{cp}. This polarity restoration ensures that the divergence constraint, $\nabla \cdot \mathbf{B} = 0$, is not violated. Furthermore, it implies that current sheets are introduced between the magnetic fields of opposite polarity.

The SCS model described above is used to determine the magnetic field in the region $R_{cp} = 2.5R_{\text{Sun}} \leq r \leq 25R_{\text{Sun}}$. The outer boundary of this SCS region, $25R_{\text{Sun}}$, is then taken as the inner boundary of the outer heliospheric region where the full set of ideal MHD equations are solved to describe the solar wind behaviour.

5.2.3 3D Ideal MHD Model for Solar Wind Solution to 1 AU

The global ideal MHD model and parallel finite-volume scheme with anisotropic block-based AMR as described in Chapter 2 are used here to simulate the solar-wind plasma in the outer heliosphere. For the sake of completeness, the system of governing equations for ideal MHD given in Chapter 2 are re-summarized here as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{5.6}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} - \mathbf{B}\mathbf{B} + p_T \mathbf{I}) = -\rho \left[\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right] - 2\rho \left(\boldsymbol{\Omega} \times \mathbf{u}\right) - \left(\nabla \cdot \mathbf{B}\right) \mathbf{B}, \tag{5.7}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot ((e + p_T)\mathbf{u} - (\mathbf{u} \cdot \mathbf{B})\mathbf{B}) = -\rho \mathbf{u} \cdot \left[\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right] - \left(\nabla \cdot \mathbf{B}\right) \mathbf{u} \cdot \mathbf{B}, \tag{5.8}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B}) = -\left(\nabla \cdot \mathbf{B}\right) \mathbf{u}, \tag{5.9}$$

where the reader is reminded that ρ, $\mathbf{u}$, e and $\mathbf{B}$ are the plasma density, velocity, specific total energy and magnetic field, respectively. Also, $\boldsymbol{\Omega}$ is the angular velocity of the reference frame if the equations are solved in a non-inertial rotating frame, t is the time and $\mathbf{x}$ is the position vector. The total pressure, p_T, is given by

$$p_T = p + \frac{|\mathbf{B}|^2}{2}, \tag{5.10}$$

where p is the thermal pressure of the plasma. The total energy e is given by

$$e = \frac{\rho|\mathbf{u}|^2}{2} + \frac{p}{\gamma - 1} + \frac{|\mathbf{B}|^2}{2}, \tag{5.11}$$

where $\gamma = C_p/C_v$ is the ratio of specific heats. A value of $\gamma = 5/3$ was used for the heliospheric model. As noted earlier in Chapter 2, Equations (5.6)–(5.9) can be expressed in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{Q} + \mathbf{S}, \tag{5.12}$$

where the vector of conserved variables, $\mathbf{U}$, and the flux tensor, $\mathbf{F}$, are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ e \\ \mathbf{B} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} - \mathbf{B}\mathbf{B} + p_T\mathbf{I} \\ (e + p_T)\mathbf{u} - (\mathbf{u}\cdot\mathbf{B})\mathbf{B} \\ \mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B} \end{bmatrix}. \tag{5.13}$$

The source term for rotational effects, $\mathbf{Q}$, and the vector containing terms arising from expressing Faraday's law in divergence form, $\mathbf{S}$, are given by

$$\mathbf{Q} = \begin{bmatrix} 0 \\ -\rho\left[\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right] - 2\rho\left(\boldsymbol{\Omega} \times \mathbf{u}\right) \\ -\rho\mathbf{u}\cdot\left[\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})\right] \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{S} = -\begin{bmatrix} 0 \\ \mathbf{B} \\ \mathbf{u}\cdot\mathbf{B} \\ \mathbf{u} \end{bmatrix} \nabla \cdot \mathbf{B}. \tag{5.14}$$

Here, $\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})$ is the centrifugal force and $2\,\boldsymbol{\Omega} \times \mathbf{u}$ is the Coriolis force. The source term, $\mathbf{S}$, is associated with the divergence constraint as proposed by Powell [161].

The preceding MHD equations are solved in a non-dimensional form using the parallel finite-volume block-based AMR framework described in Chapter 2. The non-dimensional variables are related to their dimensional counterparts by

$$\rho = \tilde{\rho}/\rho_0, \tag{5.15}$$

$$\mathbf{u} = \tilde{\mathbf{u}}/a_0, \tag{5.16}$$

$$p = \tilde{p}/p_0, \tag{5.17}$$

$$\mathbf{B} = \tilde{\mathbf{B}}/B_0, \tag{5.18}$$

$$\boldsymbol{\Omega} = \tilde{\boldsymbol{\Omega}}(l_0/a_0), \tag{5.19}$$

$$t = \tilde{t}/\tau_0, \tag{5.20}$$

$$\mathbf{x} = \tilde{\mathbf{x}}/l_0. \tag{5.21}$$

Here, the variables $\tilde{\rho}$, $\tilde{\mathbf{u}}$, $\tilde{p}$, $\tilde{\mathbf{B}}$, $\tilde{\boldsymbol{\Omega}}$, $\tilde{t}$ and $\tilde{\mathbf{x}}$ represent the dimensionalized values of density, velocity, pressure, magnetic field, angular velocity, time and position vector in SI units whereas ρ_0, a_0, p_0, B_0, τ_0 and l_0 are constants used for non-dimensionalizing them. Additionally, $\rho_0 = 1$ cm^{-3} m_p where $m_p = 1.672 \times 10^{-27}$ kg is the mass of a proton. The length scale is given by $l_0 = 6.96 \times 10^8$ m, taken to be the radius of the Sun. The velocity scale is given by $a_0 = l_0/\tau_0 = 193.333 \times 10^3$ m/s where the time scale is taken to be $\tau_0 = 1$ hr $= 3600$ s. The value used to normalize the magnetic field is given by $B_0 = \sqrt{\mu_0 \rho_0 {a_0}^2} = 8.8642 \times 10^{-9}$ T where $\mu_0 = 4\pi \times 10^{-7}$ H/m is the magnetic permeability of free space. The value used to normalize the pressure is given by $p_0 = \rho_0 {a_0}^2 = 6.252 \times 10^{-11}$ Pa. The angular velocity of the Sun is taken to be $\tilde{\boldsymbol{\Omega}} = (2\pi/27.27)$ rad/day $\hat{\mathbf{k}} = 2.67 \times 10^{-6}$ rad/s $\hat{\mathbf{k}}$. The xy-plane of the computational domain is taken to be the Sun's ecliptic plane. It must be noted that the above non-dimensionalization follows that adopted previously by Shiota *et al.* [245, 246] and is only used for the solar wind framework and not for other MHD cases described in this thesis.

5.2.4 Model Coupling and Boundary Conditions

In order to determine the solar wind solution for the coupled data-driven model, the radial component of the photospheric magnetic field $B_r(R_{\text{Sun}}, \theta, \phi)$ is obtained from actual solar magnetograms and is used as the inner boundary condition for the coronal magnetic field. The scalar potential from Equation (5.4) is taken to be a constant value at the inner boundary $r = R_{\text{Sun}}$. The PFSS model is then used to obtain the 3D coronal magnetic field in the region $R_{\text{Sun}} \leq r \leq R_s = 2.5R_{\text{Sun}}$. As noted earlier, the PFSS model assumes a purely radial magnetic field at the source surface $r = R_s$. The radial magnetic field obtained from the PFSS model is then used as the boundary condition for the SCS model which is used to evaluate the coronal magnetic field in the region $R_s = R_{cp} = 2.5R_{\text{Sun}} \leq r \leq 25R_{\text{Sun}}$. Finally, the coronal magnetic field obtained from the

SCS model at $r = 25R_{\text{Sun}}$ is used as the inner boundary condition for the ideal MHD model and solution which is taken to extend over the region $25R_{\text{Sun}} \leq r \leq 225R_{\text{Sun}}$.

At $25R_{\text{Sun}}$, the solar wind speed is assumed to have already achieved its asymptotic value and is supersonic and super-Alfvénic. Hence, a fixed boundary condition can be used here for all variables required by the MHD model. The magnetic field is available from the SCS model. However, the plasma density, velocity and pressure at this inner boundary must be obtained using other appropriate models. The Wang-Sheeley-Arge (WSA) [230–236] model is used here to correlate solar wind speed, V_{SW}, to the magnetic field. In the WSA model, V_{SW} depends on the flux tube expansion factor given by

$$f_s = \frac{|\mathbf{B}(R_{\text{Sun}})|}{|\mathbf{B}(R_s)|}\frac{R_{\text{Sun}}^2}{R_s^2}, \tag{5.22}$$

and the angular separation, θ_b, between an open magnetic field line foot print and coronal hole boundary at the photosphere. A general form of the relation between V_{SW}, f_s and θ_b can be expressed as

$$V_{SW} = a_1 + \frac{a_2}{(1+f_s)^{a_3}}\left[a_4 - a_5 \exp\left\{-\left(\frac{\theta_b}{a_6}\right)^{a_7}\right\}\right]^{a_8}, \tag{5.23}$$

where a_1-a_8 are empirically evaluated coefficients. These coefficients are in general tunable parameters that depend on the magnetogram source used to derive the coronal magnetic field and calculation parameters.

The boundary values for the plasma density and temperature are obtained similarly using an empirical model derived from the Helios solar wind measurements [250, 251] through a least-squares fitting process and given by

$$n(V_{SW}) = 4\left\{62.98 + 866.4\left(\frac{V_{SW}}{100} - 1.549\right)^{-3.402}\right\} cm^{-3}, \tag{5.24}$$

$$T(V_{SW}) = 4^{\gamma-1}\left\{-0.455 + 0.1943\frac{V_{SW}}{100}\right\} 10^6\ K. \tag{5.25}$$

Although Hayashi *et al.* [250] proposed the relations given in Equations (5.24) and (5.25) for use at $r = 50R_{\text{Sun}}$, the same relations are used here at $r = 25R_{\text{Sun}}$. This is done by assuming a constant solar wind speed from $r = 25R_{\text{Sun}}$ to $r = 50R_{\text{Sun}}$, and thus the density is multiplied by a factor of $4 = (50/25)^2$. The temperature is also modified under the assumption of polytropic expansion of the plasma with increasing radius. The mass density can be obtained from the number density by multiplying with the proton mass m_p and the plasma thermal pressure can be obtained from the number density and temperature using the ideal gas law, $p = nk_BT$, where $k_B = 1.38 \times 10^{-23}$ J K^{-1} is the Boltzmann constant. EUHFORIA, on the other hand, also uses the WSA model for the solar wind speed, but assumes constant values for kinetic energy density and plasma thermal pressure at the inner boundary of the MHD model. As a result, the plasma number density is inversely proportional to the square of the solar wind speed in this model.

For the ideal MHD model, a suitable coordinate transformation and linear interpolation is required to map the solution obtained from the inner PFSS-SCS model on a (θ, ϕ, t) grid to the inner boundary mesh for the outer global MHD model. Additionally, we must also take into account the Sun's rotation for the boundary conditions at $r = R_{\text{inner}} = 25R_{\text{Sun}}$. For a calculation performed in the Sun's corotating frame, the variables at the inner boundary of the MHD model are fixed. The latitudinal component of the solar wind velocity is assumed to be zero, i.e., $V_\theta = 0$ while the radial component of the solar wind velocity V_r is obtained from Equation (5.23). In the corotating frame, the longitudinal component is assumed to be zero, i.e., $V_\phi = 0$. The radial component of the magnetic field at the inner boundary of the MHD model is obtained from the SCS model. The latitudinal and longitudinal components are assumed to be zero, i.e., $B_\theta = 0$ and $B_\phi = 0$.

5.3 Numerical Results for Representative Solar Wind

The capabilities of the proposed data-driven solar-wind framework are now demonstrated using a representative data set obtained from the GONG magnetograms. As a first example described in Section 5.3.1 below, the predicted steady-state solution for the structure of the ambient background solar wind is obtained using a single set of magnetogram data. The inexact Newton's method discussed in Section 2.2 is used to obtain the steady-state solution. In the second example described in Section 5.3.2, multiple successive magnetogram data sets are used as a time-varying inner boundary condition to obtain an unsteady solution for the resulting solar wind. Solar wind speed, density and magnetic field predictions are obtained at the Earth and compared with satellite observations. For this unsteady simulation, the second-order explicit Runge-Kutta scheme described in Section 2.4 is used in the time-marching procedure.

The mesh used for both simulations is a cubed-sphere mesh [67,68] initially composed

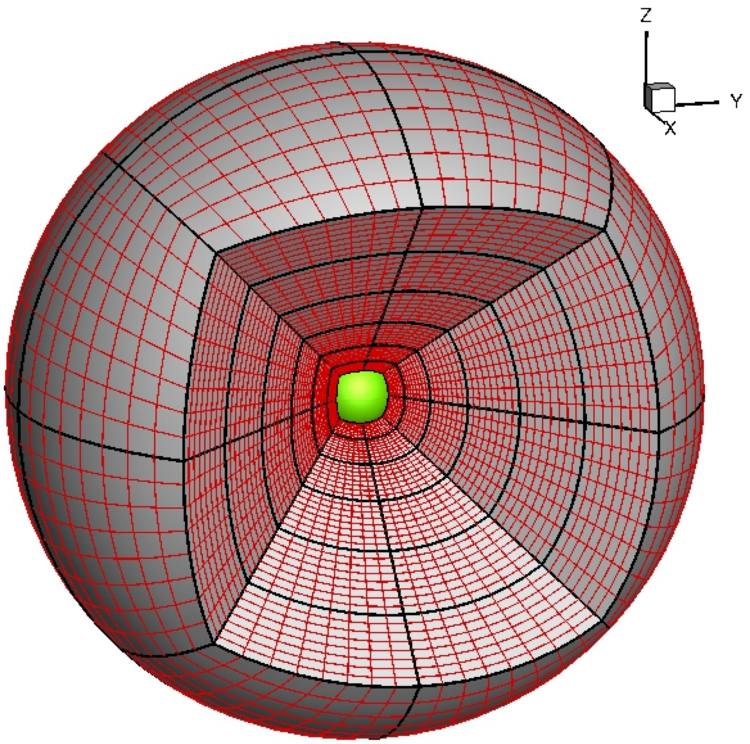
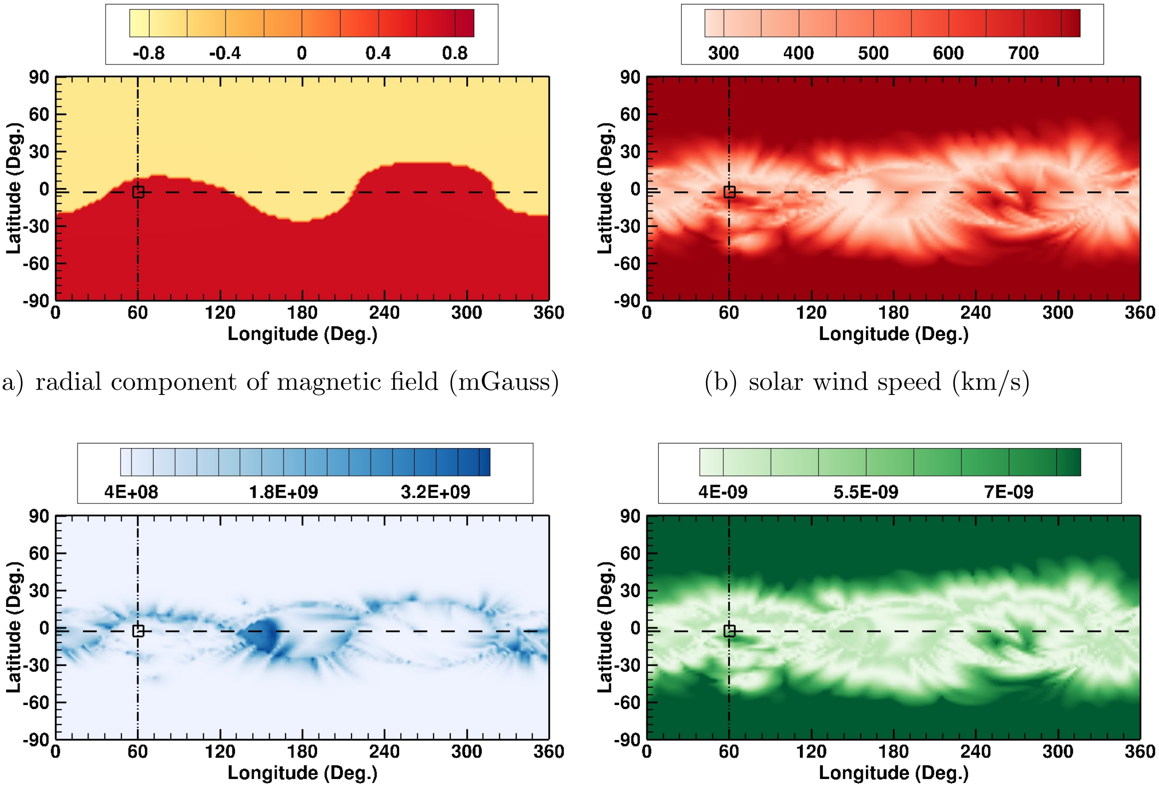


Figure 5.2: Portion of the initial mesh used for the steady state background solar wind simulation and the time-accurate simulation used for solar wind prediction at the Earth. The inner boundary is shaded in green. The thicker black lines are the block edges and the thinner red lines are the cell edges. Each block contains $8 \times 8 \times 8 = 512$ cells. The total number of blocks is 144. The total number of cells is $144 \times 512 = 73,728$.

of 144 blocks with each block containing $8 \times 8 \times 8 = 512$ cells. A portion of the initial mesh is shown in Figure 5.2. The gradient of the magnetic field weighted with the square of the radial outward distance from the center of the Sun, $\nabla(|\mathbf{B}|r^2)$, was used as the criterion for mesh adaptation, in order to accurately capture the current sheet in both cases. The weight r^2 is to account for the fact that in a supersonic spherical outflow with constant velocity, the magnetic field strength rapidly decreases in proportion to $1/r^2$ in the radially outward direction. Blocks having gradients higher than a predefined refinement threshold are flagged for refinement while those with gradients below a predefined coarsening threshold are flagged for coarsening. A maximum refinement level of 7 is imposed on both simulations, with level 0 corresponding to the initial mesh.



(a) radial component of magnetic field (mGauss)

(b) solar wind speed (km/s)

(c) density (count/m^3)

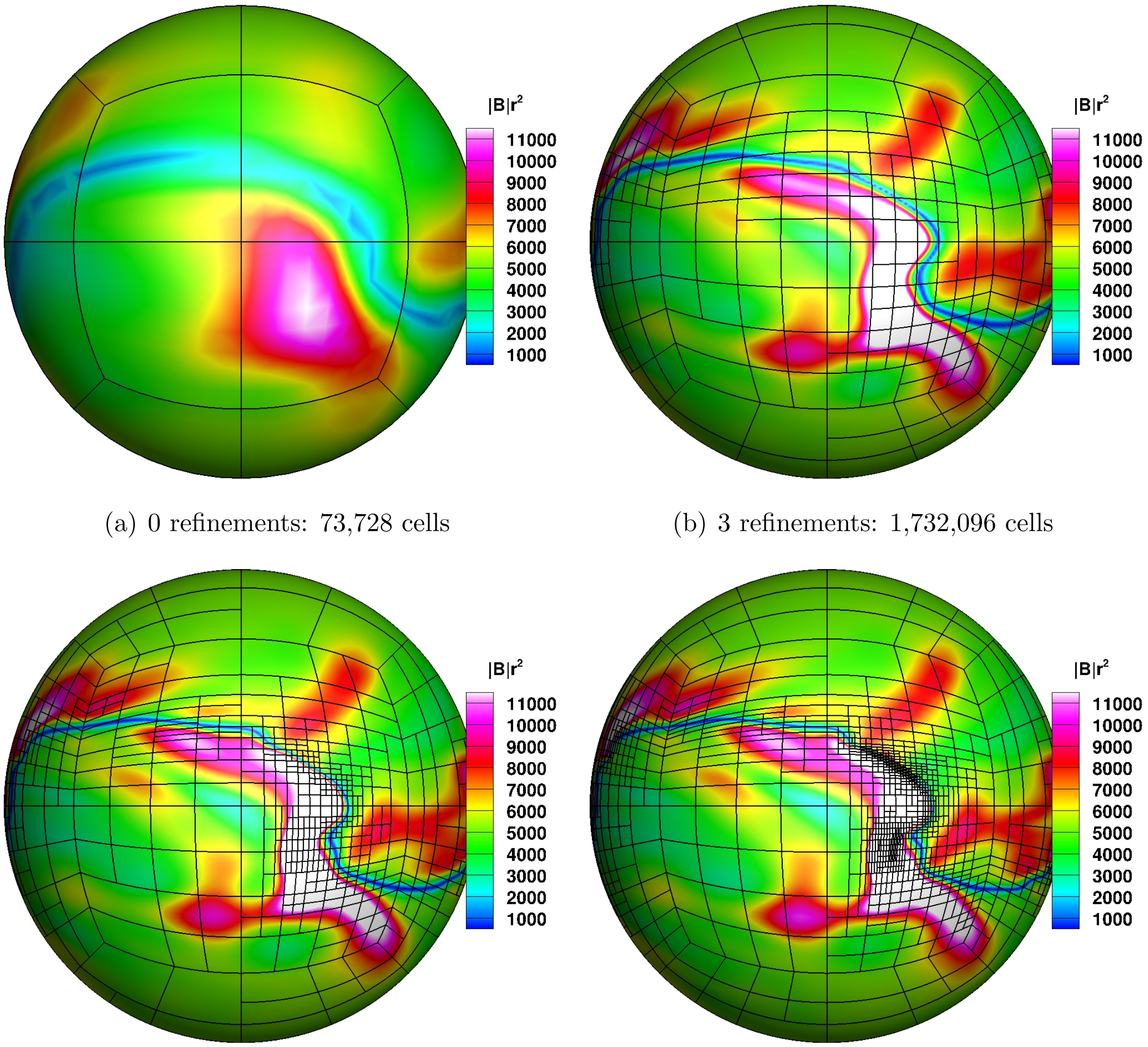
(d) pressure (Pa)

Figure 5.3: Boundary conditions at $r = 25R_{\text{Sun}}$ for the ambient solar wind simulation using the 12th February 2007 magnetogram data set.

5.3.1 Steady Background Solar Wind

For the steady-state background solar-wind simulation, a single GONG magnetogram obtained on 12th February 2007 at 11 am was used as an input for the coronal model calculation. Two-dimensional plots of the radial component of the magnetic field, solar wind speed, plasma density and thermodynamic pressure for this case at the inner boundary, $r = 25R_{\text{Sun}}$, are shown in Figure 5.3. The current sheet across which the magnetic field rapidly changes direction can be distinctively seen in Figure 5.3(a). The higher density plasma originates from the region of the current sheet as can be seen in Figure 5.3(c). The fast high pressure solar wind originates from the higher latitudes closer to the poles as can be seen in Figures 5.3(b) and 5.3(d).

Seven successive refinements are performed on the initial mesh and a converged solution is calculated after each refinement using the inexact Newton's method described in Section 2.2. The final mesh after 7 refinements contains a total of 45,799 blocks and



(a) 0 refinements: 73,728 cells

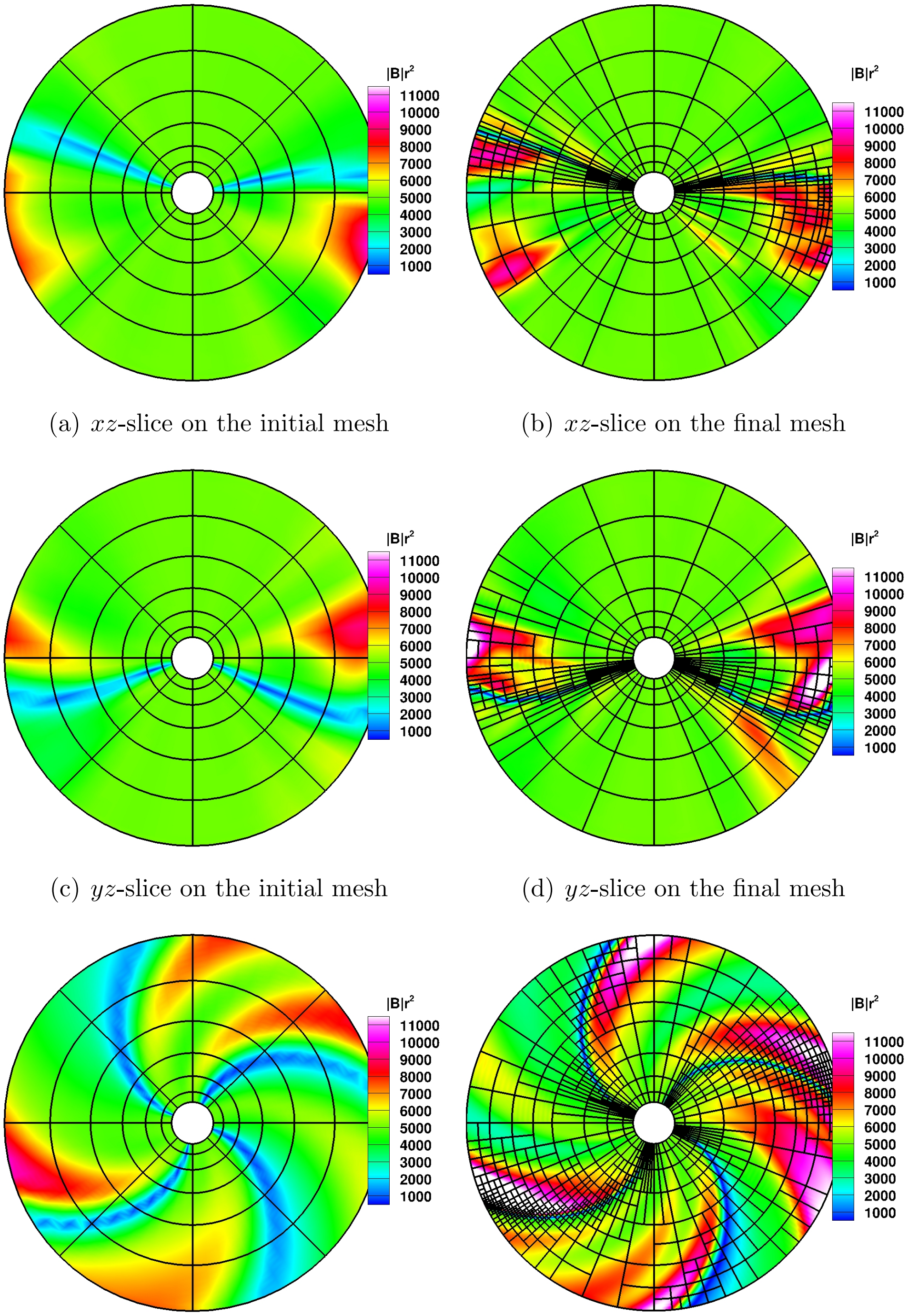
(b) 3 refinements: 1,732,096 cells

(c) 5 refinements: 5,197,312 cells

(d) 7 refinements: 23,449,088 cells

Figure 5.4: ($|\mathbf{B}|r^2$) contours at the outer boundary $r = 225R_{\text{Sun}}$ of the computational domain after 0, 3, 5 and 7 refinements, for the ambient solar wind simulation. The black lines are the block boundaries.

correspondingly, $45,799 \times 512 = 23,449,088$ cells, ≈ 23.5 million cells. Figure 5.4 shows the ($|\mathbf{B}|r^2$) contours at the outer boundary $r = 225R_{\text{Sun}}$ of the computational domain on intermediate meshes obtained after each successive adaptive mesh refinement. With each level of mesh refinement, there is an increase in the number of blocks in the vicinity of the current sheet. The proposed anisotropic AMR proves to be both very effective and efficient in capturing the discontinuous nature and complex topology of the current sheet by refining extensively only across the current sheet while keeping the grid resolution low



(a) xz-slice on the initial mesh

(b) xz-slice on the final mesh

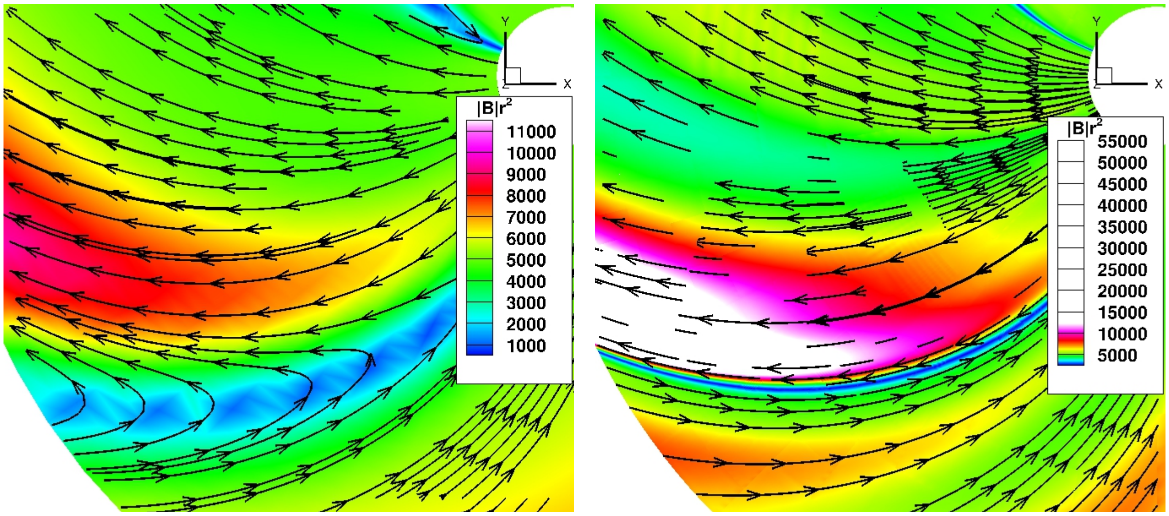
(c) yz-slice on the initial mesh

(d) yz-slice on the final mesh

(e) xy-slice on the initial mesh

(f) xy-slice on the final mesh

Figure 5.5: ($|\mathbf{B}|r^2$) contours on $xz-$ (top), $yz-$ (middle) and $xy-$ (bottom) slices on initial (73,728 cells) (left) and final (23,449,088 cells) (right) meshes showing capturing of the current sheet, for the ambient solar wind simulation. The black lines are the block boundaries.



(a) (b)

Figure 5.6: Portion of the xy- (ecliptic) plane showing ($|\mathbf{B}|r^2$) contours and magnetic field lines on initial (73,728 cells) (left) and final (23,449,088 cells) (right) meshes, for the ambient solar wind simulation using the 4^{th} July 2017 magnetogram data set.

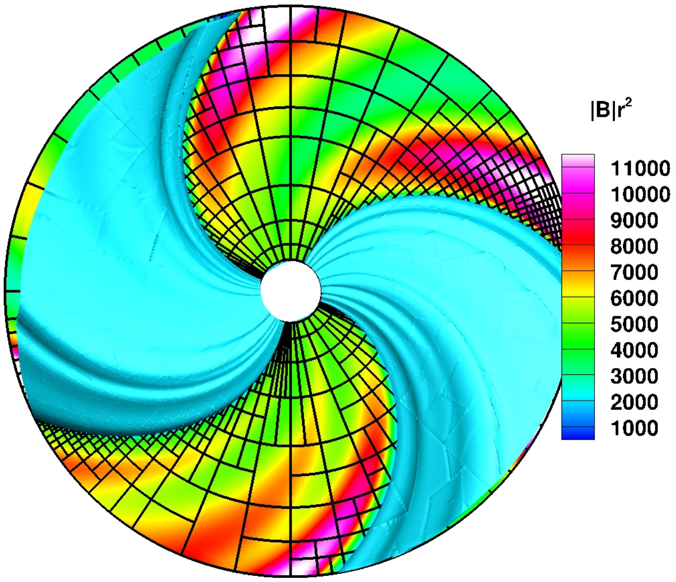


Figure 5.7: ($|\mathbf{B}|r^2$) contours on the xy- (ecliptic) plane and current sheet isosurfaces on the final mesh ($\approx$ 23.5 million cells) obtained after 7 successive anisotropic adaptive mesh refinements, for the ambient solar wind simulation using the 4^{th} July 2017 magnetogram data set.

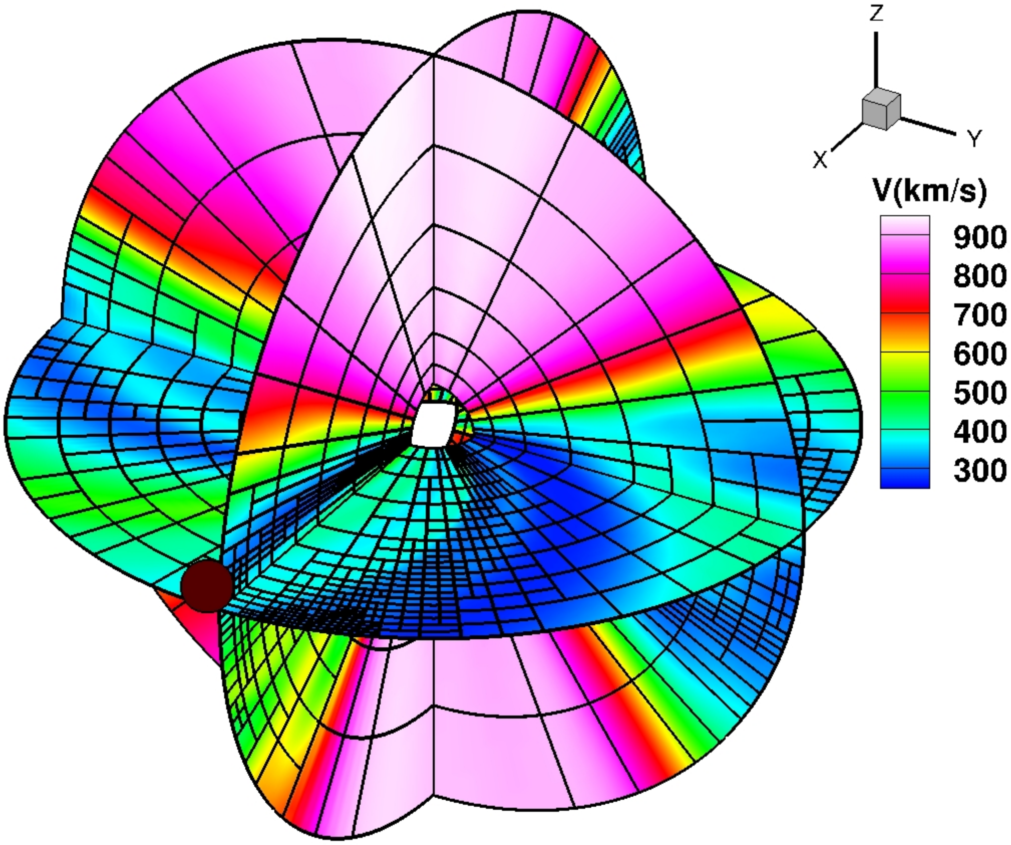


Figure 5.8: Solar wind speed profile (km/s) for the ambient solar wind simulation on the final mesh ($\approx$ 23.5 million cells) obtained after 7 successive anisotropic adaptive mesh refinements. The black lines are the block boundaries. The brown spot shows the approximate location of the Earth.

along the current sheet surface. Figure 5.5 shows ($|\mathbf{B}|r^2$) contours and the comparison between the structure of the current sheet on xz-, $yz-$ and $xy-$ slices of the computational domain on the initial and final meshes. This comparison shows qualitatively how the current sheet is captured not only at the outer boundary but throughout the entire computational domain. Figure 5.6 shows a comparison between ($|\mathbf{B}|r^2$) contours and magnetic field lines on a portion of the xy- (ecliptic) plane on the initial and final meshes. On the initial mesh, these oppositely directed lines across the current sheet artificially reconnect due to numerical diffusion, contaminating and disrupting the predicted solar-wind solution. On the final mesh, these oppositely directed lines do not exhibit reconnection and are predicted to lie very close together on either side of the narrow current sheet, accurately capturing the magnetic field discontinuity across the current

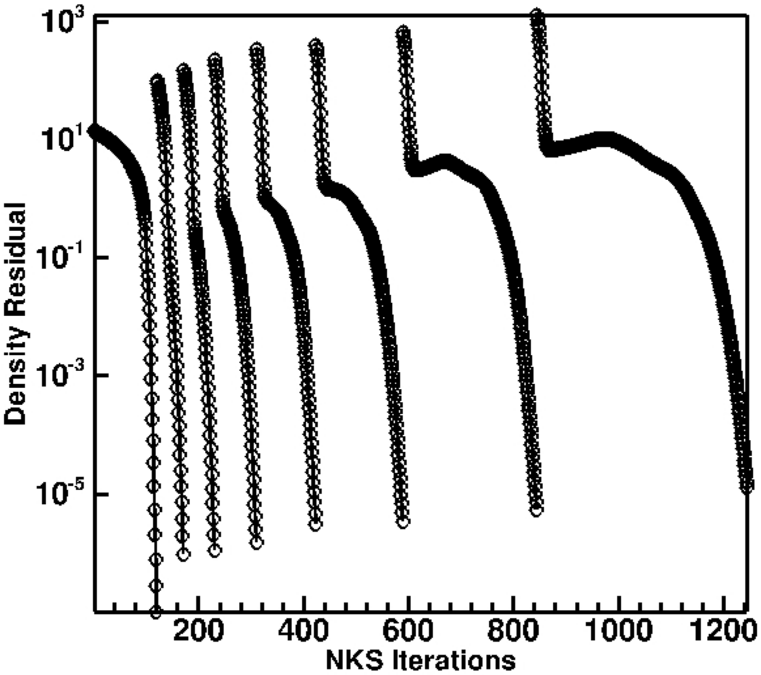


Figure 5.9: Density residual convergence history for the ambient solar wind simulation.

sheet. Without the anisotropic AMR approach, it would be extremely computationally expensive to obtain such high quality ideal MHD solutions of the solar wind and a much higher number of grid points would be required.

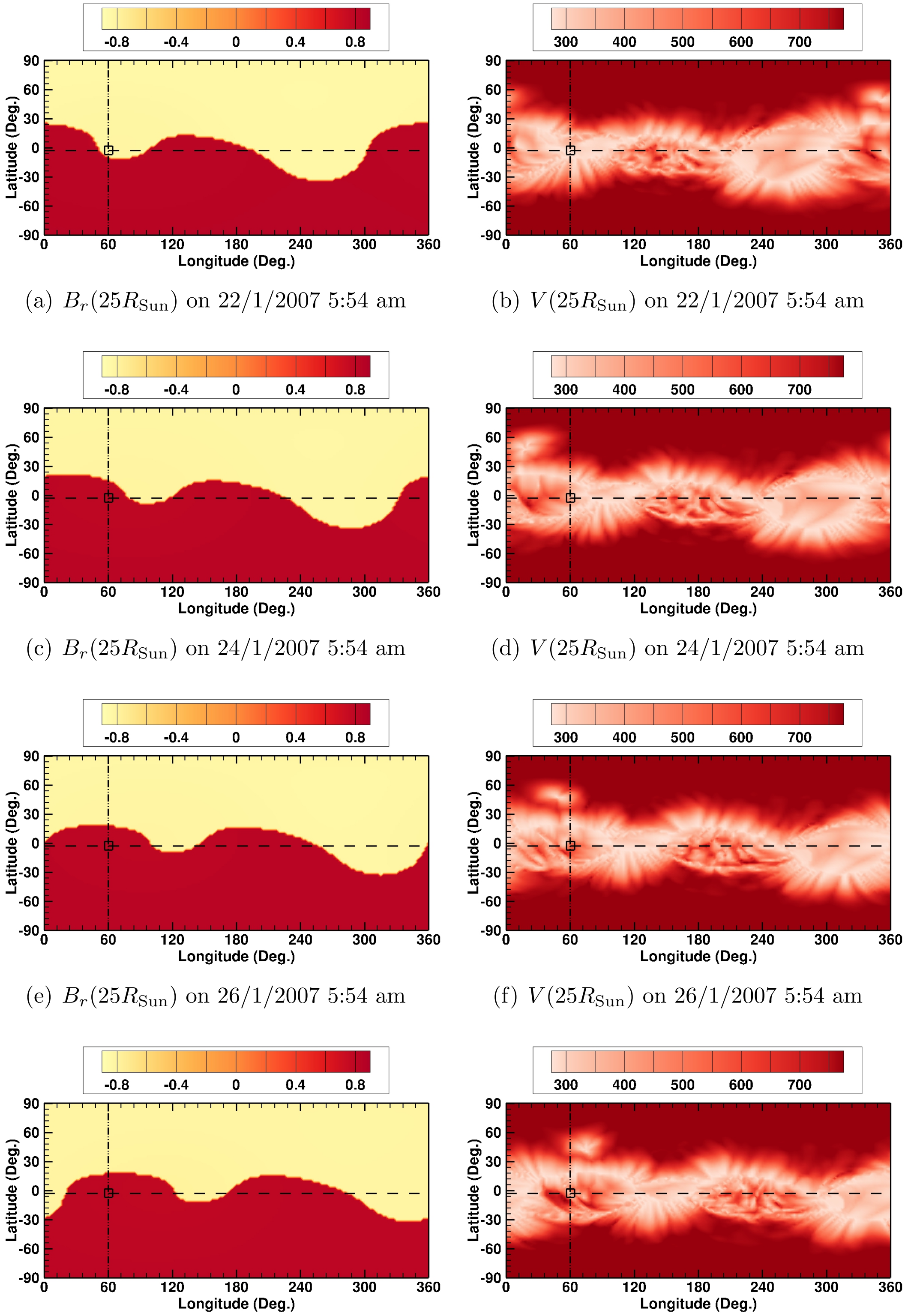
It must be noted that the plasma density, ρ, and magnetic field strength, $|\mathbf{B}|$, decrease with increasing radial distance from the Sun, being inversely proportional to the square of the radial distance, r^2. Hence, the quantities ($|\mathbf{B}|r^2$) and (ρr^2) stay constant along the spiral path the plasma follows from its point of origin at the inner boundary to its point of exit at the outer boundary. Figure 5.7 shows ($|\mathbf{B}|r^2$) contours on the ecliptic plane on the final mesh and also an isosurface of the current sheet lying on the northern side of the plane. Solar wind velocity profiles obtained on the final mesh are shown on the three Cartesian planes in Figure 5.8. The slow and fast components of the solar wind can be distinctively seen here. The fast solar wind emerges mainly from the polar regions of the Sun while the slow solar wind emerges mainly from the region near the solar equator, for a period of low solar activity. The brown coloured spot in the figure depicts the approximate location of the Earth. The flow is supersonic throughout the computational domain and the magnitude of the solar wind speed remains constant along the spiral path of the plasma. This is due to the fact that the thermal pressure and magnetic pressure gradients are too small to cause any observable acceleration of the solar wind plasma. Additionally, the Parker Spiral [252, 253] can be clearly seen in the ecliptic xy-plane of the Sun-Earth system in both Figure 5.8 and Figure 5.7.

The entire simulation for the steady-state solar wind was carried out using a total of 2,048 processors for a relatively short wall clock time of just 3 hours. The parallel cluster used consisted of Intel Xeon E5540 cores operating at 2.53GHz, with 2 GB of memory per core. The convergence of the steady-state ideal MHD solution residual for density on the 8 successively refined meshes is given in Figure 5.9. The converged steady state solution has been obtained on the final mesh comprising 23.5 million cells in a total of 1,200 outer NKS iterations over all 8 meshes with an average of about 150 outer NKS iterations per mesh level. This NKS iteration count per mesh level is somewhat higher than the 15-40 iterations typically required for the flow cases considered in Chapter 4 due to the reduction in diagonal dominance of the system introduced by the rotating frame. Nevertheless, robust convergence with a residual reduction of more than eight orders of magnitude is achieved on each mesh in a relatively efficient manner.

5.3.2 Unsteady Solar Wind Driven by Solar Magnetic Field

As an example of an unsteady solar wind simulation, a sequence of 64 consecutive GONG magnetogram data sets was used as input for the coronal model calculation. These magnetograms were obtained over a span of 16 days starting from 18^{th} January 2007 to 2^{nd} February 2007. There are four magnetograms per day with a gap of typically 6 hours between two consecutive magnetograms. The first magnetogram was dated 18^{th} January, 5:54 am while the last magnetogram was dated 2^{nd} February, 11:54 pm.

Several 2D plots of the radial component of the magnetic field obtained at $r = 25R_{\text{Sun}}$ using the PFSS-SCS model on various magnetogram data sets and the corresponding solar wind speed obtained using the WSA model are shown in Figure 5.10. A lateral shift in the location of the current sheet can be clearly observed between consecutive magnetic field boundary conditions due to the Sun's rotation. A steady-state ambient solution corresponding to the first magnetogram was first obtained on the initial mesh and 7 adaptive mesh refinements were subsequently applied to obtain the starting initial background solar-wind solution. The proposed NKS solution was used to obtain this initial steady-state solution as described in the previous subsection. After this, the simulation was marched forward in time until the date of the last magnetogram. The inner boundary condition was continuously updated using linear interpolation between successive magnetograms. The explicit second-order Runge-Kutta scheme was used for



(a) $B_r(25R_{\mathrm{Sun}})$ on 22/1/2007 5:54 am

(b) $V(25R_{\mathrm{Sun}})$ on 22/1/2007 5:54 am

(c) $B_r(25R_{\mathrm{Sun}})$ on 24/1/2007 5:54 am

(d) $V(25R_{\mathrm{Sun}})$ on 24/1/2007 5:54 am

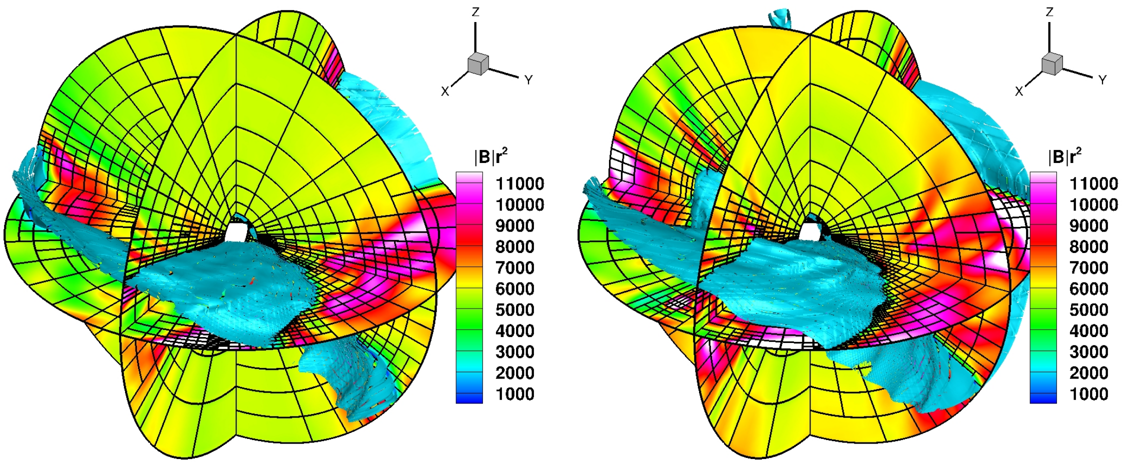
(e) $B_r(25R_{\mathrm{Sun}})$ on 26/1/2007 5:54 am

(f) $V(25R_{\mathrm{Sun}})$ on 26/1/2007 5:54 am

(g) $B_r(25R_{\mathrm{Sun}})$ on 28/1/2007 5:54 am

(h) $V(25R_{\mathrm{Sun}})$ on 28/1/2007 5:54 am

Figure 5.10: 2D plots of the radial component of the magnetic field obtained at the inner boundary $r = 25R_{\mathrm{Sun}}$ using the PFSS-SCS model on various magnetogram data sets and the corresponding solar wind speed obtained using the WSA model.



(a) (b)

Figure 5.11: ($|\mathbf{B}|r^2$) contours on the xy-, yz- and xz-slices and current sheet isosurfaces on adapted meshes showing the grid block boundaries (not individual cells) at 26^{th} January 5 pm (17,671,680 cells) and 31^{st} January 11 pm (31,553,536 cells), respectively, for the unsteady solar wind simulation. The black lines are the block boundaries.

time-marching the solution. Adaptive mesh refinement and coarsening was periodically applied to the solution after every 60 time-steps. With a CFL criterion of 0.3 used throughout the simulation, this would ensure that unsteady MHD waves should not traverse a given computational grid block between each successive mesh refinement.

Figures 5.11(b) and 5.11(a) show ($|\mathbf{B}|r^2$) contours and current sheet isosurfaces on intermediate adapted meshes at 26^{th} January 5 pm and 31^{st} January 11 pm, respectively. Figure 5.12 shows the solar wind velocity profile on the ecliptic plane at various instances in time. The brown spot again shows the position of the Earth at each time instance. In the corotating reference frame, the Earth moves in a clockwise direction at the rate of $(2\pi/27.27)$ rad/day. The time-evolution of the solar wind velocity, density and magnetic field strength at the Earth was obtained from the numerical MHD solution using the Earth's location and compared with ACE satellite data, as shown in Figure 5.13. The sharp increase in the solar wind speed observed in Figure 5.13 agrees qualitatively well with the ACE satellite observations and is also reflected in the snapshots shown in Figure 5.12 where the Earth is observed to move into the faster solar wind at a later time during the simulation. The rapid increases and peaks in the temporal variations of the plasma number density and magnetic field strength are in rather good agreement with the ACE satellite observations.

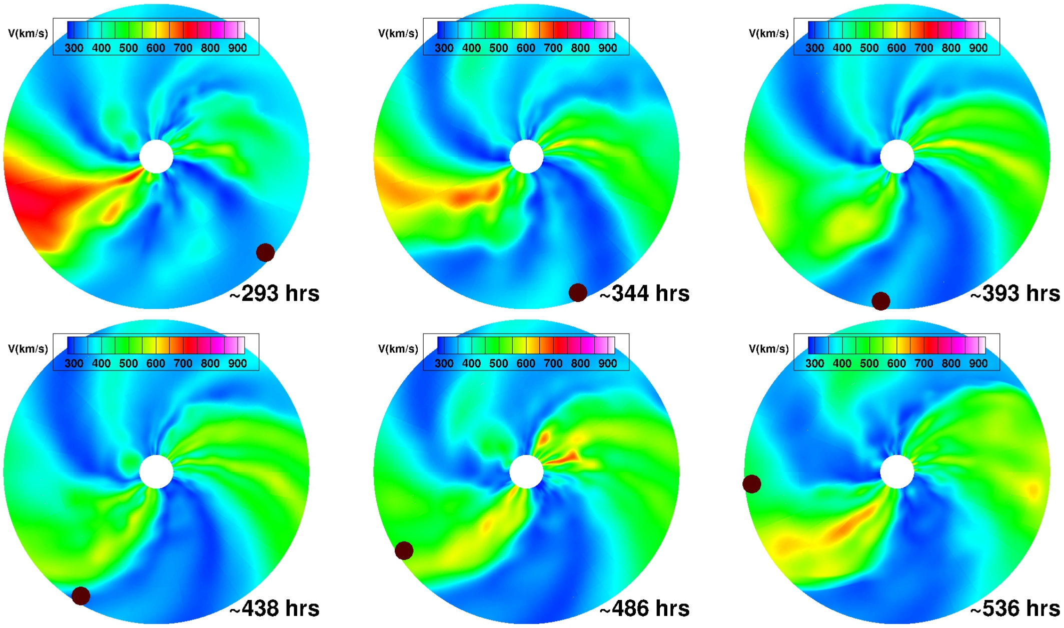
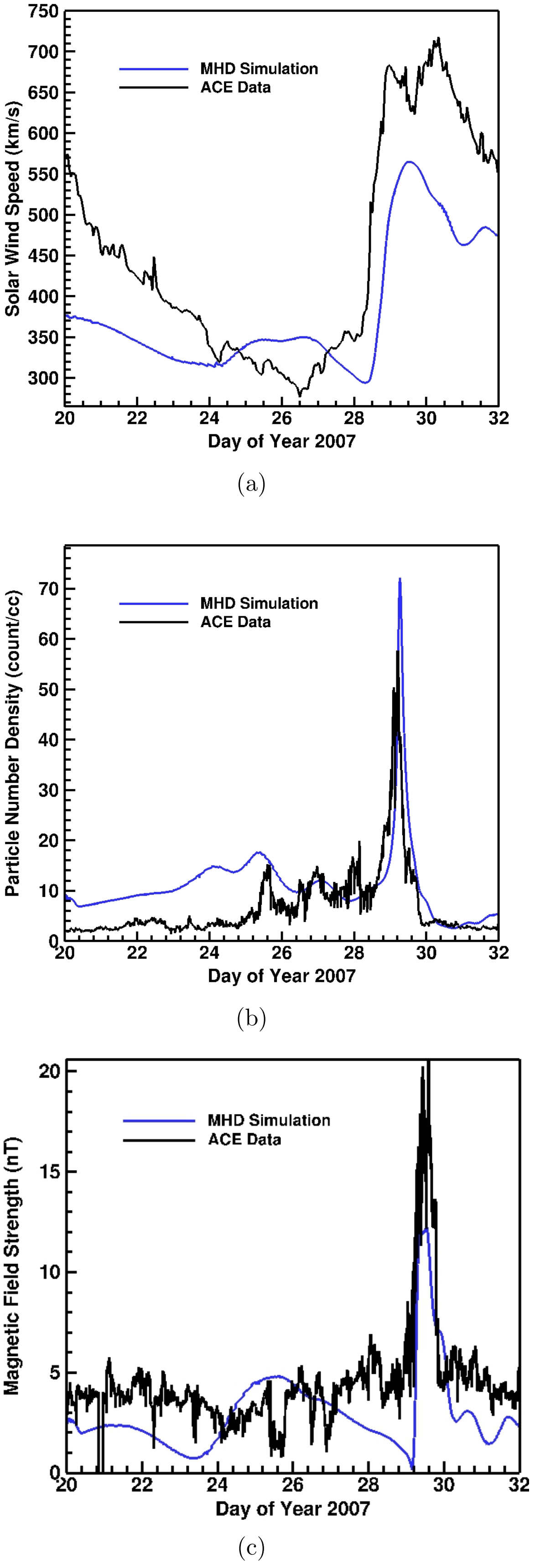


Figure 5.12: Solar wind speed contours on the ecliptic plane at various time instances. The dates and approximate times are 1) 21st January 9 am (8,776,192 cells) 2) 23rd January 12 pm (5,569,024 cells) 3) 25th January 1 pm (13,434,880 cells) 4) 27th January 10 am (9,246,720 cells) 5) 29th January 10 am (8,822,272 cells) 6) 31st January 12 pm (9,663,552 cells). The brown spot indicates the approximate location of the Earth.



(a)

(b)

(c)

Figure 5.13: Comparison between solar wind velocity, density and magnetic field forecasts at the Earth obtained using the numerical framework and ACE satellite observations.

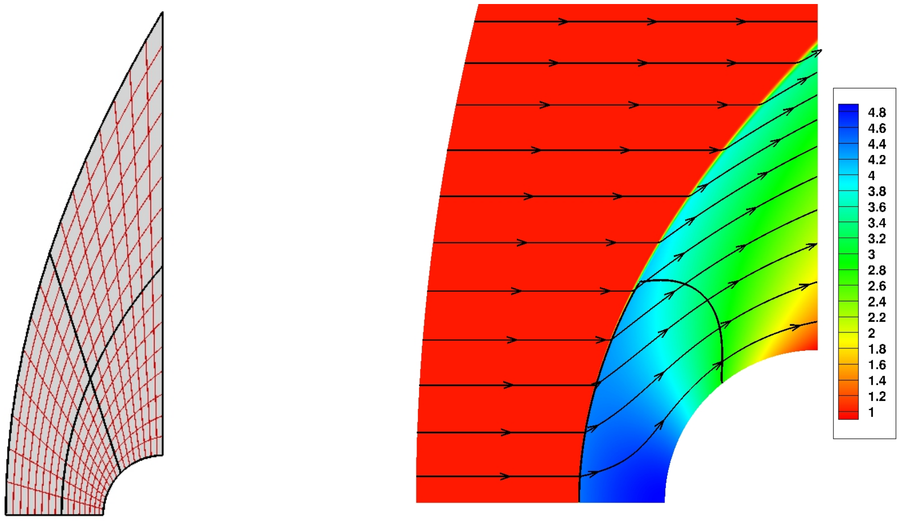
Chapter 6

Output-Based AMR for Steady Ideal MHD Flows

This chapter discusses the application of the proposed output-based anisotropic AMR scheme to several steady space physics-related flow simulations governed by the equations of ideal MHD. As with previous results for previous non-conducting flows, comparisons are made between the proposed output-based refinement approach and the uniform and gradient-based refinement approaches in terms of the mesh savings obtained for a certain accuracy level and the meshes obtained by the respective methods. Sections 6.1 to 6.3 consider the application of the scheme to standard test cases obtained from existing literature and, like the non-conducting flow problems considered previously, the problems considered include solutions that are smooth and continuous as well as those with complex shock structure. Finally, Section 6.4 considers the application of the proposed scheme to the steady-state solar wind obtained using the integrated solar wind-MHD framework described in Chapter 5.

6.1 Superfast Magnetized Flow Around a Perfectly Conducting Cylinder

In the first ideal plasma flow case considered herein, a uniform, planar, field-aligned and superfast magnetized plasma is intercepted by a perfectly conducting circular cylinder. This plasma flow problem was previously studied by De Sterck *et al.* [149] in their analysis of field-aligned MHD flows. As indicated, the incoming flow is field-aligned (i.e., the B_y and B_z components of the magnetic field are zero while only the B_x component is non-zero). The x-direction component of the magnetic field is taken to be $B_x = 0.1$. The



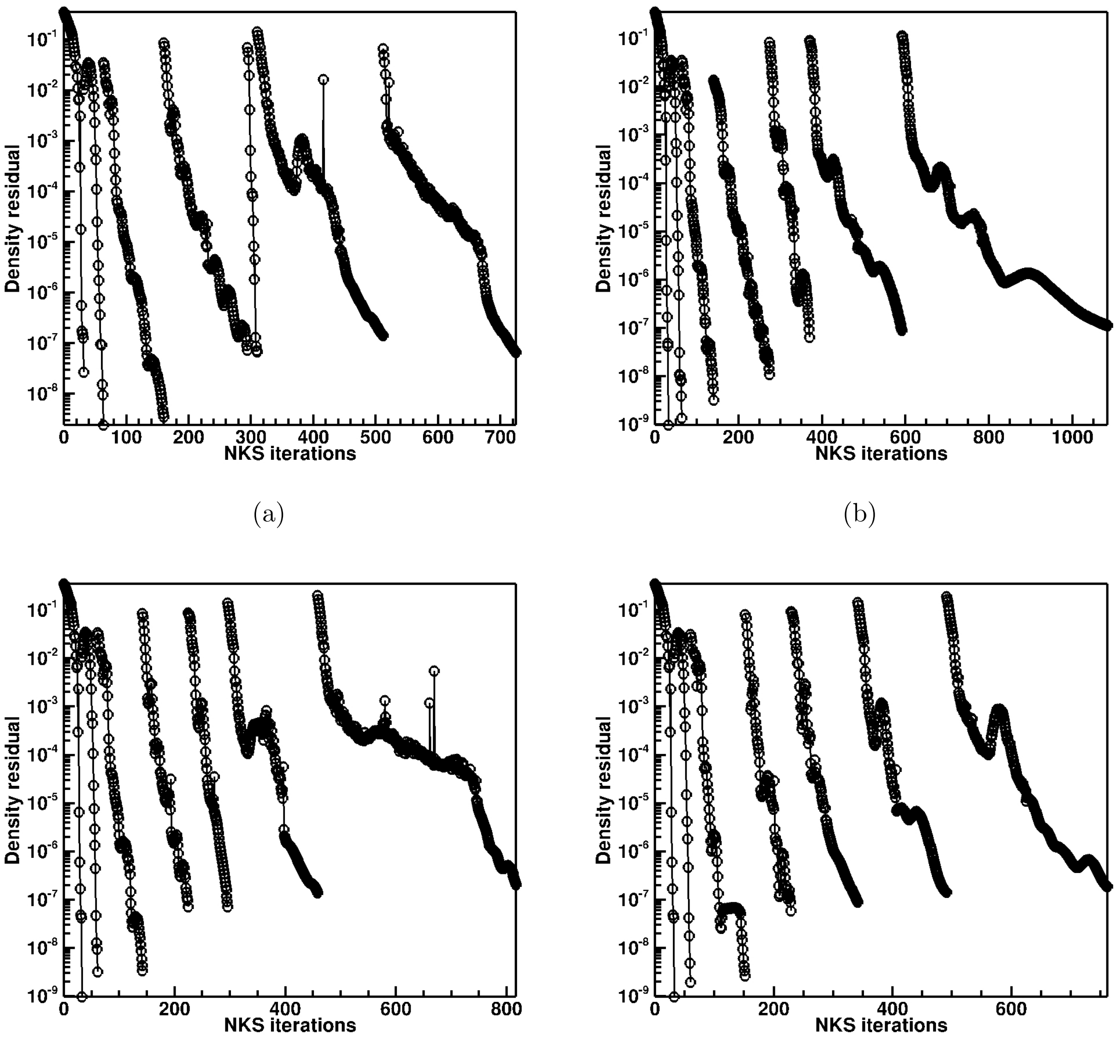
(a)

(b)

Figure 6.1: (a) Initial computational mesh for superfast magnetized flow around a perfectly conducting cylinder. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 4, the total number of cells per block is 8×8×4=256 and the total number of cells is 1,024. (b) Predicted density contours and magnetic field lines for this case after 5 uniform refinements. The single solid black contour indicates the sonic Mach 1 contour separating the subsonic and supersonic regions. Blocks: 4,096. Cells: 1,048,576.

thermodynamic pressure for the incoming flow is $p = 0.2$. The incoming flow density is taken to be equal to unity. The Alfvén wave speed for the incoming flow is given by $c_{Ax} = |B_x|/\rho = 0.1$. The inflow Alfvén Mach number is $M_A = 20$ while the inflow sonic Mach number is $M = 2\sqrt{3}$. Hence, the inflow velocity is given by $u_x = M_{Ax}c_{Ax} = 2$, while u_y and u_z are zero. The ratio of specific heats, γ, is assumed to be 5/3. The radius of the cylinder is taken to be 0.125 and the center of the cylinder is placed at the origin of the coordinate system. The initial mesh for this case, as shown in Figure 6.1(a), is composed of four blocks and the number of cells per block is $8 \times 8 \times 4 = 256$. A reflective boundary condition is used at the surface of the cylinder. A symmetry plane is used at the lower boundary of the computational domain. At the left boundary, the flow variables are held constant at the supersonic inflow conditions whereas at the right boundary, constant extrapolation is imposed for the outgoing flow.

A curved fast MHD bow shock is formed due to the obstruction of the uniform incoming flow by the perfectly conducting cylinder. At the bow shock, the magnetic field lines are refracted away from the shock normal as the plasma passes through the shock. Figure 6.1(b) shows the predicted density field and magnetic field lines for this



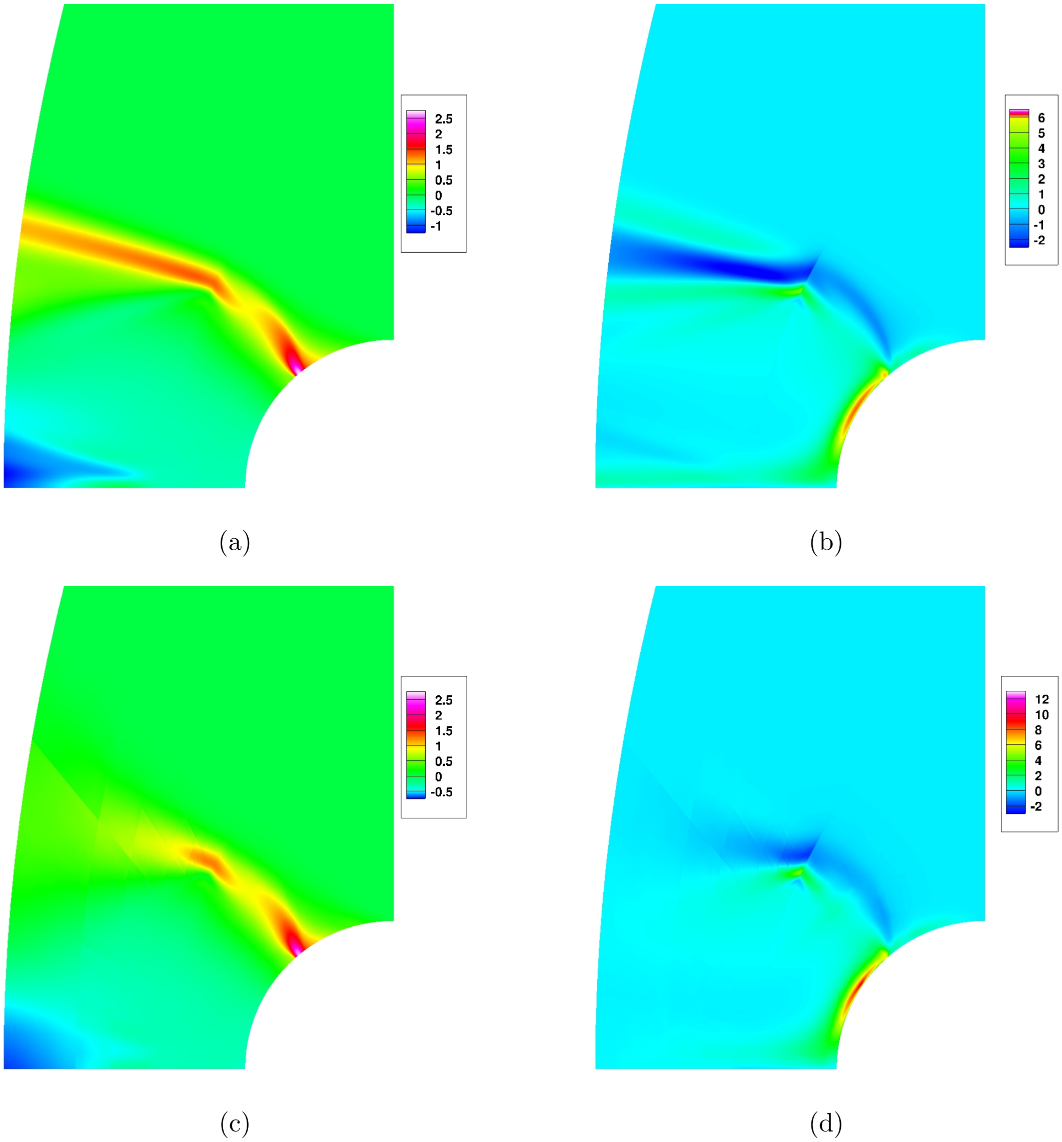
(a)

(b)

(c)

(d)

Figure 6.2: Density residual convergence history on the complete set of adaptive refined AMR meshes for steady superfast magnetized flow around a perfectly conducting cylinder: (a) uniform refinement; (b) gradient-based refinement; (c) output-based refinement using CC; (d) output-based refinement using ECC.



(a)

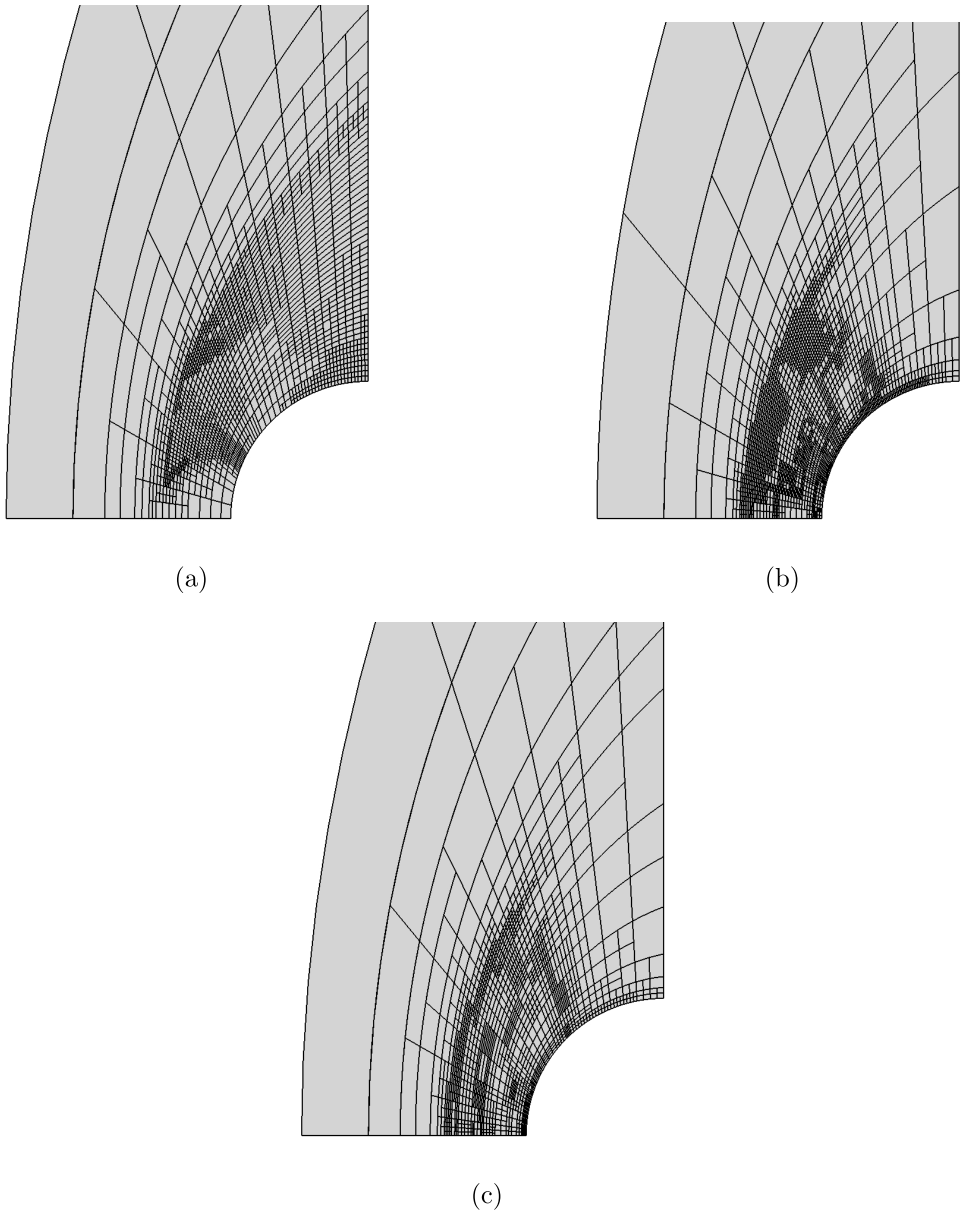
(b)

(c)

(d)

Figure 6.3: (a) The density and (b) B_y adjoint distributions for superfast magnetized flow around a perfectly conducting cylinder after 6 uniform refinements. Blocks: 16,384. Cells: 4,194,304. (c) The density and (d) B_y adjoint distributions for superfast magnetized flow around a perfectly conducting cylinder after 6 refinements using CC as refinement criterion. Blocks: 2,482. Cells: 635,392.

case obtained on a uniformly refined fine mesh. A single solid black line is also shown which indicates the sonic line ($M = 1$ boundary) following the shock. The regions downstream of the shock and above the sonic line and upstream of the shock are both superfast. The region lying between the shock and and the sonic line is subfast. For a detailed analysis of this MHD bow shock flow, the reader is referred to the previous

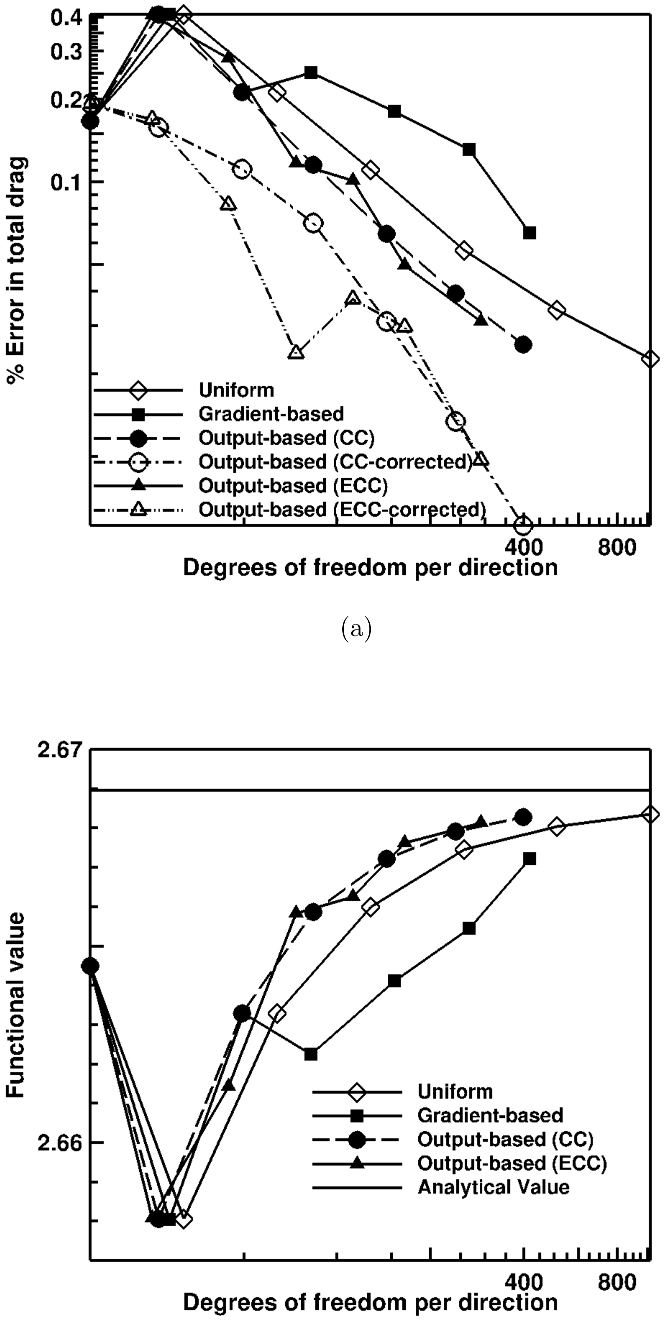

(a)

(b)

(c)

Figure 6.4: Adapted meshes showing the grid block boundaries (not individual cells) for superfast magnetized flow around a perfectly conducting cylinder: (a) After 6 refinements using density and magnetic field gradients as refinement criteria. Blocks: 2,731. Cells: 699,136. (b) After 6 refinements using CC as refinement criterion. Blocks: 2,482. Cells: 635,392. (c) After 6 refinements using ECC as refinement criterion. Blocks: 1,310. Cells: 335,360.

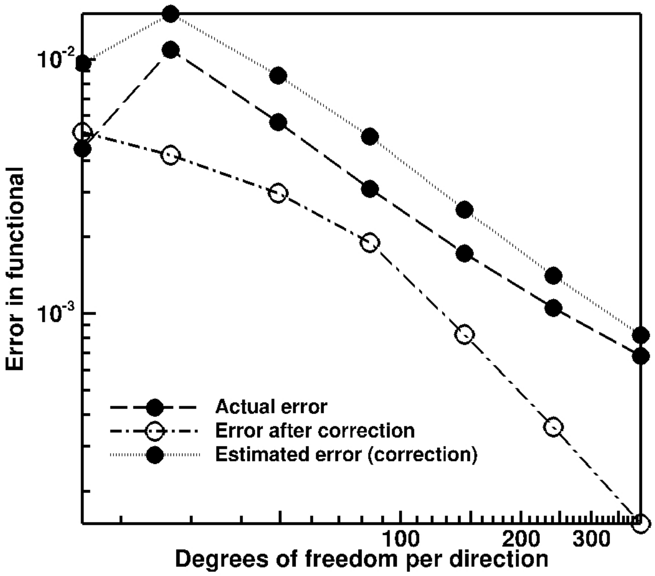
study by De Sterck *et al.* [150]. As for the non-conducting flow considered in this thesis, the proposed NKS scheme is quite effective in obtaining the steady state solutions for this case. The convergence of the steady-state solution density residual for this MHD



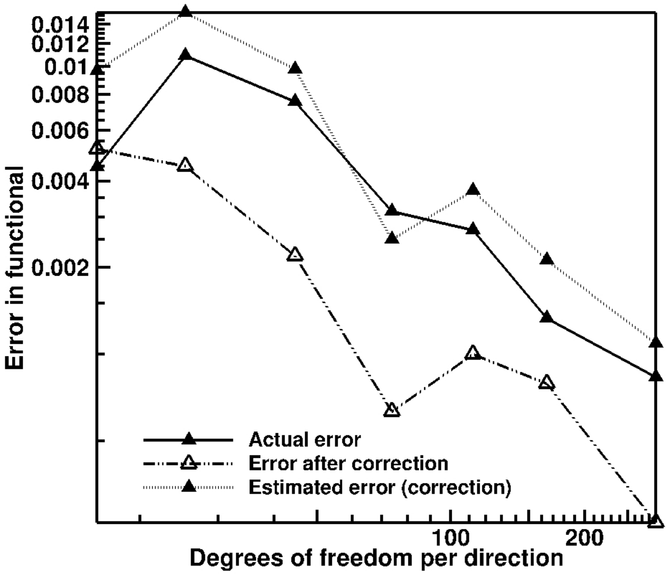
(a)

(b)

Figure 6.5: (a) Convergence of the percentage error in the predicted value of the functional and (b) convergence of the predicted value of the functional, as a function of the mesh size (number of degrees of freedom per direction) for various refinement techniques for the magnetized bow shock flow around a perfectly conducting circular cylinder.



(a)



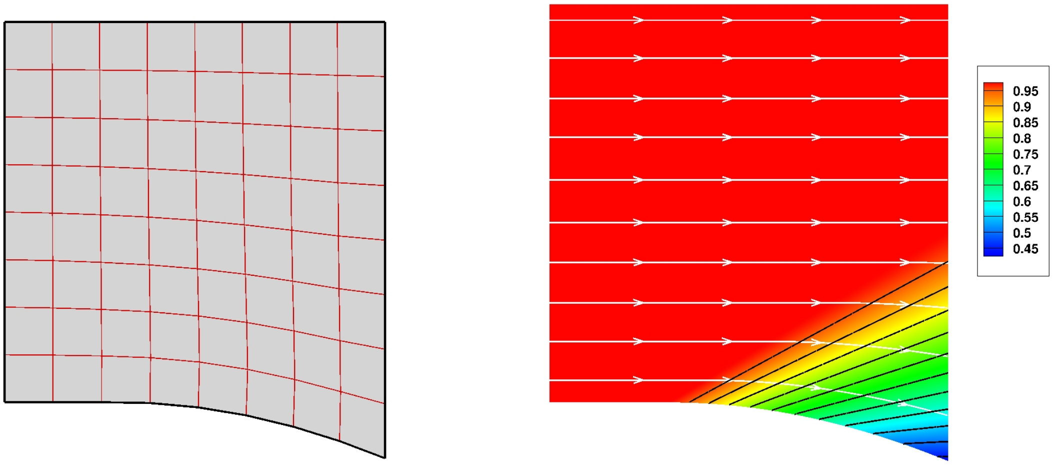
(b)

Figure 6.6: Convergence of the actual error, the error after adding the computable correction and the error estimate (absolute computable correction) as a function of the mesh size (number of degrees of freedom per direction) (a) for the CC method and (b) for the ECC method, for the magnetized bow shock flow.

bow-shock flow is given in Figure 6.2 for the complete set of four different refinement strategies considered in the thesis. Similar convergence behaviour was achieved for all of the conducting plasma flow cases considered here in this chapter.

The functional used for this field-aligned plasma flow is the total pressure drag which comprises both the thermal and magnetic components of the plasma pressure on the cylinder in the direction of the incoming flow. Figure 6.3 shows the predicted density and B_y adjoint solutions for this case corresponding to converged solution obtained following 6 uniform refinements of the initial mesh and after 6 refinements using the CC refinement criterion. Additionally, Figure 6.4 shows a comparison between meshes obtained after several adaptive refinements on the initial mesh, using various gradient-based and output-based refinement criteria. Figure 6.4(a) shows the mesh obtained after 6 gradient-based refinements of the initial mesh, based on the magnitude of the gradients in the magnetic field and density as refinement criteria. As for the non-conducting flow results of Chapter 4, the gradient-based approach focuses on refining the entire span of the fast MHD bow shock extending from the lower boundary to the outflow boundary. However, this is not required for an accurate value of the functional. Conversely, Figure 6.4(b) shows the anisotropic mesh obtained after 6 CC-based refinements of the initial mesh. This output-based method captures only a portion of the bow shock and regions downstream of the shock close to the cylinder. Moreover, Figure 6.4(c) shows that the anisotropically refined mesh obtained after 6 ECC-based refinements of the initial mesh has similar desirable features in terms of obtaining an accurate functional.

As for other previous non-conducting cases, a reference value of the total drag force on the cylinder was determined using a uniformly refined mesh, with a resolution finer than the highest mesh refinement level achieved for both the gradient-based and output-based adaptive mesh refinement techniques. The difference between the numerically computed value of the total drag force and this reference fine mesh solution was used as the measure of the error in the drag force. Figures 6.5(a) and 6.5(b) show plots of this percentage error and the predicted total drag against the total number of degrees of freedom per direction for this effectively two-dimensional plasma flow problem, i.e., $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells, respectively. For the output-based techniques, the error obtained after adding the computable correction to the functional is also shown in Figure 6.5(a) whereas the solid horizontal line in the closeup of Figure 6.5(b) represents the computed fine mesh reference value of the functional for this case (2.668961154). It is evident from the two figures that the gradient-based method manages to reduce the error up to a certain point and then fails to reduce it further, leading to a plateau in the convergence of the error as the mesh is refined. The uniform and output-based methods,

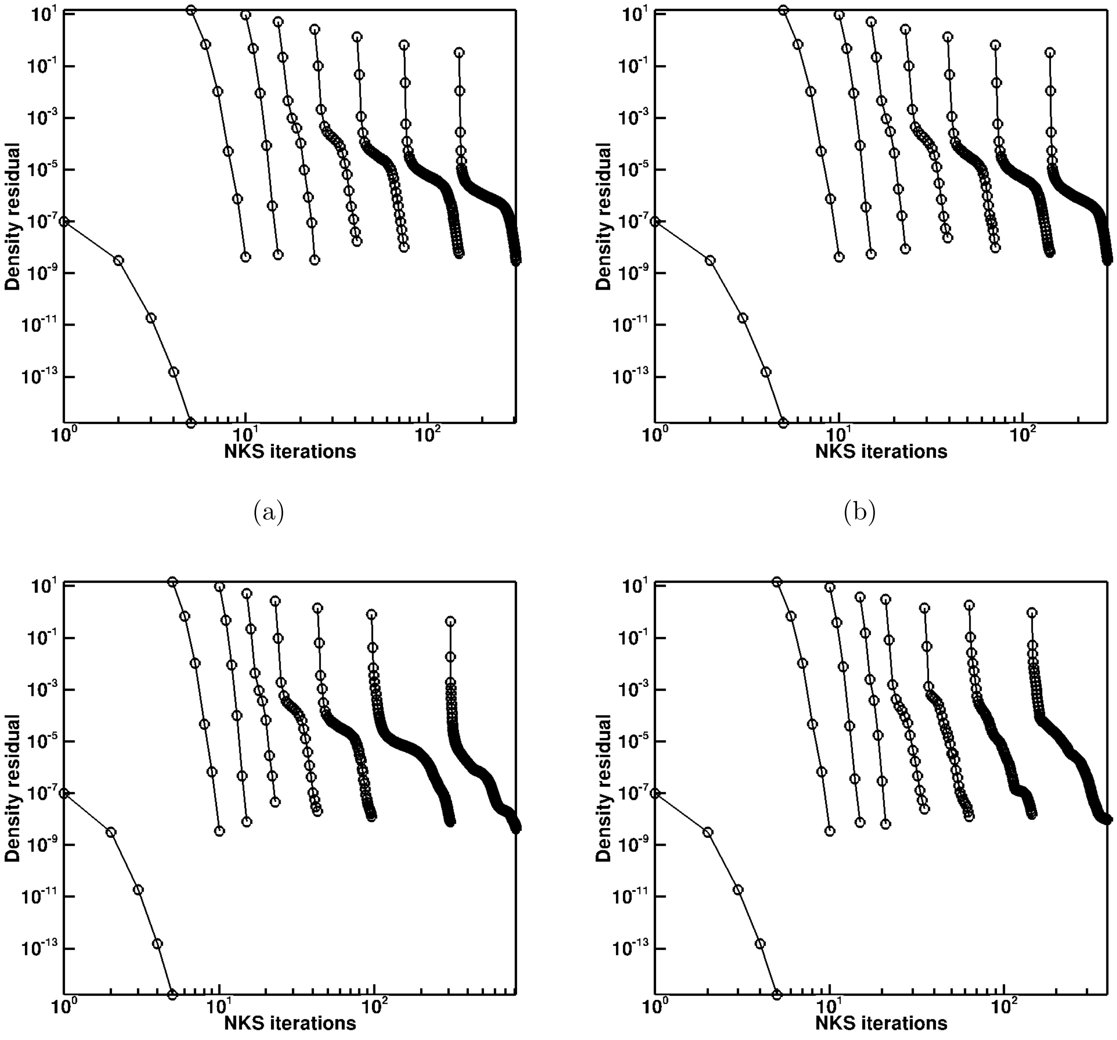


(a) (b)

Figure 6.7: (a) Initial computational mesh for steady plasma flow through an expanding tube. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 1, the total number of cells per block is 8×8×4=512 and the total number of cells is 512. (b) Predicted density contours and magnetic field lines for this case after 5 uniform refinements. Blocks: 1,024. Cells: 262,144.

on the other hand, manage to continuously reduce the error by almost 2 orders of magnitude. The output-based method employing CC as the refinement criterion achieves a mesh size reduction of 85% as compared to the uniform refinement approach, while the output-based method employing ECC achieves a reduction of almost 90%, at an accuracy level of ∼0.03%.

As further evidence of the convergence of the output-based methods, computed values of the actual error in the functional, corrected error, and estimated error or computable correction are shown in Figure 6.6 for both the CC- and ECC-based methods. The actual and estimated errors approach equality after several mesh adaptations. As expected, it can be observed that the values of the actual error and estimated error based on the computable correction approach each other as the mesh is refined for both the CC and ECC methods. Moreover, the error in the total drag force is reduced significantly after application of the computable correction to the value of the functional.



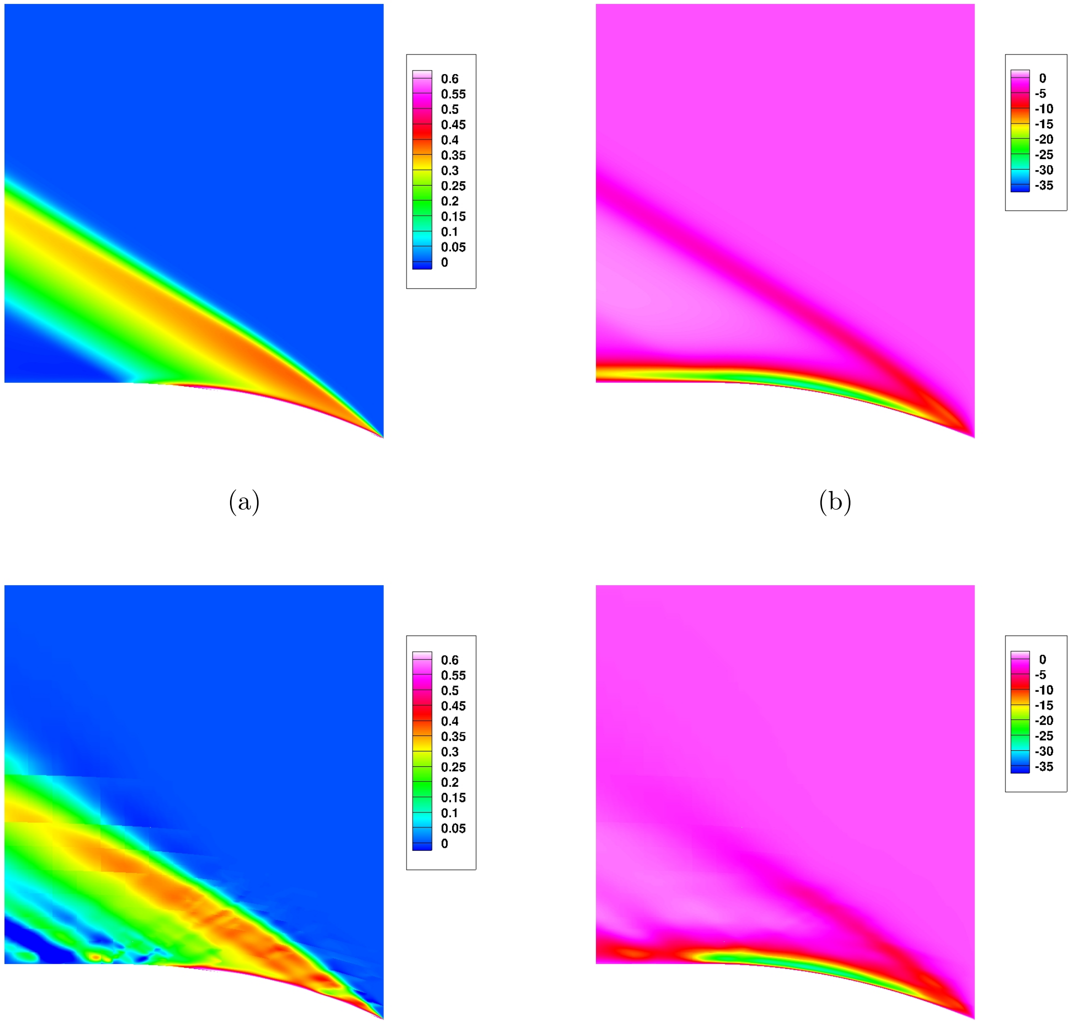
(a)

(b)

(c)

(d)

Figure 6.8: Density residual convergence history on the complete set of adaptive refined AMR meshes for steady plasma flow through an expanding tube: (a) uniform refinement; (b) gradient-based refinement; (c) output-based refinement using CC; (d) output-based refinement using ECC.



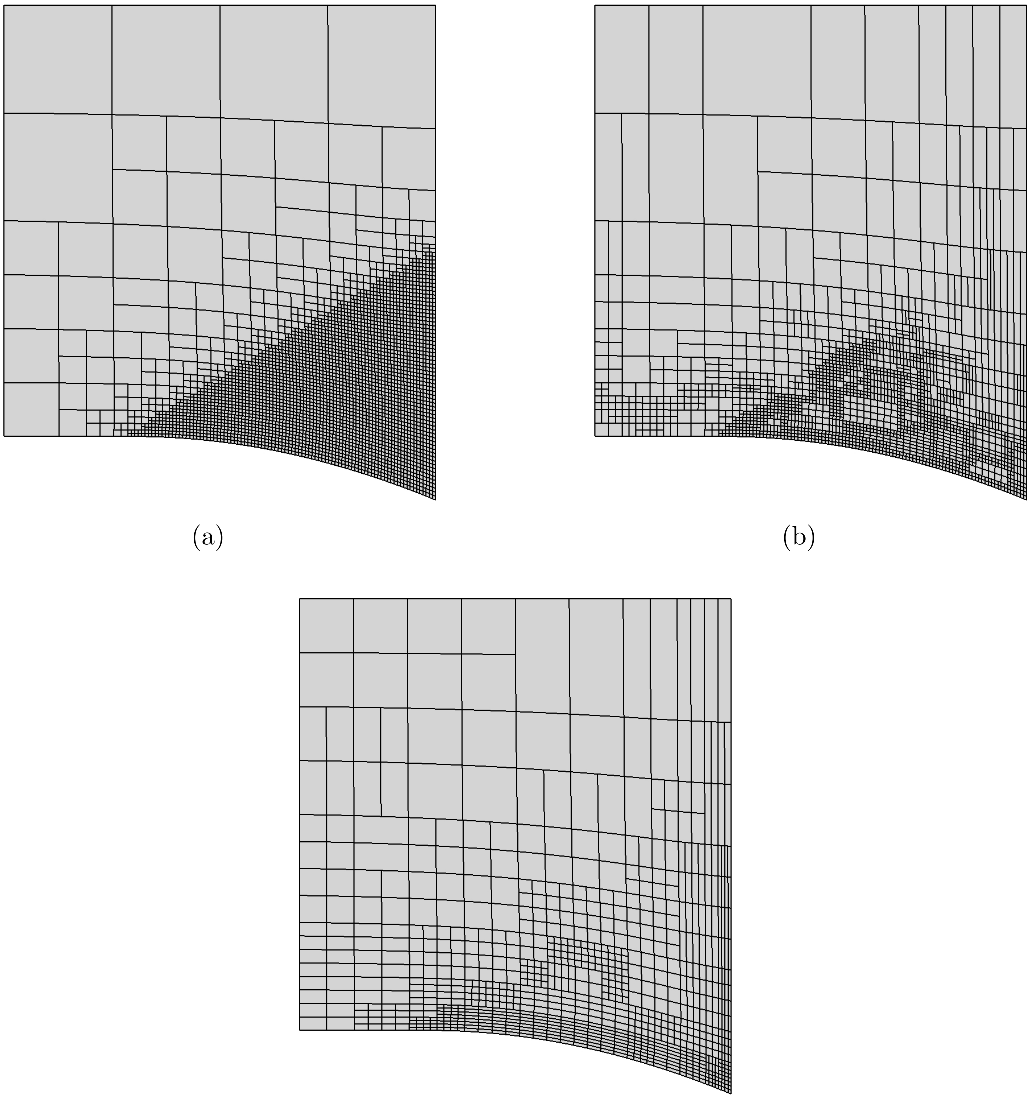
(a) (b)

(c) (d)

Figure 6.9: (a) The y-momentum and (b) B_y adjoint distributions for steady plasma flow through an expanding tube after 7 uniform refinements. Blocks: 16,384. Cells: 4,194,304. (c) The y-momentum and (d) B_y adjoint distributions for steady plasma flow through an expanding tube after 7 refinements using CC as refinement criterion. Blocks: 1,906. Cells: 487,936.

6.2 Superfast Plasma Flow Through an Expanding Tube

The second MHD case considered here is also taken from the previous field-aligned flow study of De Sterck *et al.* [149] in which a uniform superfast flow with a horizontal magnetic



(a)

(b)

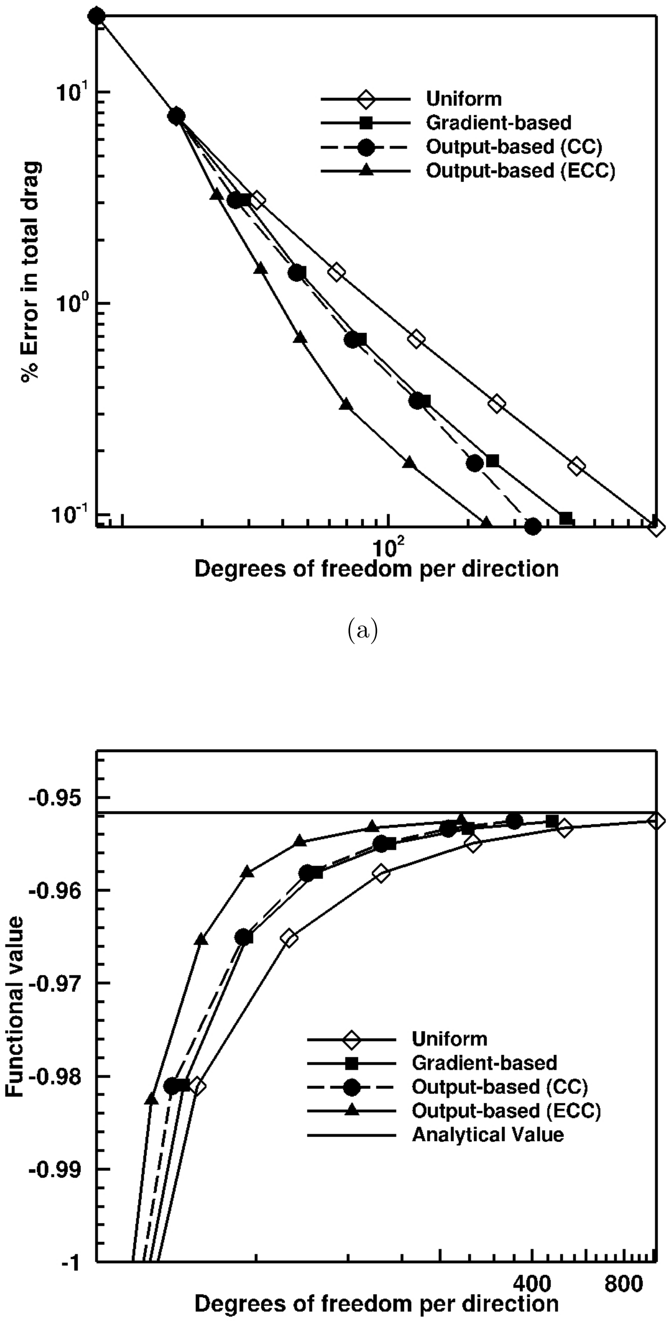
(c)

Figure 6.10: Adapted meshes showing the grid block boundaries (not individual cells) for steady plasma flow through an expanding tube: (a) After 7 refinements using density gradients as refinement criterion. Blocks: 3,420. Cells: 875,520. (b) After 7 refinements using CC as refinement criterion. Blocks: 1,906. Cells: 487,936. (c) After 7 refinements using ECC as refinement criterion. Blocks: 857. Cells: 219,392. Meshes shown in (a), (b) and (c) have approximately the same functional error.

field encounters an expanding tube. The length and height of the tube are 1 unit each, while the lower boundary of the tube has a partially sinusoidal profile. The computational domain is thus given by ($x \in [0, 1], y \in [y_0(x), 1]$), where $y_0(x) = 0$ for $x \in [0, 0.3]$ and $y_0(x) = -1+\cos(\pi/4(x-0.3))$ for $x \in [0.3, 1]$. The initial mesh, as shown in Figure 6.7(a), was composed of a single block and the number of cells per block is $8 \times 8 \times 4 = 256$, with 4 cells in the out-of-plane direction. At $y = y_0$ and $y = 1$, reflective boundary conditions were imposed. At the nozzle inlet, $x = 0$, the flow properties were held fixed with $\rho = 1$, $p = 1$, $v_x = 8.0$ and $B_x = 4.0$ for the supersonic and super-Alfvénic inflow. Thus, the sonic and Alfvén inflow Mach numbers are $M = 8\sqrt{3/5}$ and $M_A = 2.0$, respectively. At the nozzle exit, constant extrapolation is imposed for the exiting flow, which remains superfast. Figure 6.7(b) shows the predicted density contours and magnetic field lines for this case obtained on a fine uniformly refined mesh. As long as the lower wall remains straight, the flow stays uniform. When the tube wall starts to curve downward at $x = 0.3$, the plasma flow expands as the characteristic waves propagate downstream in a superfast flow. The flow is non-uniform only in a region downstream from the fast characteristics originating from the lower boundary at the start of the wall curvature ($x = 0.3$). This characteristic also coincides with the first density contour line emerging from the lower boundary at ($x = 0.3$), as can be seen in Figure 6.7(b). In this downstream rarefaction region, the density, pressure and magnetic strength continuously decrease. For reference purposes, the effectiveness of the proposed NKS algorithm for this case is illustrated by the convergence histories of the steady-state solution density residual given in Figure 6.8 for the complete set of refined meshes for all four refinement strategies considered.

As for the preceding superfast cylinder flow, the functional used for this case is again the total pressure drag comprising the thermal and magnetic pressure components on the curved lower boundary, in the direction of the incoming flow. Figure 6.9 shows the predicted y-momentum and B_y adjoint solutions for this functional corresponding to a converged solution obtained following 7 uniform refinements of the initial mesh as well as after 7 refinements using CC as the refinement criterion. Figure 6.10 shows a comparison between meshes obtained after several adaptive refinements on the initial mesh, using various gradient-based and output-based refinement criteria. Figure 6.10(a) shows the anisotropic mesh obtained after 7 gradient-based refinements on the initial mesh, based on the magnetic field and density gradients as refinement criteria. This approach refines the entire region downstream of the fast characteristic originating from the lower boundary at the start of the flow turning. In contrast, Figure 6.10(b) shows the anisotropic mesh obtained after 7 CC-based refinements of the initial mesh. As compared to the mesh of the gradient-based method, the resulting CC-based mesh is largely refined only in the lower half of the rarefaction region in regions near the tube wall expected to influence the drag functional. The ECC-based mesh obtained after 7 levels of refinement of the initial mesh exhibits shown in Figure 6.10(c) exhibits similar features.

As an exact analytical value of the drag functional is not available for the expanding



(a)

(b)

Figure 6.11: (a) Convergence of the percentage error in the predicted value of the functional and (b) convergence of the predicted value of the functional, as a function of the mesh size (number of degrees of freedom per direction) for various refinement techniques for the magnetized flow through an expanding tube.

tube plasma flow, a reference value of the total drag force on the curved boundary was determined here using a uniformly refined mesh, with a resolution finer than the highest mesh refinement level achieved for both the gradient-based and output-based adaptive mesh refinement techniques. The resulting computed value of the reference drag is -0.951652. The difference between the numerically computed value of the total drag force for a given mesh and this reference fine mesh solution was used as the measure of the error in the drag force. Figures 6.11(a) and 6.11(b) show plots of this percentage error and the predicted value of the total drag against the total number of degrees of freedom per direction, i.e., $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells, respectively. The horizontal solid line in the closeup of the convergence of the predicted drag force value for the various mesh refinement techniques of Figure 6.11(b) indicates the converged reference fine mesh value of the functional. For this case, all refinement methods considered here manage to continuously reduce the functional error by more than two orders of magnitude. Moreover, the standard gradient-based approach performs significantly better than the uniform approach. However, as compared to the gradient-based method, the output-based method employing CC as the refinement criterion achieves a mesh size reduction of 44% while the output-based method employing ECC achieves a reduction of 75%. These mesh savings have been recorded on the finest mesh refinement level, with an error level of ∼0.1%.

6.3 Superfast Rotating Cylindrical Outflow

In the next MHD flow problem previously studied by Sakurai [254], numerical results for the quasi-2D steady superfast expansion outflow of a magnetized plasma emanating from a rotating cylindrical object are examined. The computational domain for this case is an annular region between two concentric circles with inner radius $R_i = 1.0$ and outer radius $R_o = 6.0$. The initial mesh, as shown in Figure 6.12(a), was composed of 4 blocks and the number of cells per block is $8 \times 8 \times 4 = 256$, with 4 cells in the out-of-plane direction. At the inner boundary, $R_i = 1$, a uniform inflow with radial magnetic field is imposed with inflow density $\rho = 1$, inflow pressure $p = 1$, inflow radial velocity $u_r = 3$ and radial magnetic field given by $B_r = 1$. The incoming flow also has a rotational velocity component $u_\theta = 1$ in the counter-clockwise direction. Thus, the purely radial magnetic field imposed at the inner boundary is not aligned with the inflow velocity. The ratio of specific heats, γ, is taken to be 5/3. At the outer boundary, $R_o = 6$, the superfast plasma is allowed to flow freely outwards and a constant extrapolation of all boundary conditions is imposed accordingly.

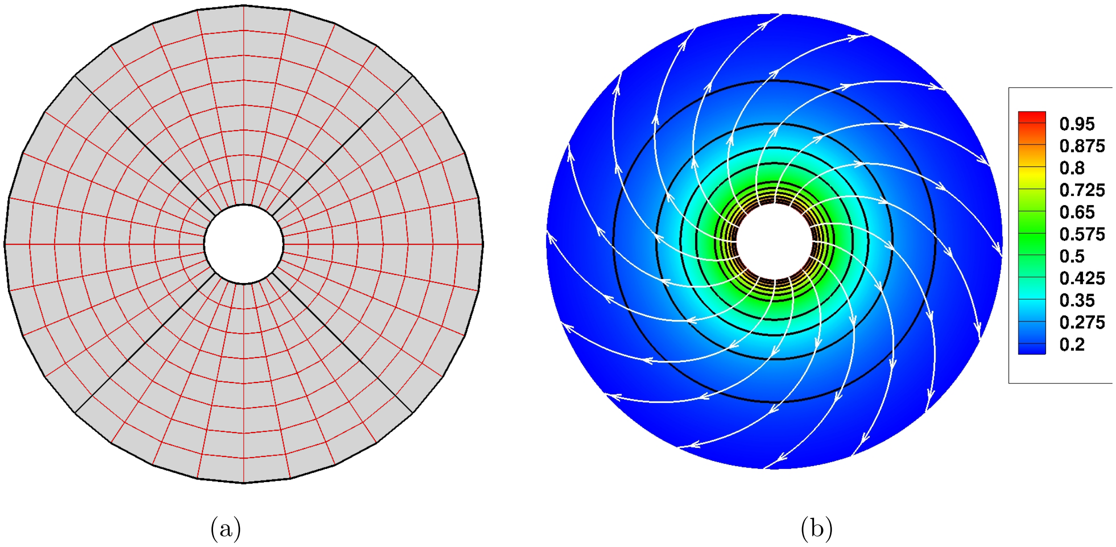


Figure 6.12: (a) Initial computational mesh for rotating cylindrical outflow. The thicker black lines are the block edges and the thinner red lines are the cell edges. The total number of blocks is 4, the total number of cells per block is 8×8×4=512 and the total number of cells is 2,048. (b) Predicted density contours and magnetic field lines for this case after 5 uniform refinements. Blocks: 4,096. Cells: 1,048,576.

The resulting predicted plasma flow field is shown in Figure 6.12(b), with the curved white rays representing the magnetic field lines and the colour contour field and solid black lines representing the computed plasma density. The flow is superfast over the entire computational domain. The magnetic field lines are clearly not aligned with the streamlines in the inertial non-rotating frame of reference. However, it can be shown that in the reference frame of the cylinder, rotating at an angular velocity $\Omega = -1$, the velocity streamlines and magnetic field lines are aligned everywhere. Such a field-aligned radial outflow can be thought of and used as a simplified model for stellar winds.

The preceding quasi-2D rotating MHD outflow can be completely specified analytically in terms of six constants, s, f_m, f_B, Ω, r_A and h, which are given by

$$s = \frac{p}{\rho^\gamma}, \tag{6.1}$$

$$f_m = \rho u_r r, \tag{6.2}$$

$$f_B = B_r r, \tag{6.3}$$

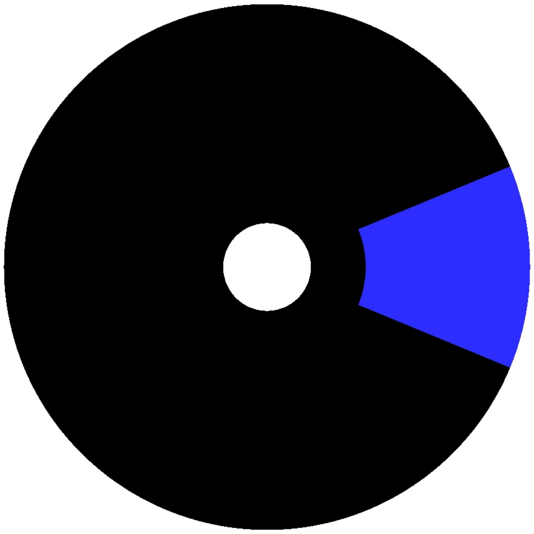


Figure 6.13: Region shaded blue indicating region over which the enthalpy is integrated for the functional for rotating cylindrical outflow.

$$(u_\theta - \Omega r)B_r - u_r B_\theta = 0, \tag{6.4}$$

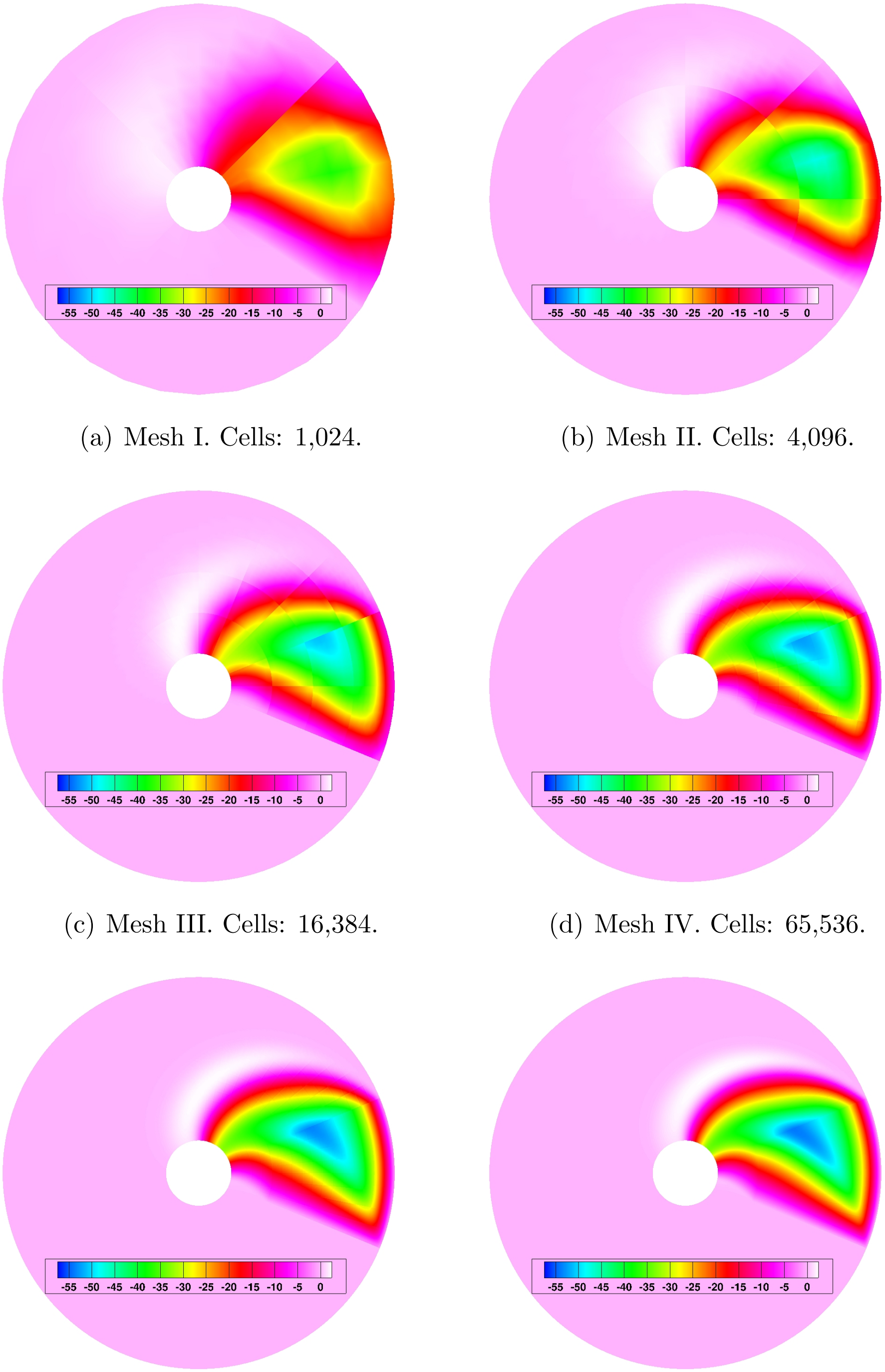
$$\Omega {r_A}^2 = r\left(u_\theta - \frac{B_r B_\theta}{\rho u_r}\right), \tag{6.5}$$

$$h = \frac{1}{2}{u_r}^2 + \frac{1}{2}(u_\theta - \Omega r)^2 + \frac{\gamma}{\gamma - 1}\frac{p}{\rho} - \frac{1}{2}\Omega^2 r^2, \tag{6.6}$$

where s is the entropy, f_m and f_B are the radial momentum flux and magnetic flux, respectively, Ω is the angular velocity of the rotating cylindrical object, r_A is the Alfvén radius and h is the Bernoulli constant. These six flow invariants completely specify the plasma outflow for the entire domain. As mentioned by Sakurai [254], the Bernoulli function, $H(r, \rho)$, can be derived and expressed as

$$H(r,\rho) = \frac{f_m^2}{2\rho^2 r^2} + \frac{1}{2}{r_A}^2\Omega^2\left(\frac{r_A/r - r/r_A}{1 - \rho f_B^2/f_m^2}\right)^2 + \frac{\gamma}{\gamma - 1}s\rho^{\gamma-1} - \frac{1}{2}\Omega^2 r^2 \tag{6.7}$$

Fixing the constants s, f_m, f_B, Ω and r_A, the profile $\rho(r)$ can be obtained as level curves of the Bernoulli function $H(r, \rho) = h$ for varying h, thus specifying the solution over the entire domain as a function of the radial position, r. For a more detailed analysis of this case, the reader is referred to the previous studies by Sakurai [254] and De Sterck *et al.* [149].



(a) Mesh I. Cells: 1,024.

(b) Mesh II. Cells: 4,096.

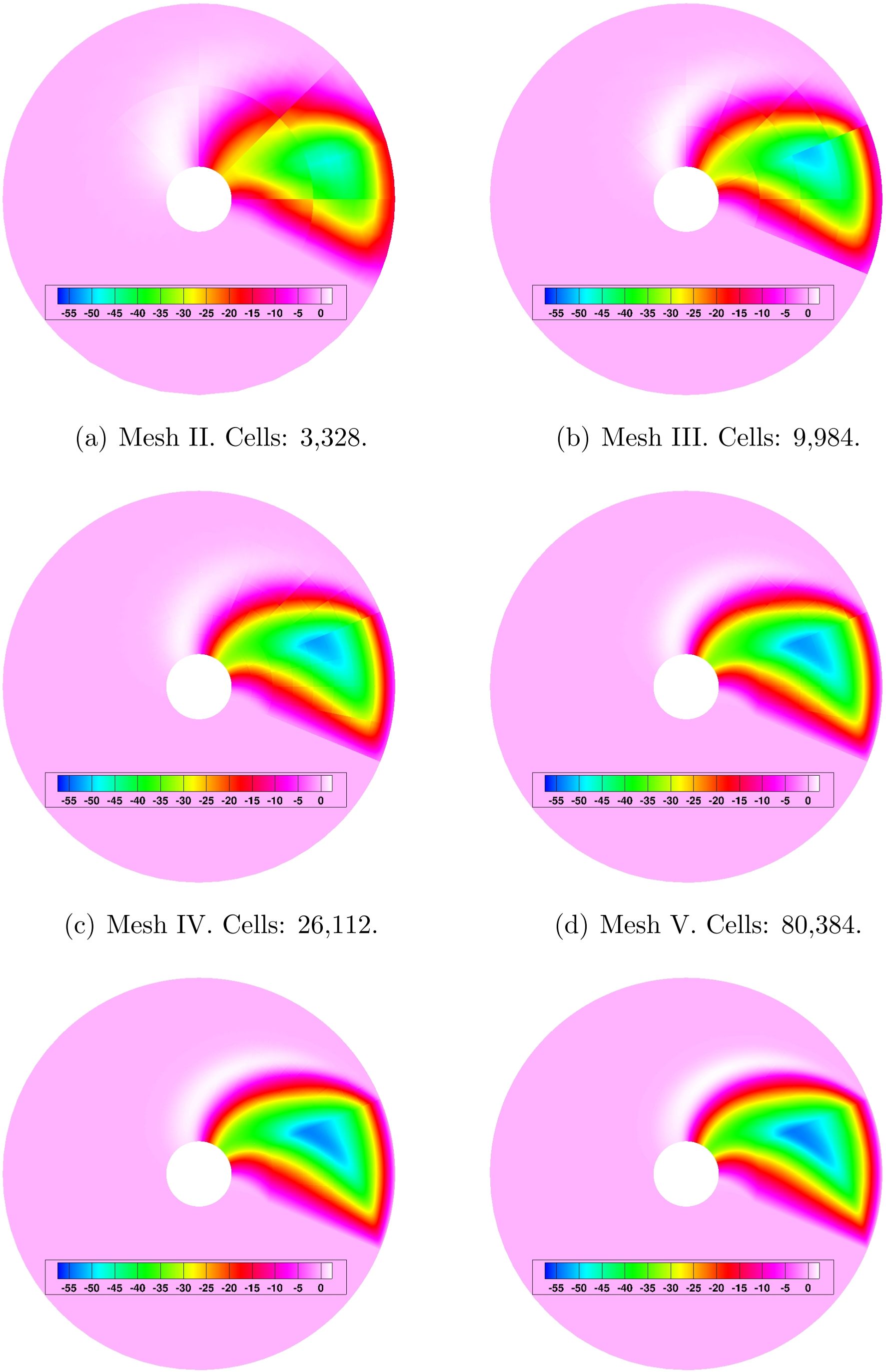
(c) Mesh III. Cells: 16,384.

(d) Mesh IV. Cells: 65,536.

(e) Mesh V. Cells: 262,144.

(f) Mesh VI. Cells: 1,048,576.

Figure 6.14: Density adjoint distribution for rotating cylindrical outflow on successive uniformly refined meshes.



(a) Mesh II. Cells: 3,328.

(b) Mesh III. Cells: 9,984.

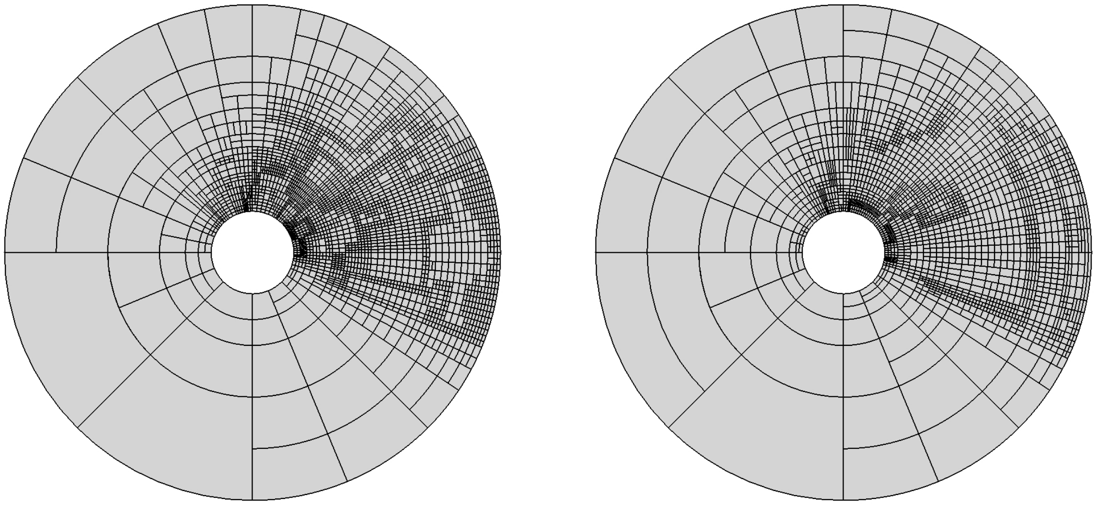
(c) Mesh IV. Cells: 26,112.

(d) Mesh V. Cells: 80,384.

(e) Mesh VI. Cells: 236,800.

(f) Mesh VII. Cells: 691,200.

Figure 6.15: Density adjoint distribution for rotating cylindrical outflow on successive adaptively refined meshes using CC as refinement criterion.


(a)

(b)

Figure 6.16: (a) Adapted meshes showing the grid block boundaries (not individual cells) for rotating cylindrical outflow: (a) After 6 refinements using CC as refinement criterion. Blocks: 2,700. Cells: 691,200. (b) After 6 refinements using ECC as refinement criterion. Blocks: 2,016. Cells: 516,096.

The functional used for the plasma outflow case considered here is the total enthalpy of the plasma integrated over the region ($r \in [2.236, 6.0], \theta \in [-\pi/8, \pi/8]$), for a cylindrical

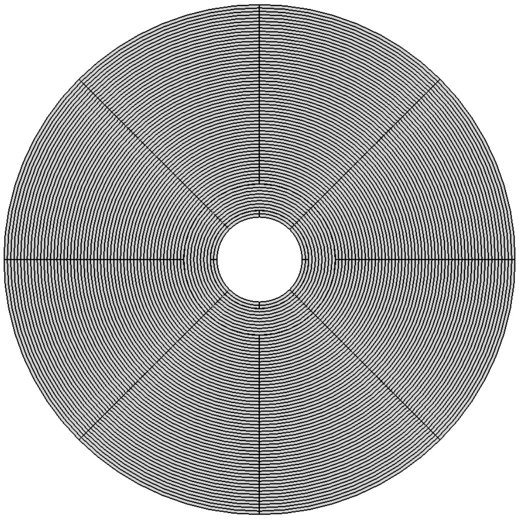
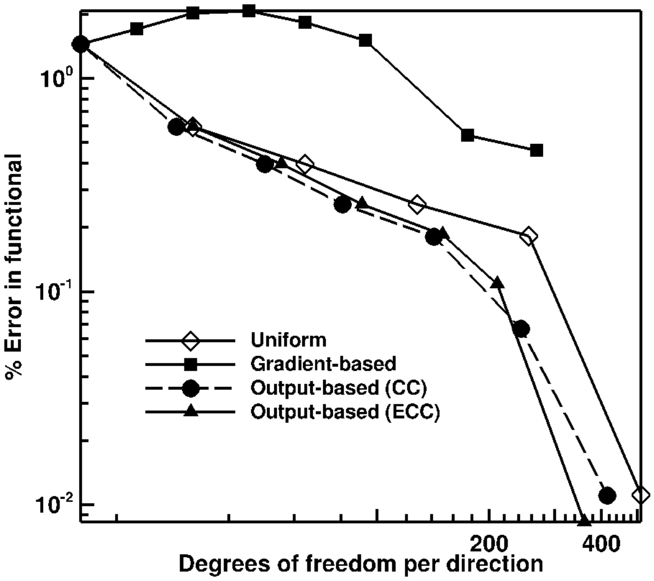
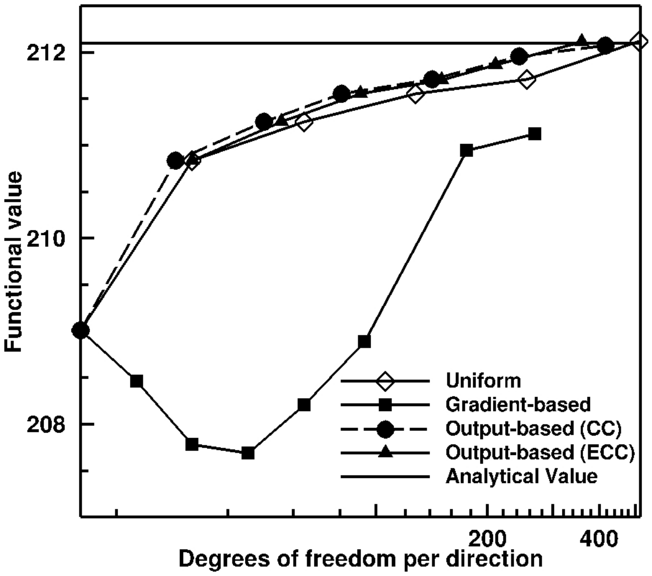


Figure 6.17: Adapted mesh showing the grid block boundaries (not individual cells) for rotating cylindrical outflow after 7 refinements using velocity, density, and magnetic field gradients as refinement criteria. Blocks: 1,124. Cells: 287,744.



(a)



(b)

Figure 6.18: (a) Convergence of the percentage error in the predicted value of the first functional and (b) convergence of the predicted value of the first functional, as a function of the mesh size (number of degrees of freedom per direction) for various refinement techniques for the rotating cylindrical outflow.

coordinate system with its origin at the centre of the annulus and the positive x-axis as the $\theta = 0^\circ$ direction. This region is indicated in Figure 6.13. Using the flow solution, as given in Equations (6.1)–(6.6), the analytical value of this proposed functional can be found to be 212.098. The resulting predicted solution for the density adjoint for this case is shown on successive uniformly refined meshes in Figure 6.14 and for successive adaptively refined meshes obtained using the CC-method in Figure 6.15. Figures 6.16(a) and 6.16(b) additionally show the corresponding meshes obtained after 6 adaptive refinements of the initial mesh using the two output-based refinement criteria for this functional. As would be desired for an accurate functional, both methods effectively concentrate the refinement of the mesh along the spiral path of the outwardly flowing plasma that passes through the region of integration of the functional, producing quite similar meshes. In contrast, Figure 6.17 shows the mesh obtained after 7 gradient-based refinements of the initial mesh for the plasma outflow problem. The velocity, density, and magnetic field gradients were all used as criteria for the simulation with the gradient-based technique. Most of the gradient-based refinement of the mesh occurs in an axisymmetric manner in the radial direction. While not necessarily effective in reducing the functional of interest, this behaviour is expected since all the solution variables exclusively depend on the radial position r, as given by Equations (6.1)–(6.6).

The true error in the functional, i.e., the difference between the numerical value of the functional and the analytical value, and the convergence of the predicted value of the functional are plotted as a percentage against the total number of degrees of freedom per direction, i.e., $\sqrt{N_{\text{cells}}/4}$, where N_{cells} is the total number of grid cells in the computational mesh, in Figures 6.18(a) and 6.18(b), respectively. From these convergence results it is quite clear that the gradient-based approach fails to significantly reduce the functional error, reducing it only by a factor of about one half after 7 refinements compared to the initial mesh. This can be attributed to the inability of the gradient method to detect errors in the azimuthal direction as well as those associated with the proposed functional and therefore only refining in the radial direction where the highest solution gradients exist. In contrast, the two output-based methods are able to reduce the error by more than one order of magnitude after several mesh refinements. Moreover, the output-based method employing the CC refinement criterion achieves a mesh size reduction of approximately 34% as compared to that of the uniform refinement approach, while the anisotropic AMR procedure based on the ECC output-base procedure achieves a reduction of approximately 51% compared to the uniformly refined mesh results. The corresponding error level is $\sim$0.01%.

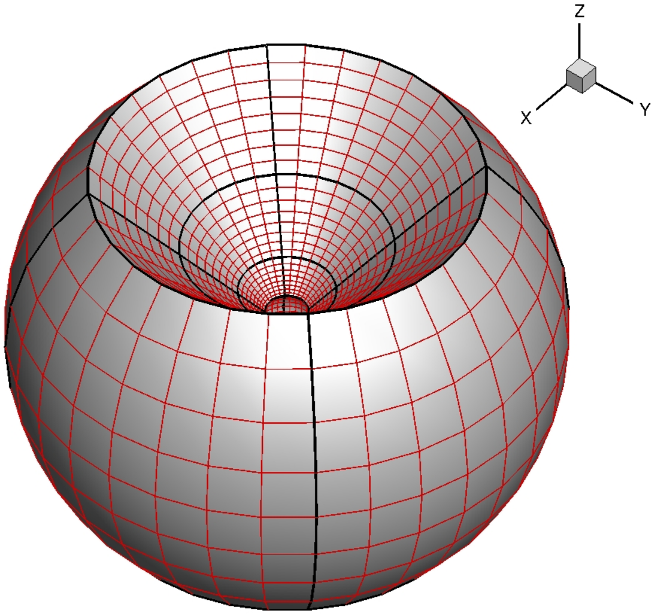
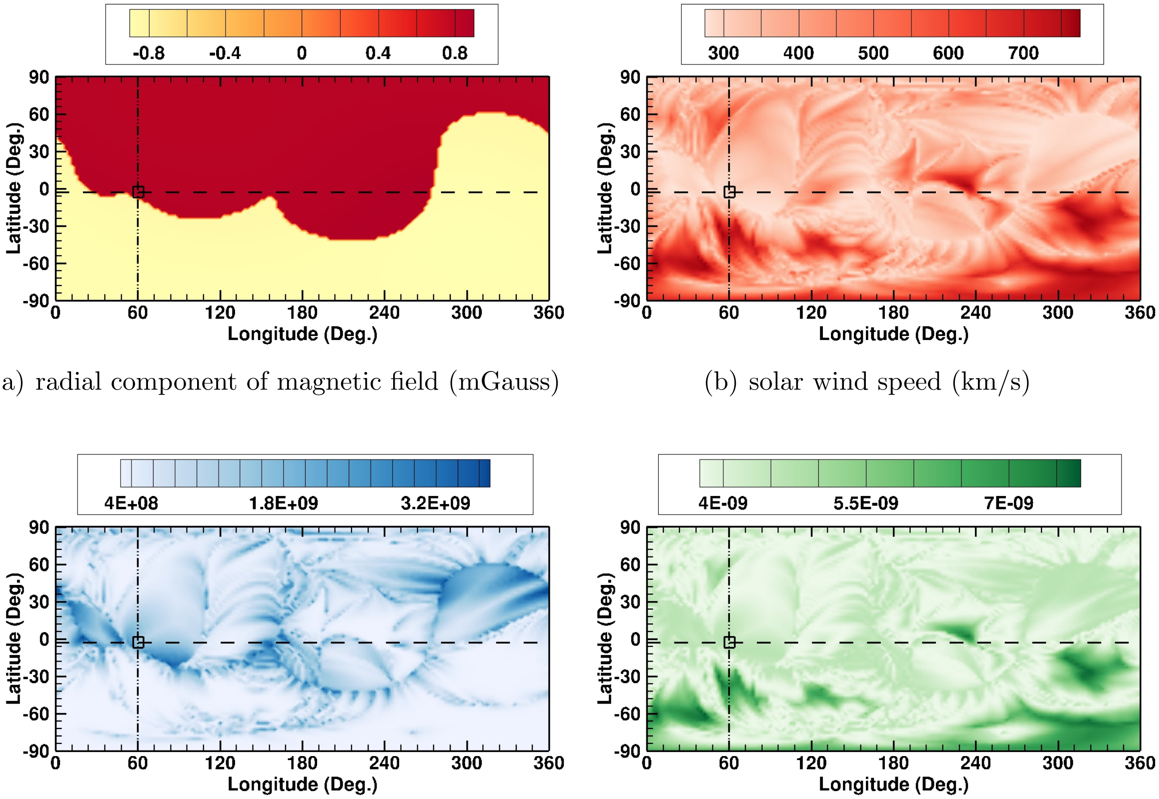


Figure 6.19: Initial mesh used for the steady state background solar wind simulation. The thicker black lines are the block edges and the thinner red lines are the cell edges. Each block contains $8 \times 8 \times 8 = 512$ cells. The total number of blocks is 12. The total number of cells is $12 \times 512 = 6,144$.

6.4 Steady Background Solar Wind Flow

As the final ideal MHD test case, the steady state background solar wind plasma flow described previously in Chapter 5 is reconsidered, with an inner boundary at $25R_{Sun}$ and the outer boundary at a radius of $225R_{Sun}$. The computational domain is a spherical shell with inner and outer radii as $25R_{Sun}$ and $225R_{Sun}$, covering a longitudinal span of 360° and a latitudinal span of 90° with 0° latitude being the equatorial plane. The initial spherical mesh, as shown in Figure 6.19, is composed of 18 blocks and the number of cells per block is $8 \times 8 \times 8 = 512$. The magnetogram photospheric maps of the Sun dated 25^{th} June 2015 at 1 am were used to define the inflow conditions at $25R_{Sun}$.

Two-dimensional plots of the radial component of the magnetic field, solar wind speed, plasma density and thermodynamic pressure at the inner boundary, $r = 25R_{Sun}$, are shown in Figure 6.20. At the inner boundary, the flow variables are kept fixed at the superfast inflow conditions given in Figure 6.20. At the outer boundary, constant extrapolation is imposed for the outgoing superfast flow. The ratio of specific heats, γ, is assumed to be 5/3. The computations are performed in the non-inertial corotating



(a) radial component of magnetic field (mGauss)

(b) solar wind speed (km/s)

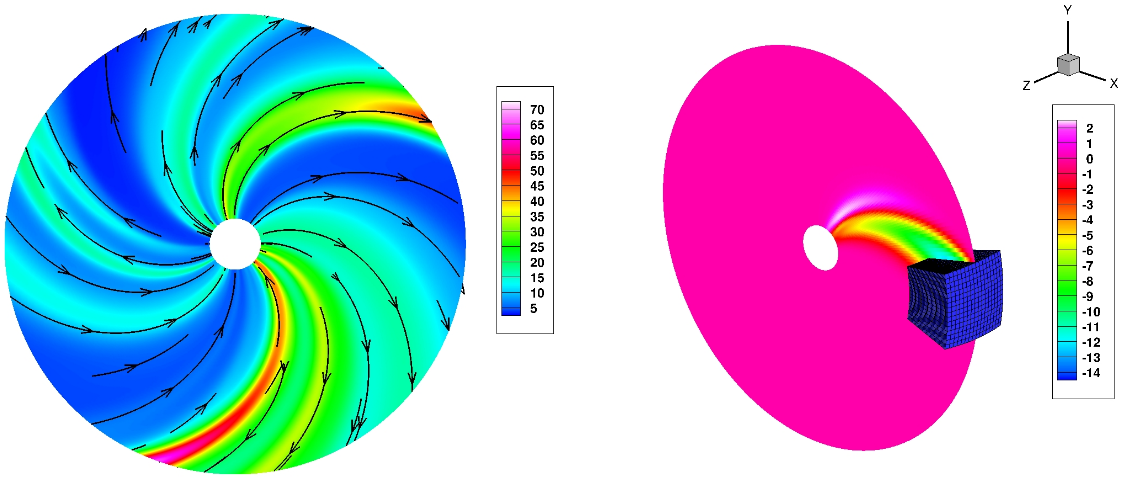
(c) density (count/m^3)

(d) pressure (Pa)

Figure 6.20: Boundary conditions at $r = 25R_{Sun}$ for the ambient solar wind simulation using the 25th June 2015 magnetogram data set.

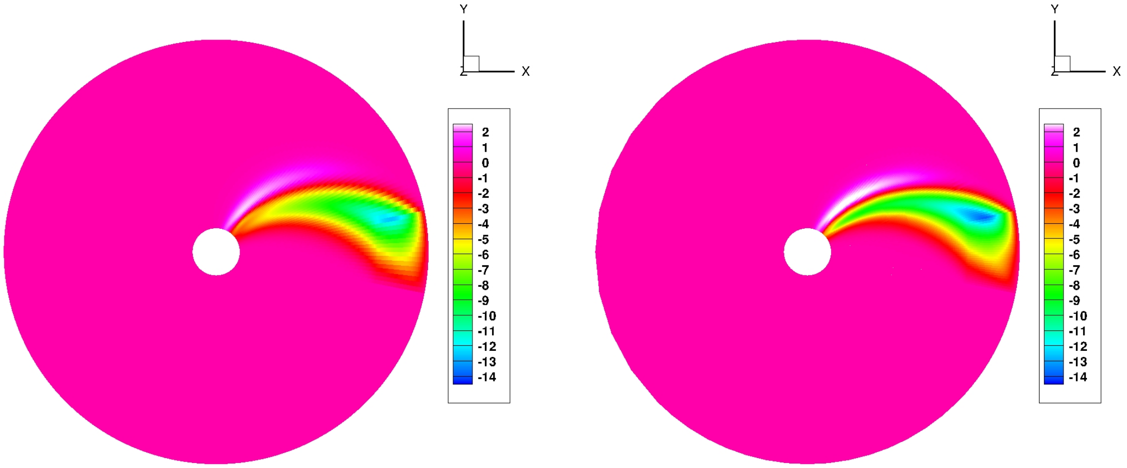
frame of the Sun. As discussed in Chapter 5, the main features are the formation of the current sheet across which the magnetic field is discontinuous, the Parker spiral and the fast and slow components of the solar wind. The plasma density, ρ, and magnetic field strength, $|\mathbf{B}|$, decrease with increasing radial distance from the Sun, being inversely proportional to the square of the radial distance, r^2. The flow is superfast and therefore supersonic throughout the computational domain. The quantities, $(|\mathbf{B}|r^2)$, (ρr^2), and the magnitude of the solar wind velocity stay constant along the spiral path of the plasma. Figure 6.21(a) shows contours of $(\rho(r/214.9R_{Sun})^2)$ on the ecliptic plane of the Sun with streamlines depicting the magnetic field lines. The magnetic field lines change sign across the region where the current sheet intersects the ecliptic plane.

The functional used here for the present solar-wind simulation was the enthalpy integrated over the region $(r \in [170.351167R_{Sun}, 225.0R_{Sun}], \theta \in [-\pi/16, \pi/16], \phi \in [-\pi/16, \pi/16])$ for a spherical coordinate system with its origin at the Sun's centre, the



(a) (b)

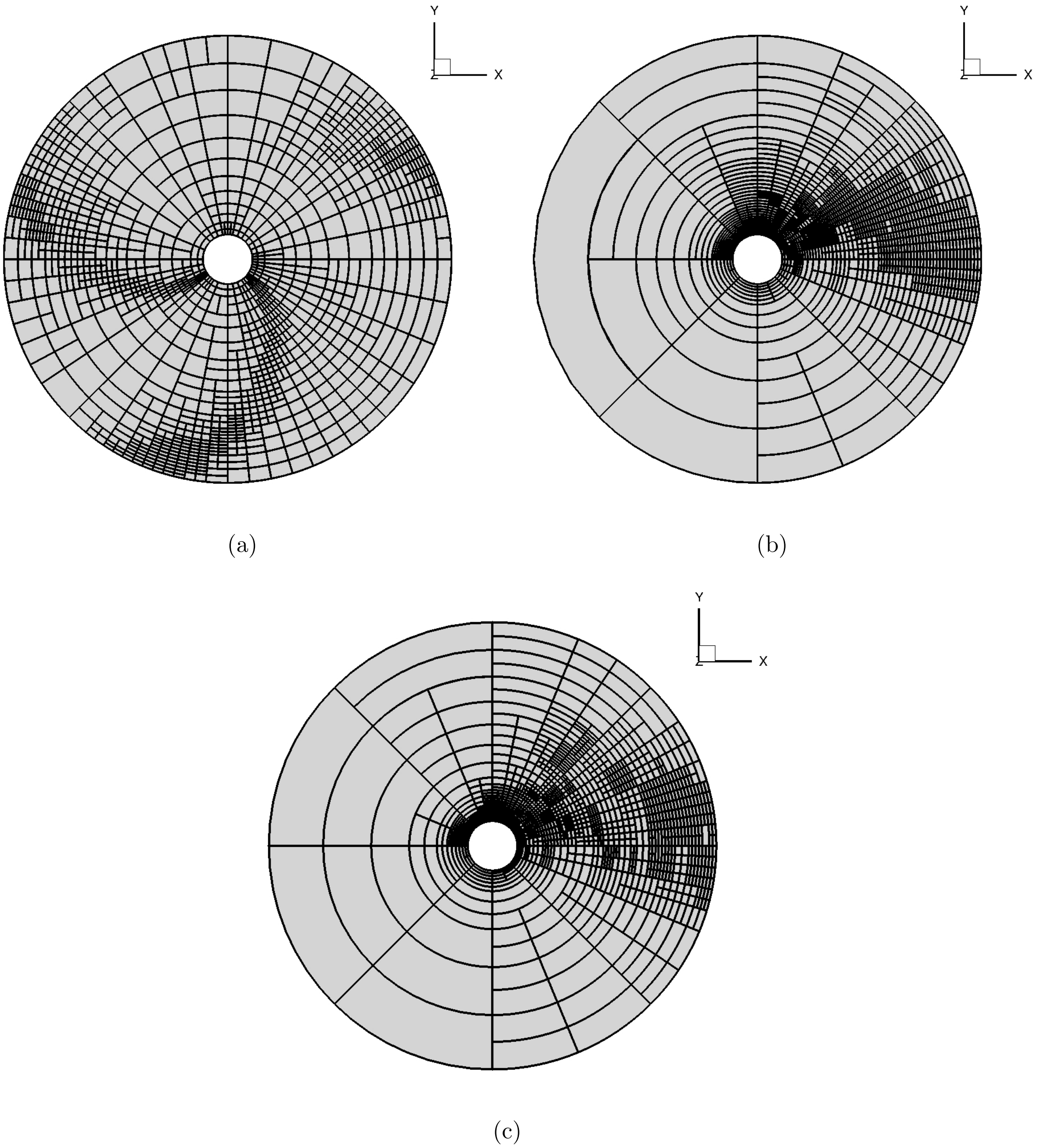
Figure 6.21: (a) Predicted contours of $(\rho(r/214.9R_{Sun})^2)$ (count/m^3) and magnetic field lines for the steady background solar wind on the ecliptic xy-plane after 3 uniform refinements. Blocks: 6,144. Cells: 3,145,728 (b) Region shaded blue indicating region over which the enthalpy is integrated.



(a) (b)

Figure 6.22: (a) Density adjoint distribution for the steady background solar wind on the ecliptic xy-plane : (a) after 3 uniform refinements. Blocks: 6,144. Cells: 3,145,728. (b) after 4 refinements using CC as refinement criterion. Blocks: 4,445. Cells: 2,275,840.

positive x-axis as the $\theta = 0°$ direction and the ecliptic plane represented by $\phi = 0°$. Such a functional places an emphasis on the solution in the vicinity of Earth, which would be



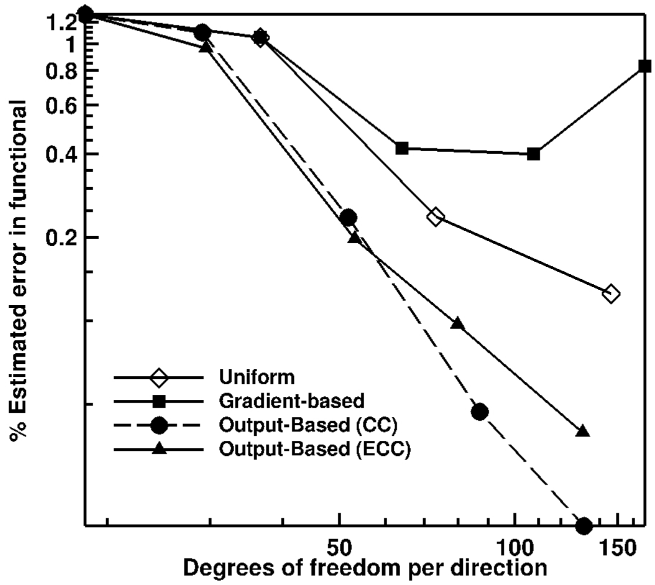
(a)

(b)

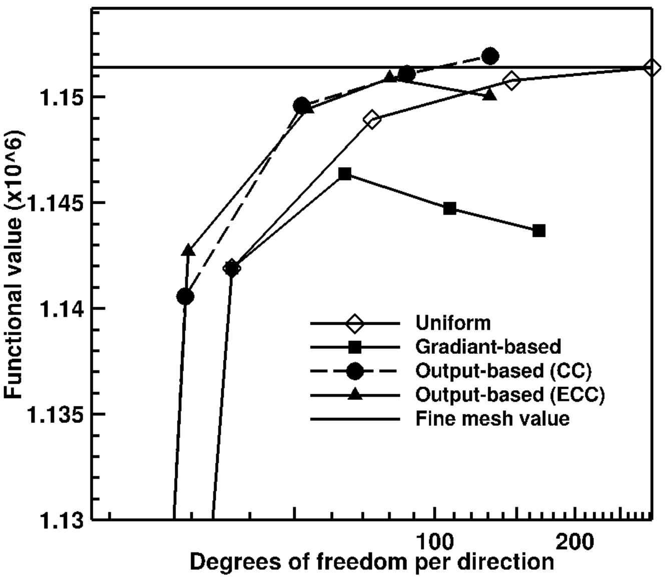
(c)

Figure 6.23: Sliced ecliptic xy-plane cross-sections of adapted meshes showing the grid block boundaries (not individual cells) for the ambient solar wind simulation using the 25^{th} June 2015 magnetogram data set: (a) After 5 refinements using ($|\mathbf{B}|r^2$) gradient as refinement criterion. Blocks: 27,466. Cells: 14,062,592. (b) After 5 refinements using CC as refinement criterion. Blocks: 22,254. Cells: 11,394,048. (c) After 5 refinements using ECC as refinement criterion. Blocks: 19,881. Cells: 10,179,072.

natural for many space weather simulations. Figure 6.21(b) depicts the proposed region



(a)



(b)

Figure 6.24: (a) Convergence of the estimated percentage error in the predicted value of the functional as a function of the mesh size (number of degrees of freedom per direction) for ambient solar wind simulation using the 25^{th} June 2015 magnetogram data set. (b) Convergence of the value of the functional for various refinement techniques for this case.

of interest for the functional evaluation. Figure 6.22 shows the corresponding predicted distributions of the 1^{st} component of the adjoint solution vector on the ecliptic xy-plane after 3 uniform refinements of the initial mesh as well as after 4 refinements using the CC refinement criterion. Additionally, Figure 6.23 shows a comparison of the resulting meshes obtained after application of adaptive refinement strategies to the initial mesh, for both the gradient-based and output-based refinement methods. Figure 6.23(a) shows a slice of the mesh obtained after 5 adaptive refinements using the gradient of $(|\mathbf{B}|r^2)$ as the refinement criterion. The latter focuses on refining the region of the current sheet across which the $(|\mathbf{B}|r^2)$ gradients are high, as expected from the results described in Section 5.3.1 of the previous chapter. Figures 6.23(b) and 6.23(c) show slices of the computational mesh obtained after 5 output-based adaptive refinements obtained using the CC and ECC output-based approaches, respectively. It is very evident that the output-based methods focus the mesh refinement along the spiral path of the plasma that eventually reaches and flows through the volume over which the functional of interest is defined.

An analytical value of the functional is not available for this last MHD case. Additionally, performing calculations on a uniformly refined mesh finer than the highest mesh refinement level attained for any of the adaptive mesh refinement techniques is deemed infeasible for this simulation due to limitations of the available computational resources. Hence, estimated values of the functional error were calculated using the adjoint-based error estimation procedure for all the meshes considered herein. Figure 6.24(a) shows a plot of the estimated percentage error against the total number of degrees of freedom per direction for this fully three-dimensional flow, i.e., $\sqrt[3]{N_{\text{cells}}}$, where N_{cells} is the total number of grid cells in the computational mesh, for the various refinement techniques considered for this case. Additionally, Figure 6.24(b) shows a close-up of the convergence of functional values for the various mesh refinement techniques. Here, the solid horizontal line indicates the value of the functional obtained on the finest uniform mesh considered in the study (i.e., the mesh obtained following 5 uniform refinements of the initial mesh). For the gradient-based technique using $(|\mathbf{B}|r^2)$ as the refinement criterion, the estimated error is not greatly reduced as the computational mesh is refined and rather inaccurate values of the functional are obtained even after several refinements. This can be explained by the fact that the current sheet does not necessarily lie in the region that influences the functional. The output-based refinement techniques are however shown to provide significant improvements in the prediction of the functional compared to both uniform and standard gradient-based refinement approaches. Depending on the desired level of error in the funcitonal, it is estimated that mesh savings of at least 65% are achievable for the current solar-wind problem using output-based methods as compared to the uniform refinement method. The preceding findings provide strong evidence for the potential benefits of the proposed output-based AMR strategy for predicting space weather phenomena.

Chapter 7

Conclusions

The key objective of this thesis was the development of a numerical technique to achieve significant reductions in the computational mesh size required for accurate predictions of integrated engineering quantities of interest for steady 2D and 3D ideal MHD plasma flows. In order to achieve this, a parallel output-based mesh refinement algorithm was developed and applied in conjunction with a block-based anisotropic AMR method, first to the solution of steady, non-conducting aerodynamic flows governed by the Euler and Navier-Stokes equations and subsequently extended to electrically conducting flows governed by the equations of ideal MHD. In the proposed approach, the governing system of partial-differential equations was discretized using a finite-volume approach, representing the solution on body-fitted hexahedral meshes distributed parallelly across a multiple number of processors. An inexact variant of Newton's iterative method was used to solve this discretized system of governing equations in combination with a limited piecewise linear reconstruction procedure to obtain second-order accurate solutions. The anisotropic block-based AMR technique allowed the economical generation of meshes by increasing the mesh resolution only in directions as dictated by the solution content and refinement criteria. A light, flexible binary tree data structure tracked connectivities between blocks spread across multiple processors, enabling efficient, parallel AMR via domain decomposition. The discrete adjoint formulation was used to calculate local sensitivities of functionals to the flow variables. Primal flow quantity residual errors were evaluated on a finer mesh using an h-refinement strategy and the adjoint solution was used to weigh these residual errors to obtain error estimates of the functionals. These error estimates were used to drive the mesh adaptation, resulting in anisotropically adapted meshes customized for accurate functional evaluation.

The various flow cases studied using the proposed output-based anisotropic AMR framework demonstrated substantial mesh size reductions as compared to more traditional physics-based mesh adaptation strategies while efficiently reducing the target functional error. Additionally, a novel space weather simulation framework was also developed that coupled a data-driven model of the inner solar wind with the anisotropic block-based AMR finite-volume framework. The framework, where the solar wind model was used as a driver for the heliosphere MHD simulation, showed remarkable success in accurately capturing features of the solar wind and also predicting real-time space weather and solar-wind plasma properties at the Earth. The latter also demonstrated the capability of the parallel AMR framework to deal with more realistic flow problems rather than just canonical, standardized test cases. In the sections to follow, the original contributions made in this thesis and recommendations for future extensions of this work are discussed.

7.1 Original Contributions

The original contributions of this thesis can be summarized as follows:

- An *a posteriori*, output-based, error estimation algorithm was combined with a parallel block-based anisotropic AMR framework for the numerical solution of steady 3D flows. The thesis represents the first application of this particular combination of refinement strategies. The performance of the proposed output-based AMR strategy was demonstrated for various types of flows including inviscid aerodynamic flows governed by the Euler equations, viscous aerodynamic flows governed by the Navier-Stokes equations and electrically conducting gaseous flows governed by the ideal MHD equations.

- The thesis also represents a first use of functional error estimates obtained using adjoint-based error estimation to drive the mesh refinement framework to 3D MHD flows. The benefits obtained by the proposed algorithm in terms of mesh savings and efficient functional error reduction were demonstrated for a range of 3D ideal MHD flows.

- A data-driven 3D integrated solar wind-MHD model that employs photospheric magnetic field maps obtained from solar magnetograms and uses them as input to drive the MHD simulation was developed. The capabilities of this framework to accurately simulate the background solar wind structure as well as potentially forecast solar wind properties at the Earth were demonstrated. This framework includes capabilities that are not available together in any other available model:

(1) the new model uses a cubed-sphere grid covering a longitudinal range of 360° in ϕ and a latitudinal range of 180° in θ, i.e., all directions in 3D space; (2) it has parallel block-based anisotropic AMR capabilities; (3) it can perform calculations in the inertial or the Sun's corotating frame of reference, (4) it can perform simulations of both the steady-state ambient or background solar wind as well as the fully unsteady solar wind that could be used for space weather forecasting and possibly the study of disturbances such as CMEs in a background solar wind.

7.2 Recommendations for Future Research

The parallel output-based anisotropic AMR framework discussed in this thesis provides a means for accurate prediction of both aerodynamic and MHD flows. However, it is also a solid foundation for building additional capabilities in areas which were considered beyond the scope of this thesis. Brief descriptions for various possibilities for future research are given below.

- The adjoint problem was formulated only for the Euler, Navier-Stokes and ideal MHD equations. The parallel *a posteriori* output-based error estimation algorithm presented here is rather generalizable and extending it to other systems of governing equations should be a reasonably easy task. Adjoint-based error estimation has already been used to study reactive flow by various researchers in the past [110–118]. However, the use of a highly scalable parallel block-based AMR framework in combination with adjoint-based error estimation to study steady solutions to reactive flow problems has not been explored. Extensions to more sophisticated fluid systems such as the Reynolds-Averaged Navier-Stokes (RANS) equations would enable accurate calculation of aerodynamic forces without neglecting some of the essential physics.

- The output-based AMR framework for MHD could also be used for more practical applications involving complex geometries, such as the accurate calculation of forces and moments on outer space vehicles, especially propulsion technologies such as the magnetic sail [255] and the electric sail [256] that harness the momentum of the solar wind. The framework could also be extended to solve the equations of resistive MHD in order to be able to simulate a wider range of space physics flows.

- The output-based error estimation algorithm in this thesis was limited to the study of steady flows. However, many flows of practical interest involve unsteady phenomena. Although time-accurate output-based mesh adaptation for unsteady flows have been the subject of research in the past by various researchers [105, 119, 120], effective and practical strategies for time-dependent problems are still not readily available. Such methods could also be used to study unsteady space weather phenomena such as the propagation of coronal mass ejections (CMEs) [225, 226].

- In the error estimation framework developed here, the residual errors are calculated by prolonging the solution onto a finer mesh. These errors can also be calculated by using a higher-order residual reconstruction on the same mesh, i.e., a *p*-derived residual error [90, 92, 146]. Circumventing the creation of a new mesh every time error indicators need to be calculated could potentially provide savings in time required for the output-based AMR procedure.

- In the adjoint calculation algorithm used here, linearized approximations of the residual Jacobians were used to construct the linear system for obtaining the adjoint solution. A higher-order Jacobian calculation would require taking into account the influence of neighbouring cells on the residual through the second-order limited piecewise linear reconstruction used here. Differentiating the residual with respect to the solution in these neighbouring cells is a tedious task. Libraries for automatic differentiation [199, 200] could also be used for this purpose. This would result in more accurate calculations of the adjoint variables and as a result, more accurate values of functional error estimates.

- In the current block-based AMR framework, mesh resolution is increased locally by refining individual blocks to form new blocks. Such a procedure is referred to as *h*-refinement wherein new, smaller cells are created to increase the mesh resolution. Added benefits could be achieved by increasing the order of polynomial reconstruction of the solution in the cell, i.e., by applying *p*-refinement. Such combined *hp*-refinement techniques [83, 90, 99, 108, 109] would also be worth exploring in the future.

- The proposed data-driven solar wind-MHD framework has been applied to the prediction of the background solar wind and its evolution in response to changes in the global solar magnetic field. It has the potential to be extended to include unsteady phenomena and disturbances such as coronal mass ejections (CMEs) [225, 226].

Bibliography

[1] L. Freret and C. P. T. Groth. Anisotropic non-uniform block-based adaptive mesh refinement for three-dimensional inviscid and viscous flows. Paper 2015-2613, AIAA, June 2015.

[2] T. I. Gombosi. *Physics of the space environment*. Cambridge University Press, 1998.

[3] M. Moldwin. *An introduction to space weather*. Cambridge University Press, 2008.

[4] P. A. Davidson. *An Introduction to Magnetohydrodynamics*. Cambridge University Press, 2001.

[5] J. P. H. Goedbloed and S. Poedts. *Principles of Magnetohydrodynamics: With Applications to Laboratory and Astrophysical Plasmas*. Cambridge University Press, 2004.

[6] V. J. Pizzo. A three-dimensional model of corotating streams in the solar wind: 3. magnetohydrodynamic streams. *Journal of Geophysical Research: Space Physics*, 87(A6):4374–4394, 1982.

[7] A. Usmanov. A global numerical 3-D MHD model of the solar wind. *Solar Physics*, 146(2):377–396, 1993.

[8] R. Lionello, Z. Mikić, and D. D. Schnack. Magnetohydrodynamics of solar coronal plasmas in cylindrical geometry. *Journal of Computational Physics*, 140:172–201, 1998.

[9] J. A. Linker, Z. Mikić, D. A. Biesecker, R. J. Forsyth, S. E. Gibson, A. J. Lazarus, A. Lecinski, P. Riley, A. Szabo, and B. J. Thompson. Magnetohydrodynamic modeling of the solar corona during Whole Sun Month. *Journal of Geophysical Research: Space Physics*, 104(A5):9809–9830, 1999.

[10] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. De Zeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *Journal of Computational Physics*, 154:284–309, 1999.

[11] C. P. T. Groth, D. L. De Zeeuw, T. I. Gombosi, and K. G. Powell. Global three-dimensional MHD simulation of a space weather event: CME formation, interplanetary propagation, and interaction with the magnetosphere. *Journal of Geophysical Research*, 105(A11):25,053–25,078, 2000.

[12] I. I. Roussev, T. I. Gombosi, I. V. Sokolov, M. Velli, W. Manchester IV, D. L. DeZeeuw, P. Liewer, G. Tóth, and J. Luhmann. A three-dimensional model of the solar wind incorporating solar magnetogram observations. *The Astrophysical Journal Letters*, 595(1):L57–L61, 2003.

[13] W. B. Manchester, T. I. Gombosi, I. Roussev, D. L. De Zeeuw, I. V. Sokolov, Kenneth G. Powell, G. Tóth, and M. Opher. Three-dimensional MHD simulation of a flux rope driven CME. *Journal of Geophysical Research: Space Physics*, 109(A1):A01102–1–A01102–17, 2004.

[14] W. B. Manchester IV, T. I. Gombosi, I. Roussev, A. Ridley, D. L. De Zeeuw, I. V. Sokolov, K. G. Powell, and G. Tóth. Modeling a space weather event from the sun to the Earth: CME generation and interplanetary propagation. *Journal of Geophysical Research: Space Physics*, 109(A2):A02107–1–A02107–15, 2004.

[15] G. Tóth, D. L. De Zeeuw, T. I. Gombosi, W. B. Manchester, A. J. Ridley, I. V. Sokolov, and I. I. Roussev. Sun-to-thermosphere simulation of the 28–30 October 2003 storm with the space weather modeling framework. *Space Weather*, 5(6):S06003–1—–S06003–16, 2007.

[16] W. B. Manchester IV, A. Vourlidas, G. Tóth, N. Lugaz, I. I. Roussev, I. V. Sokolov, T. I. Gombosi, D. L. De Zeeuw, and M. Opher. Three-dimensional MHD simulation of the 2003 October 28 coronal mass ejection: comparison with LASCO coronagraph observations. *The Astrophysical Journal*, 684(2):1448–1460, 2008.

[17] G. Tóth, B. van der Holst, I. V. Sokolov, D. L. De Zeeuw, T. I. Gombosi, F. Fang, W. B. Manchester, X. Meng, D. Najib, K. G. Powell, Q. F. Stout, A. Glocer, Y.-J. Ma, and M. Opher. Adaptive numerical algorithms in space weather modeling. *Journal of Computational Physics*, 231(3):870–903, 2012.

[18] B. van der Holst, I. V. Sokolov, X. Meng, M. Jin, W. B. Manchester IV, G. Tóth, and T. I. Gombosi. Alfvén wave solar model (AWSoM): coronal heating. *The Astrophysical Journal*, 782(2):81–95, 2014.

[19] P. Riley, J. A. Linker, and Z. Mikić. An empirically-driven global MHD model of the solar corona and inner heliosphere. *Journal of Geophysical Research: Space Physics*, 106(A8):15889–15901, 2001.

[20] D. Odstrčil, J. A. Linker, R. Lionello, Z. Mikić, P. Riley, V. J. Pizzo, and J. G. Luhmann. Merging of coronal and heliospheric numerical two-dimensional MHD models. *Journal of Geophysical Research: Space Physics*, 107(A12):1493–1–1493–11, 2002.

[21] R. Lionello, J. A. Linker, and Z. Mikić. Multispectral emission of the sun during the first Whole Sun Month: Magnetohydrodynamic simulations. *The Astrophysical Journal*, 690(1):902–912, 2009.

[22] X. Feng, L. Yang, C. Xiang, S. T. Wu, Y. Zhou, and D. Zhong. Three-dimensional solar *wind* modeling from the sun to Earth by a SIP-CESE MHD model with a six-component grid. *The Astrophysical Journal*, 723(1):300–319, 2010.

[23] X. Feng, L. Yang, C. Xiang, C. Jiang, X. Ma, S. T. Wu, D. Zhong, and Y. Zhou. Validation of the 3D AMR SIP-CESE solar wind model for four Carrington rotations. *Solar Physics*, 279(1):207–229, 2012.

[24] V. G. Merkin, R. Lionello, J. G. Lyon, J. Linker, T. Török, and C. Downs. Coupling of coronal and heliospheric magnetohydrodynamic models: solution comparisons and verification. *The Astrophysical Journal*, 831(1):23–36, 2016.

[25] R. Keppens, Z. Meliani, A. J. van Marle, P. Delmont, A. Vlasis, and B. van der Holst. Parallel, grid-adaptive approaches for relativistic hydro and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):718—744, 2012.

[26] O. Porth, C. Xia, T. Hendrix, S. P. Moschou, and R. Keppens. MPI-AMRVAC for solar and astrophysics. *The Astrophysical Journal Supplement Series*, 214(1):4–29, 2014.

[27] O. Porth, H. Olivares, Y. Mizuno, Z. Younsi, L. Rezzolla, M. Moscibrodzka, H. Falcke, and M. Kramer. The black hole accretion code. *Computational Astrophysics and Cosmology*, 4(1):1–42, 2017.

[28] C. Xia, P. F. Chen, and R. Keppens. Simulations of prominence formation in the magnetized solar corona by chromospheric heating. *The Astrophysical Journal Letters*, 748(2):L26–1–L26–16, 2012.

[29] C. Xia, R. Keppens, and Y. Guo. Three-dimensional prominence-hosting magnetic configurations: Creating a helical magnetic flux rope. *The Astrophysical Journal*, 780(2):130–154, 2013.

[30] C. Xia, R. Keppens, P. Antolin, and O. Porth. Simulating the in situ condensation process of solar prominences. *The Astrophysical Journal Letters*, 792(2):L38–1–L38–15, 2014.

[31] C. Xia and R. Keppens. Formation and plasma circulation of solar prominences. *The Astrophysical Journal*, 823(1):22–40, 2016.

[32] Y. Zhou, C. Xia, R. Keppens, C. Fang, and P. F. Chen. Three-dimensional MHD simulations of solar prominence oscillations in a magnetic flux rope. *The Astrophysical Journal*, 856(2):179–192, 2018.

[33] C. Xia, J. Teunissen, I. El Mellah, E. Chané, and R. Keppens. MPI-AMRVAC 2.0 for solar and astrophysical applications. *The Astrophysical Journal Supplement Series*, 234(2):30–55, 2018.

[34] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, 1989.

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

[36] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512, 1984.

[37] M. J. Berger. Data structures for adaptive grid generation. *SIAM Journal for Scientific and Statistical Computing*, 7(3):904–916, 1986.

[38] M. J. Aftosmis, M. J. Berger, and G. Adomavicius. A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries. Paper 2000-0808, AIAA, January 2000.

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

[40] C. Pantano, R. Deiterding, D. J. Hill, and D. I. Pullin. A low numerical dissipation patch-based adaptive mesh refinement method for large-eddy simulation of compressible flows. *Journal of Computational Physics*, 221:63–87, 2007.

[41] M. J. Berger and R. J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 35(6):2298–2316, 1998.

[42] M. J. Berger and R. J. LeVeque. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. Paper 89-1930, AIAA, June 1989.

[43] D. De Zeeuw and K. G. Powell. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics*, 104:56–68, 1993.

[44] D. L. De Zeeuw. *A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations*. PhD thesis, University of Michigan, September 1993.

[45] M. J. Aftosmis, M. J. Berger, and S. M. Murman. Applications of space-filling curves to Cartesian methods for CFD. Paper 2004-1232, AIAA, January 2004.

[46] J. J. Quirk and U. R. Hanebutte. A parallel adaptive mesh refinement algorithm. Report 93-63, ICASE, August 1993.

[47] X. Gao and C. P. T. Groth. Parallel adaptive mesh refinement scheme for three-dimensional turbulent non-premixed combustion. Paper 2008-1017, AIAA, January 2008.

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

[49] X. Gao and C. P. T. Groth. A parallel solution-adaptive method for three-dimensional turbulent non-premixed combusting flows. *Journal of Computational Physics*, 229(5):3250–3275, 2010.

[50] M. J. Williamschen and C. P. T. Groth. Parallel anisotropic block-based adaptive mesh refinement algorithm for three-dimensional flows. Paper 2013-2442, AIAA, June 2013.

[51] C. Burstedde, L. C. Wilcox, and O. Ghattas. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.

[52] F. E. Ham, F. S. Lien, and A. B. Strong. A Cartesian grid method with transient anisotropic adaptation. *Journal of Computational Physics*, 179:469–494, 2002.

[53] G. Iaccarino and F. Ham. LES on Cartesian grids with anisotropic refinement. In *Complex Effects in Large Eddy Simulations*, volume 56 of *Lecture Notes in Computational Science and Engineering*, pages 219–233. Springer, Berlin, 2007.

[54] D. Zuzio and J. L. Estivalezes. An efficient block parallel AMR method for two phase interfacial flow simulations. *Computers & Fluids*, 44(1):339–357, 2011.

[55] B. van der Holst, G. Tóth, I. V. Sokolov, K. G. Powell, J. P. Holloway, E. S. Myra, Q. Stout, M. L. Adams, J. E. Morel, S. Karni, Fryxell. B., and R. P. Drake. CRASH: A block-adaptive-mesh code for radiative shock hydrodynamics-implementation and verification. *The Astrophysical Journal Supplement Series*, 194(2):23–42, May 2011.

[56] B. van der Holst and R. Keppens. Hybrid block-AMR in Cartesian and curvilinear coordinates: MHD applications. *Journal of Computational Physics*, 226:925–946, 2007.

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

[58] X. Gao, S. A. Northrup, and C. P. T. Groth. Parallel solution-adaptive method for two-dimensional non-premixed combusting flows. *Progress in Computational Fluid Dynamics*, 11(2):76–95, 2011.

[59] C. K. S. Lam and C. P. T. Groth. Numerical prediction of three-dimensional non-equilibrium gaseous flows using the Gaussian moment closure. Paper 2011-3401, AIAA, June 2011.

[60] C. P. T. Groth and S. A. Northrup. Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. Paper 2005-5333, AIAA, June 2005.

[61] S. A. Northrup and C. P. T. Groth. Parallel implicit adaptive mesh refinement scheme for unsteady fully-compressible reactive flows. Paper 2013-2433, AIAA, June 2013.

[62] S. A. Northrup and C. P. T. Groth. Solution of laminar diffusion flames using a parallel adaptive mesh refinement algorithm. Paper 2005–0547, AIAA, January 2005.

[63] X. Gao and C. P. T. Groth. Parallel adaptive mesh refinement scheme for turbulent non-premixed combusting flow prediction. Paper 2006-1448, AIAA, January 2006.

[64] X. Gao and C. P. T. Groth. A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows. *International Journal of Computational Fluid Dynamics*, 20(5):349–357, 2006.

[65] P. K. Jha and C. P. T. Groth. Tabulated chemistry approaches for laminar flames: Evaluation of flame-prolongation of ILDM and flamelet methods. *Combustion Theory and Modelling*, 16(1):31–57, 2012.

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

[67] L. Ivan, H. De Sterck, A. Susanto, and C. P. T. Groth. High-order central ENO finite-volume scheme for hyperbolic conservation laws on three-dimensional cubed-sphere grids. *Journal of Computational Physics*, 282:157–182, 2015.

[68] L. Ivan, H. De Sterck, S. A. Northrup, and C. P. T. Groth. Multi-dimensional finite-volume scheme for hyperbolic conservation laws on three-dimensional solution-adaptive cubed-sphere grids. *Journal of Computational Physics*, 255:205–227, 2013.

[69] L. Ivan, H. De Sterck, S. A. Northrup, and C. P. T. Groth. Hyperbolic conservation laws on three-dimensional cubed-sphere grids: A parallel solution-adaptive simulation framework. *Journal of Computational Physics*, 255:205–227, 2013.

[70] L. Ivan, H. De Sterck, S. A. Northrup, and C. P. T. Groth. Three-dimensional MHD on cubed-sphere grids: Parallel solution-adaptive simulation framework. Paper 2011-3382, AIAA, June 2011.

[71] A. Susanto, L. Ivan, H. De Sterck, and C. P. T. Groth. High-order central ENO finite-volume scheme for ideal MHD. *Journal of Computational Physics*, 250:141–164, 2013.

[72] L. Ivan and C. P. T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. *Journal of Computational Physics*, 257:830–862, 2013.

[73] S. D. McDonald, M. R. J. Charest, and C. P. T. Groth. High-order CENO finite-volume schemes for multi-block unstructured mesh. Paper 2011-3854, AIAA, June 2011.

[74] L. Ivan and C. P. T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. Paper 2011-0367, AIAA, January 2011.

[75] L. Ivan and C. P. T. Groth. High-order central ENO finite-volume scheme with adaptive mesh refinement. Paper 2007-4324, AIAA, June 2007.

[76] Z. J. Zhang and C. P. T. Groth. Parallel high-order anisotropic block-based adaptive mesh refinement finite-volume scheme. Paper 2011-3695, AIAA, June 2011.

[77] L. Ivan. *Development of High-Order CENO Finite-Volume Schemes with Block-Based Adaptive Mesh Refinement*. PhD thesis, University of Toronto, August 2011.

[78] L. Freret, L. Ivan, H. De Sterck, and C. P. T. Groth. High-order finite-volume method with block-based AMR for magnetohydrodynamics flows. *Journal of Scientific Computing*, 79(1):176–208, April 2019.

[79] L. Freret, L. Ivan, H. De Sterck, and C. P. T. Groth. A high-order finite-volume method with anisotropic AMR for ideal MHD flows. Paper 2017-0845, AIAA, January 2017.

[80] M. R. J. Charest, C. P. T. Groth, and Ö. L. Gülder. Solution of the equation of radiative transfer using a Newton-Krylov approach and adaptive mesh refinement. *Journal of Computational Physics*, 231:3023–3040, 2012.

[81] M. R. J. Charest and C. P. T. Groth. A high-order central ENO finite-volume scheme for three-dimensional turbulent reactive flows on unstructured mesh. Paper 2013-2567, AIAA, June 2013.

[82] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West Journal of Numerical Mathematics*, 4:237–264, 1996.

[83] V. Heuveline and R. Rannacher. Duality-based adaptivity in the *hp*-finite element method. *Journal of Numerical Mathematics*, 11:95–103, 2003.

[84] R. Becker, V. Heuveline, and R. Rannacher. An optimal control approach to adaptivity in computational fluid dynamics. *International Journal for Numerical Methods in Fluids*, 40:105–120, 2002.

[85] D. A. Venditti and D. L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *Journal of Computational Physics*, 164:204–227, 2000.

[86] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176:40–69, 2002.

[87] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22–46, 2003.

[88] M. Nemec and M. J. Aftosmis. Adjoint sensitivity computations for an embedded-boundary Cartesian mesh method. *Journal of Computational Physics*, 227(4):2724–2742, 2008.

[89] M. Nemec and M. J. Aftosmis. Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes. Paper 2007-4187, AIAA, June 2007.

[90] M. Ceze and K. J. Fidkowski. Anisotropic *hp*-adaptation framework for functional prediction. *AIAA Journal*, 51(2):492–509, 2012.

[91] M. Maier and R. Rannacher. Duality-based adaptivity in finite element discretization of heterogeneous multiscale problems. *Journal of Numerical Mathematics*, 24(3):167–187, 2016.

[92] M. Yano and D. Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *Journal of Computational Physics*, 231(22):7626–7649, 2012.

[93] S. Copeland, A. Lonkar, F. Palacios, and J. Alonso. Adjoint-based goal-oriented mesh adaptation for nonequilibrium hypersonic flows. Paper 2013-0552, AIAA, January 2013.

[94] T. Leicht and R. Hartmann. Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations. *Journal of Computational Physics*, 229(19):7344–7360, 2010.

[95] R. Hartmann, J. Held, and T. Leicht. Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations. *Journal of Computational Physics*, 230:4268–4284, May 2011.

[96] D. Estep, S. Tavener, and T. Wildey. *a posteriori* error estimation and adaptive mesh refinement for a multiscale operator decomposition approach to fluid-solid heat transfer. *Journal of Computational Physics*, 229(11):4143–4158, 2010.

[97] T. Dunne. An Eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation. *International Journal for Numerical Methods in Fluids*, 51(9-10):1017–1039, 2006.

[98] P. W. Power, C. C. Pain, M. D. Piggott, F. Fang, G. J. Gorman, A. P. Umpleby, A. J. H. Goddard, and I. M. Navon. Adjoint *a posteriori* error measures for anisotropic mesh optimisation. *Computers & Mathematics with Applications*, 52(8):1213–1242, 2006.

[99] B. R. Ahrabi, W. K. Anderson, and J. C. Newman. An adjoint-based hp-adaptive stabilized finite-element method with shock capturing for turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 318(Supplement C):1030–1065, 2017.

[100] M. Wintzer, M. Nemec, and M. Aftosmis. Adjoint-based adaptive mesh refinement for sonic boom prediction. Paper 2008-6593, AIAA, August 2008.

[101] M. Park and D. Darmofal. Output-adaptive tetrahedral cut-cell validation for sonic boom prediction. Paper 2008-6594, AIAA, August 2008.

[102] W. Jones, E. Nielsen, and M. Park. Validation of 3D adjoint based error estimation and mesh adaptation for sonic boom prediction. Paper 2006-1150, AIAA, January 2006.

[103] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *Journal of Computational Physics*, 229(8):2866–2897, 2010.

[104] A. Loseille, A. Dervieux, and F. Alauzet. On 3-D goal-oriented anisotropic mesh adaptation applied to inviscid flows in aeronautics. Paper 2010-1067, AIAA, January 2010.

[105] A. Belme, A. Dervieux, and F. Alauzet. Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. *Journal of Computational Physics*, 231:6323–6348, 2012.

[106] R. P. Dwight. Goal-oriented mesh adaptation for finite volume methods using a dissipation-based error indicator. *International Journal for Numerical Methods in Fluids*, 56(8):1193–1200, 2008.

[107] J. Peter, M. Nguyen-Dinh, and P. Trontin. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates-with applications to Euler flows. *Computers & fluids*, 66:194–214, 2012.

[108] L. Wang and D. J. Mavriplis. Adjoint-based *h-p* adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *Journal of Computational Physics*, 228(20):7643–7661, 2009.

[109] E. H. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*-anisotropic meshes ii: A posteriori error analysis and adaptivity. *Applied numerical mathematics*, 59(9):2179–2194, 2009.

[110] S. Sun and M. F. Wheeler. A posteriori error estimation and dynamic adaptivity for symmetric discontinuous Galerkin approximations reactive transport systems. *Computer Methods in Applied Mechanics and Engineering*, 195:632–652, 2006.

[111] R. Becker, M. Braack, and B. Vexler. Numerical parameter estimation for chemical models in multidimensional reactive flows. *Combustion Theory and Modelling*, 8:661–682, 2004.

[112] R. Becker, M. Braack, and B. Vexler. Parameter identification for chemical models in combustion problems. *Applied Numerical Mathematics*, 54(3):519–536, 2005. Selected papers from the 16th Chemnitz Finite Element Symposium 2003.

[113] R. Becker, M. Braack, and R. Rannacher. Numerical simulation of laminar flames at low Mach number by adaptive finite elements. *Combustion Theory and Modelling*, 3:503–534, September 1999.

[114] M. Braack, K. H. Schlünzen, N. Taschenberger, and M. Uphoff. Error estimation and adaptive chemical transport modeling. *Meteorologische Zeitschrift*, 23(4):341–348, 09 2014.

[115] A. Bourlioux, A. Ern, and P. Turbis. A posteriori error estimation for subgrid flamelet models. *Multiscale Modeling & Simulation*, 8(2):481–497, 2010.

[116] E. Burman, A. Ern, and V. Giovangigli. An adaptive finite element method with crosswind diffusion for low Mach, steady, laminar combustion. *Journal of Computational Physics*, 188:472–492, July 2003.

[117] M. Braack and A. Ern. A posteriori control of modeling errors and discretization errors. *Multiscale Modeling and Simulation*, 1(2):221–238, 2003.

[118] L. Formaggia, S. Micheletti, and S. Perotto. Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems. *Applied Numerical Mathematics*, 51(4):511–533, 2004. Applied Scientific Computing: Advances in Grid Generation, Approximation and Numerical Modeling.

[119] P. W. Power, M. D. Piggott, F. Fang, G. J. Gorman, C. C. Pain, D. P. Marshall, A. J. H. Goddard, and I. M. Navon. Adjoint goal-based error norms for adaptive mesh ocean modelling. *Ocean Modelling*, 15(1):3–38, 2006. The Third International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows.

[120] B. N. Davis and R. J. LeVeque. Adjoint methods for guiding adaptive mesh refinement in tsunami modeling. *Pure and Applied Geophysics*, 173(12):4055–4074, 2016.

[121] A. Zdunek, W. Rachowicz, and N. Sehlstedt. Toward *hp*-adaptive solution of 3D electromagnetic scattering from cavities. *Computers & Mathematics with Applications*, 49(1):23–38, 2005.

[122] I. Gomez-Revuelto, L. E. Garcia-Castillo, and M. Salazar-Palma. Goal-oriented self-adaptive *hp*-strategies for finite element analysis of electromagnetic scattering and radiation problems. *Progress In Electromagnetics Research*, 125:459–482, 2012.

[123] D. Pardo, L. Demkowicz, C. Torres-Verdín, and L. Tabarovsky. A goal-oriented *hp*-adaptive finite element method with electromagnetic applications. part i: electrostatics. *International Journal for Numerical Methods in Engineering*, 65(8):1269–1309, 2006.

[124] D. Pardo, L. Demkowicz, C. Torres-Verdín, and M. Paszynski. A self-adaptive goal-oriented *hp*-finite element method with electromagnetic applications. part ii: Electrodynamics. *Computer methods in applied mechanics and engineering*, 196(37-40):3585–3597, 2007.

[125] D. Lathouwers. Goal-oriented spatial adaptivity for the S_N equations on unstructured triangular meshes. *Annals of Nuclear Energy*, 38(6):1373–1381, 2011.

[126] X. Ju and R. Mahnken. Goal-oriented adaptivity for linear elastic micromorphic continua based on primal and adjoint consistency analysis. *International Journal for Numerical Methods in Engineering*, 112(8):1017–1039, 2017.

[127] D. E. Hart, C. E. Goodyer, M. Berzins, P. K. Jimack, and L. E. Scales. Adjoint error estimation for EHL-like models. *International Journal for Numerical Methods in Fluids*, 47(10-11):1069–1075, 2005.

[128] A. Joshi, W. Bangerth, and E. M. Sevick-Muraca. Adaptive finite element based tomography for fluorescence optical imaging in tissue. *Opt. Express*, 12(22):5402–5417, Nov 2004.

[129] C. Burstedde, O. Ghattas, G. Stadler, T. Tu, and L. C. Wilcox. Parallel scalable adjoint-based adaptive solution of variable-viscosity Stokes flow problems. *Computer Methods in Applied Mechanics and Engineering*, 198(21):1691–1700, 2009. Advances in Simulation-Based Engineering Sciences – Honoring J. Tinsley Oden.

[130] L. Mathelin and O. Le Maître. Dual-based *a posteriori* error estimate for stochastic finite element methods. *Communications in Applied Mathematics and Computational Science*, 2(1):83–115, 2007.

[131] Y. Li and K. Key. 2D marine controlled-source electromagnetic modeling: Part 1-an adaptive finite-element algorithm. *Geophysics*, 72(2):WA51–WA62, 2007.

[132] V. J. Ervin and W.J. Layton. *A posteriori* error estimation for two level discretizations of flows of electrically conducting, incompressible fluids. *Computers & Mathematics with Applications*, 31(11):105–114, 1996.

[133] J. Zhao, S. Mao, and W. Zheng. Anisotropic adaptive finite element method for magnetohydrodynamic flow at high Hartmann numbers. *Applied Mathematics and Mechanics*, 37(11):1479–1500, Nov 2016.

[134] S. Lankalapalli, J. E. Flaherty, M. S. Shephard, and H. Strauss. An adaptive finite element method for magnetohydrodynamics. *Journal of Computational Physics*, 225:363–381, July 2007.

[135] H. Baty. FINMHD: an adaptive finite-element code for magnetic reconnection and plasmoid chains formation in magnetohydrodynamics. *The Astrophysical Journal Supplement Series*, 243(2):23–36, 2019.

[136] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[137] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, Sep 1988.

[138] M. Nemec and D. W. Zingg. Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations. *AIAA Journal*, 40(6):1146–1154, 2002.

[139] A. H. Truong, C. A. Oldfield, and D. W. Zingg. Mesh movement for a discrete-adjoint Newton-Krylov algorithm for aerodynamic optimization. *AIAA Journal*, 46(7):1695–1704, 2008.

[140] J. E. Hicken and D. W. Zingg. Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA Journal*, 48(2):400–413, 2010.

[141] A. C. Marta and J. J. Alonso. Toward optimally seeded airflow on hypersonic vehicles using control theory. *Computers & Fluids*, 39(9):1562–1574, 2010.

[142] A. C. Marta. *Rapid development of discrete adjoint solvers with applications to magnetohydrodynamic flow control.* PhD thesis, Stanford University, 2007.

[143] A. C. Marta, C. A. Mader, J. R. R. A. Martins, E. Van der Weide, and J. J. Alonso. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. *International Journal of Computational Fluid Dynamics*, 21(9-10):307–327, 2007.

[144] A. Marta and J. Alonso. Discrete adjoint formulation for the ideal MHD equations. Paper 2006-3345, AIAA, June 2006.

[145] N. Narechania, L. Freret, and C. P. T. Groth. Block-based anisotropic AMR with *a posteriori* adjoint-based error estimation for three-dimensional inviscid and viscous flows. Paper 2017-4113, AIAA, June 2017.

[146] C. Ngigi, L. Freret, and C. P. T. Groth. Comparison of *h*-and *p*-derived output-based error estimates for directing anisotropic adaptive mesh refinement in three-dimensional inviscid flows. Paper 2018-0829, AIAA, January 2018.

[147] H. Schlichting and K. Gersten. *Boundary-Layer Theory.* Springer, New York, 8th edition, 2000.

[148] F. M. White. *Fluid Mechanics*. McGraw-Hill, New York, 5th edition, 2003.

[149] H. De Sterck, A. Csík, D. V. Abeele, S. Poedts, and H. Deconinck. Stationary two-dimensional magnetohydrodynamic flows with shocks: Characteristic analysis and grid convergence study. *Journal of Computational Physics*, 166:28–62, 01 2001.

[150] H. De Sterck, B. C. Low, and S. Poedts. Characteristic analysis of a complex two-dimensional magnetohydrodynamic bow shock flow with steady compound shocks. *Physics of Plasmas*, 6(3):954–969, 1999.

[151] C. R. Evans and J. F. Hawley. Simulation of magnetohydrodynamic flows-a constrained transport method. *The Astrophysical Journal*, 332:659–677, 1988.

[152] W. Dai and P. R. Woodward. On the divergence-free condition and conservation laws in numerical simulations for supersonic magnetohydrodynamical flows. *Astrophysical Journal*, 494(1):317–335, 1998.

[153] W. Dai and P. R. Woodward. A simple finite difference scheme for multidimensional magnetohydrodynamical equations. *Journal of Computational Physics*, 142(2):331–369, 1998.

[154] D. Ryu, F. Miniati, T. W. Jones, and A. Frank. A divergence-free upwind code for multidimensional magnetohydrodynamic flows. *Astrophysical Journal*, 509(1):244–255, 1998.

[155] D. S. Balsara and D. S. Spicer. A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations. *Journal of Computational Physics*, 149(2):270–292, 1999.

[156] J. U. Brackbill and D. C. Barnes. The effect of nonzero $\nabla\cdot$ B on the numerical solution of the magnetohydrodynamic equations. *Journal of Computational Physics*, 35:426–430, 1980.

[157] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.

[158] A. Dedner, F. Kemm, D. Kröner, C. D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the MHD equations. *Journal of Computational Physics*, 175(2):645–673, 2002.

[159] J. Teunissen and R. Keppens. A geometric multigrid library for quadtree/octree AMR grids coupled to MPI-AMRVAC. *Computer Physics Communications*, 245:106866–106880, 2019.

[160] G. Tóth. The $\nabla\cdot$ B= 0 constraint in shock-capturing magnetohydrodynamics codes. *Journal of Computational Physics*, 161(2):605–652, 2000.

[161] K. G. Powell. An approximate Riemann solver for magnetohydrodynamics (that works in more than one dimension). Report 94-24, ICASE, July 1994.

[162] S. K. Godunov. An interesting class of quasilinear systems. *Soviet Mathematics Doklady*, 2:947–949, 1961.

[163] S. K. Godunov. Symmetric form of the equations of magnetohydrodynamics. *Numerical Methods for Mechanics of Continuum Medium*, 1:26–34, 1972.

[164] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.

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

[166] J. J. Gottlieb and C. P. T. Groth. Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. *Journal of Computational Physics*, 78:437–458, 1988.

[167] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[168] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis*, 25:294–318, 1988.

[169] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1):25–34, 1994.

[170] T. Linde. A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws. *International Journal for Numerical Methods in Fluids*, 40:391–402, 2002.

[171] M.-S. Liou. A sequel to AUSM: AUSM$^+$. *Journal of Computational Physics*, 129:364–382, 1996.

[172] M.-S. Liou. A sequel to AUSM, part ii: AUSM$^+$-up for all speeds. *Journal of Computational Physics*, 214:137–170, 2006.

[173] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics of Computation*, 38(158):339–374, 1982.

[174] W. Dai and P. R. Woodward. An approximate Riemann solver for ideal magnetohydrodynamics. *Journal of Computational Physics*, 111:354–372, 1994.

[175] A. L. Zachary and P. Colella. A higher-order Godunov method for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 99:341–347, 1992.

[176] M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 75:400–422, 1988.

[177] S. Li. An HLLC Riemann solver for magneto-hydrodynamics. *Journal of Computational Physics*, 203(1):344–357, 2005.

[178] K. F. Gurski. An HLLC-type approximate Riemann solver for ideal magnetohydrodynamics. *SIAM Journal on Scientific Computing*, 25(6):2165–2187, 2004.

[179] T. Miyoshi and K. Kusano. A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics. *Journal of Computational Physics*, 208(1):315–344, 2005.

[180] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions. Paper 93-0880, AIAA, January 1993.

[181] P. Cargo and G. Gallice. Roe matrices for ideal MHD and systematic construction of Roe matrices for systems of conservation laws. *Journal of Computational Physics*, 136(2):446–466, 1997.

[182] S. R. Mathur and J. Y. Murthy. A pressure-based method for unstructured meshes. *Numerical Heat Transfer*, 31:191–215, 1997.

[183] J. S. Sachdev, C. P. T. Groth, and J. J. Gottlieb. A parallel solution-adaptive scheme for predicting multi-phase core flows in solid propellant rocket motors. *International Journal of Computational Fluid Dynamics*, 19(2):159–177, 2005.

[184] J. G. McDonald, J. S. Sachdev, and C. P. T. Groth. Application of Gaussian moment closure to micro-scale flows with moving and embedded boundaries. *AIAA Journal*, 51(9):1839–1857, 2014.

[185] W. A. Mulder and B. van Leer. Experiments with implicit upwind methods for the Euler equations. *Journal of Computational Physics*, 59:232–246, 1985.

[186] H. Lomax, T. H. Pulliam, and D. W. Zingg. *Fundamentals of Computational Fluid Dynamics*. Springer-Verlag, New York, 2001.

[187] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear equations. *SIAM Journal for Scientific and Statistical Computing*, 7(3):856–869, 1986.

[188] J.C. Butcher. A history of Runge-Kutta methods. *Applied Numerical Mathematics*, 20(3):247–260, 1996.

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

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

[191] S. K. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. Paper 2000-0667, AIAA, January 2000.

[192] S. K. Nadarajah. *A Discrete Adjoint Approach to Aerodynamic Shape Optimization*. PhD thesis, Stanford University, January 2003.

[193] M. Nemec and D. W. Zingg. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*, 42(6):1057–1065, 2004.

[194] S. K. Nadarajah and A. Jameson. Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods. *AIAA Journal*, 45(7), July 2007.

[195] J. P. Thomas, K. C. Hall, and E. H. Dowell. Discrete adjoint approach for modeling unsteady aerodynamic design sensitivities. *AIAA Journal*, 43(9), September 2005.

[196] M. Ceze and K. Fidkowski. A robust adaptive solution strategy for high-order implicit CFD solvers. Paper 2011-3696, AIAA, 2011.

[197] K. J. Fidkowski and D. L. Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.

[198] J. E. Hicken and D. W. Zingg. Dual consistency and functional accuracy: a finite-difference perspective. *Journal of Computational Physics*, 256:161–182, 2014.

[199] C. H. Bischof, P. D. Hovland, and B. Norris. On the implementation of automatic differentiation tools. *Higher-Order and Symbolic Computation*, 21(3):311–331, 2008.

[200] B. Carpenter, M. D. Hoffman, M. Brubaker, D. Lee, P. Li, and M. Betancourt. The Stan Math Library: Reverse-mode automatic differentiation in C++. *arXiv preprint arXiv:1509.07164*, abs/1509.07164, 2015.

[201] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The Complex-step Derivative Approximation. *ACM Trans. Math. Softw.*, 29(3):245–262, September 2003.

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

[203] M. A. Heroux, J. M. Willenbring, and R. Heaphy. Trilinos Developers Guide. Technical Report SAND2003-1898, Sandia National Laboratories, 2003.

[204] M. A. Heroux, J. M. Willenbring, and R. Heaphy. Trilinos Developers Guide Part II: ASCI Software Quality engineering Practices Version 1.0. Technical Report SAND2003-1899, Sandia National Laboratories, 2003.

[205] M. Sala, M. A. Heroux, and D. M. Day. Trilinos Tutorial. Technical Report SAND2004-2189, Sandia National Laboratories, 2004.

[206] M. A. Heroux and J. M. Willenbring. Trilinos Users Guide. Technical Report SAND2003-2952, Sandia National Laboratories, 2003.

[207] M. A. Heroux. Epetra Reference Manual, 2.0 Edition. Technical report, Sandia National Laboratories, 2002.

[208] M. A. Heroux. AztecOO Users Guide. Technical Report SAND2004-3796, Sandia National Laboratories, 2004.

[209] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[210] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 24(3):979–1004, 2003.

[211] R. Kannan and Z. J. Wang. A study of viscous flux formulations for a p-multigrid spectral volume Navier-Stokes solver. *Journal of Scientific Computing*, 41(2):165–199, 2009.

[212] R. Kannan and Z. J. Wang. A variant of the LDG flux formulation for the spectral volume method. *Journal of Scientific Computing*, 46(2):314–328, 2011.

[213] Y. Sun, Z. J. Wang, and Y. Liu. Efficient implicit non-linear LU-SGS approach for compressible flow computation using high-order spectral difference method. *Communications in Computational Physics*, 5(2–4):760–778, 2009.

[214] A. Jameson D. J. Mavriplis and J. Martinelli. Multigrid solution of the Navier-Stokes equations on triangular meshes. Paper 89-0120, AIAA, January 1989.

[215] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 51(9-10):1131–1156, 2006.

[216] D. H. Mackay and A. R. Yeates. The Sun's Global Photospheric and Coronal Magnetic Fields: Observations and Models. *Living Reviews in Solar Physics*, 9:6–68, November 2012.

[217] M. D. Altschuler and G. Newkirk. Magnetic fields and the structure of the solar corona. *Solar Physics*, 9(1):131–149, Sep 1969.

[218] K. Schatten, J. Wilcox, and N. Ness. A model of interplanetary and coronal magnetic fields. *Solar Physics*, 6(3):442–455, 1969.

[219] M. Altschuler, R. Levine, M. Stix, and J. Harvey. High resolution mapping of the magnetic field of the solar corona. *Solar Physics*, 51(2):345–375, 1977.

[220] K. H. Schatten. Current sheet magnetic model for the solar corona. *Cosmic Electrodynamics*, 2:232–242, 1971.

[221] X. Zhao and J. Hoeksema. A coronal magnetic field with horizontal volume and sheet currents. *Solar Physics*, 151(1):91–105, 1994.

[222] P. Riley, J. A. Linker, Z. Mikić, R. Lionello, S. A. Ledvina, and J. G. Luhmann. A comparison between global solar magnetohydrodynamic and potential field source surface model results. *The Astrophysical Journal*, 653(2):1510–1516, 2006.

[223] M. J. Owens, H. E. Spence, S. McGregor, W. J. Hughes, J. M. Quinn, C. N. Arge, P. Riley, J. Linker, and D. Odstrčil. Metrics for solar wind prediction models: Comparison of empirical, hybrid, and physics-based schemes with 8 years of l1 observations. *Space Weather*, 6(8):S08001–1–S08001–16, 2008.

[224] L. Nikolic. Modelling the magnetic field of the solar corona with potential-field source-surface and Schatten current sheet models. 2016.

[225] D. F. Webb and T. A. Howard. Coronal mass ejections: Observations. *Living Reviews in Solar Physics*, 9(1):3–85, June 2012.

[226] T. Howard. *Coronal mass ejections: an introduction*. Springer, 2011.

[227] T. Detman, Z. Smith, M. Dryer, C. D. Fry, C. N. Arge, and V. Pizzo. A hybrid heliospheric modeling system: Background solar wind. *Journal of Geophysical Research: Space Physics*, 111(A7):A07102–1–A07102–14, 2006.

[228] D. Odstrčil. Modeling 3-D solar wind structure. *Advances in Space Research*, 32(4):497–506, 2003.

[229] A. Parsons, D. Biesecker, D. Odstrčil, G. Millward, S. Hill, and V. Pizzo. Wang-Sheeley-Arge-ENLIL cone model transitions to operations. *Space Weather*, 9(3):S03004–1–S03004–2, 2011.

[230] Y. M. Wang and N. R. Shelley Jr. Solar wind speed and coronal flux-tube expansion. *The Astrophysical Journal*, 355:726–732, 1990.

[231] Y. M. Wang and N. R. Sheeley Jr. Solar implications of ULYSSES interplanetary field measurements. *The Astrophysical Journal*, 447:143–147, 1995.

[232] Y. Wang, N. R. Shelley Jr, and J. L. Phillips. Coronal flux-tube expansion and the polar wind. *Advances in Space Research*, 16(9):365–365, 1995.

[233] C. N. Arge and V. J. Pizzo. Improvement in the prediction of solar wind conditions using near-real time solar magnetic field updates. *Journal of Geophysical Research: Space Physics*, 105(A5):10465–10479, 2000.

[234] C. N. Arge, D. Odstrčil, V. J. Pizzo, and L. R. Mayer. Improved method for specifying solar wind speed near the sun. *American Institute of Physics Conference Proceedings*, 679(1):190–193, 2003.

[235] C. N. Arge, J. G. Luhmann, D. Odstrčil, C. J. Schrijver, and Y. Li. Stream structure and coronal sources of the solar wind during the May 12th, 1997 CME. *Journal of Atmospheric and Solar-Terrestrial Physics*, 66(15-16):1295–1309, 2004.

[236] N. R. Sheeley Jr. Origin of the Wang-Sheeley-Arge solar wind model. *History of Geo- and Space Sciences*, 8(1):21–28, 2017.

[237] D. N. Baker, G. Poh, D. Odstrčil, C. N. Arge, M. Benna, C. L. Johnson, H. Korth, D. J. Gershman, G. C. Ho, W. E. McClintock, T. A. Cassidy, A. Merkel, J. M. Raines, D. Schriver, J. A. Slavin, S. C. Solomon, P. M. Travnicek, R. M. Winslow, and T. H. Zurbuchen. Solar wind forcing at Mercury: WSA-ENLIL model results. *Journal of Geophysical Research: Space Physics*, 118(1):45–57, 2013.

[238] A. Taktakishvili, M. Kuznetsova, P. MacNeice, M. Hesse, L. Rastätter, A. Pulkkinen, A. Chulaki, and D. Odstrčil. Validation of the coronal mass ejection predictions at the Earth orbit estimated by ENLIL heliosphere cone model. *Space Weather*, 7(3):S03004–1–S03004–7, 2009.

[239] A. Taktakishvili, A. Pulkkinen, P. MacNeice, M. Kuznetsova, M. Hesse, and D. Odstrčil. Modeling of coronal mass ejections that caused particularly large geomagnetic storms using ENLIL heliosphere cone model. *Space Weather*, 9(6):S06002–1–S06002–12, 2011.

[240] J. Hinterreiter, J. Magdalenic, M. Temmer, C. Verbeke, I. C. Jebaraj, E. Samara, E. Asvestari, S. Poedts, J. Pomoell, E. Kilpua, L. Rodriguez, C. Scolini, and A. Isavnin. Assessing the performance of EUHFORIA modeling the background solar wind. *Solar Physics*, 294(12):170–188, 2019.

[241] J. Pomoell and S. Poedts. EUHFORIA: European heliospheric forecasting information asset. *Journal of Space Weather and Space Climate*, 8:A35–A48, 2018.

[242] C. Verbeke, J. Pomoell, and S. Poedts. The evolution of coronal mass ejections in the inner heliosphere: implementing the spheromak model with EUHFORIA. *Astronomy & Astrophysics*, 627:A111–A125, 2019.

[243] C. Scolini, L. Rodriguez, M. Mierla, J. Pomoell, and S. Poedts. Observation-based modelling of magnetised coronal mass ejections with EUHFORIA. *Astronomy & Astrophysics*, 626:A122–A148, 2019.

[244] A. Nakamizo, T. Tanaka, Y. Kubo, S. Kamei, H. Shimazu, and H. Shinagawa. Development of the 3-D MHD model of the solar corona-solar wind combining system. *Journal of Geophysical Research: Space Physics*, 114(A7):A07109–1–A07109–16, 2009.

[245] D. Shiota, R. Kataoka, Y. Miyoshi, T. Hara, C. Tao, K. Masunaga, Y. Futaana, and N. Terada. Inner heliosphere MHD modeling system applicable to space weather forecasting for the other planets. *Space Weather*, 12(4):187–204, 2014.

[246] D. Shiota and R. Kataoka. Magnetohydrodynamic simulation of interplanetary propagation of multiple coronal mass ejections with internal magnetic flux rope (SUSANOO-CME). *Space Weather*, 14(2):56–75, 2016.

[247] A. Kageyama and T. Sato. "Yin-Yang grid": An overset grid in spherical geometry. *Geochemistry, Geophysics and Geosystems*, 5(9):Q09005–1–Q09005–15, 2004.

[248] L. Nikolic, L. Trichtchenko, and D. Boteler. A numerical framework for operational solar wind prediction. *Plasma and Fusion Research*, 9:3406099–3406099, 2014.

[249] National Solar Observatory, Global Oscillation Network Group. http://gong.nso.edu.

[250] K. Hayashi, M. Kojima, M. Tokumaru, and K. Fujiki. MHD tomography using interplanetary scintillation measurement. *Journal of Geophysical Research: Space Physics*, 108(A3):1102–1–1102–13, 2003.

[251] R. Kataoka, T. Ebisuzaki, K. Kusano, D. Shiota, S. Inoue, T. T. Yamamoto, and M. Tokumaru. Three-dimensional MHD modeling of the solar wind structures associated with 13 December 2006 coronal mass ejection. *Journal of Geophysical Research: Space Physics*, 114(A10):A10102–1–A10102–10, 2009.

[252] E. N. Parker. Dynamics of the interplanetary gas and magnetic fields. *Astrophysical Journal*, 128(3):664–676, 1958.

[253] H. Alfvén. On the theory of comet tails. *Tellus*, 9(1):92–96, 1957.

[254] T. Sakurai. Magnetohydrodynamic solar/stellar wind models. *Computer Physics Reports*, 12(4):247–273, 01 1990.

[255] R. M. Winglee, J. Slough, T. Ziemba, and A. Goodson. Mini-magnetospheric plasma propulsion: Tapping the energy of the solar wind for spacecraft propulsion. *Journal of Geophysical Research: Space Physics*, 105(A9):21067–21077, 2000.

[256] P. Janhunen. Electric sail for spacecraft propulsion. *Journal of Propulsion and Power*, 20(4):763–764, 2004.