Application of r-Adaptation Techniques for Discretization Error Improvement in CFD

William C. Tyson

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Aerospace Engineering

Christopher J. Roy, Chair
Jeffrey T. Borggaard
Heng Xiao

December 8, 2015
Blacksburg, Virginia

Keywords: CFD, mesh adaptation, truncation error, discretization error

Application of r-Adaptation Techniques for Discretization Error Improvement in CFD

William C. Tyson

(ABSTRACT)

Computational fluid dynamics (CFD) has proven to be an invaluable tool for both engineering design and analysis. As the performance of engineering devices become more reliant upon the accuracy of CFD simulations, it is necessary to not only quantify and but also to reduce the numerical error present in a solution. Discretization error is often the primary source of numerical error. Discretization error is introduced locally into the solution by truncation error. Truncation error represents the higher order terms in an infinite series which are truncated during the discretization of the continuous governing equations of a model. Discretization error can be reduced through uniform grid refinement but is often impractical for typical engineering problems. Grid adaptation provides an efficient means for improving solution accuracy without the exponential increase in computational time associated with uniform grid refinement. Solution accuracy can be improved through local grid refinement, often referred to as h-adaptation, or by node relocation in the computational domain, often referred to as r-adaptation. The goal of this work is to examine the effectiveness of several r-adaptation techniques for reducing discretization error. A framework for geometry preservation is presented, and truncation error is used to drive adaptation. Sample problems include both subsonic and supersonic inviscid flows. Discretization error reductions of up to an order of magnitude are achieved on adapted grids.

This was work funded by the Air Force Office of Scientific Research (AFOSR) Computational Mathematics Program managed by Dr. Fariba Fahroo under grant FA9550-12-1-0173.

To my wife, Abigail,

for her endless love and support

Acknowledgments

First, and foremost, I thank God for His abundant love and for giving me the strength and perseverance to make it through the long hours required to complete this work. I thank Him for allowing me the extraordinary opportunity to pursue this degree as well as all the blessings He has afforded me throughout my life. God is good, all the time.

I thank my research advisor, Dr. Chris Roy, for funding me throughout this work and for always being available to provide me with much needed guidance. I also thank him for teaching me how to conduct and present meaningful research.

I thank my committee members, Dr. Jeff Borggaard and Dr. Heng Xiao, for agreeing to serve on my committee and for offering their insight.

I thank our graduate program coordinator, Rachel Hall Smith, for all her administrative help completing the necessary requirements for this degree.

I thank our system administrators, Steve Edwards and Jonathan Spence, for all your help keeping the lab computers up and running.

I thank Chip and Andrew for many insightful conversations in the lab and sharing a laugh every now and then.

I thank Joe for agreeing to mentor me as an undergraduate. Your guidance, kindness, and friendship have meant a great deal to me.

Lastly, I thank my family. To my brother, Matthew, and my sister, Holly, thank you for your "encouragement" throughout my academic career. Without you to push me, I would likely not be where I am today. To my mother, Theresa, and my father, Edward, thank you for your unconditional love and support throughout my life and for teaching me to always do my best. A son could not ask for better parents. To my wife, Abigail, thank you for your patience, your kindness, your support, and for all the sacrifices that you make for me. I love you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since its inception, computational fluid dynamics (CFD) has become a powerful tool for both engineering design and analysis. CFD simulations can provide valuable insight into the flow physics of a particular problem while saving both time and monetary resources over an equivalent experimental setup. But, as CFD becomes more the norm in engineering practice, it can be easy to forget the inherent error in each simulation including modeling error, numerical error, and other inherent uncertainties in the system. Numerical errors are typically categorized as one of the following: round-off error, iterative error, and discretization error. With the increased precision and computing power of modern computers, round-off error and iterative error are not generally the leading contributors to the overall error in a solution. Discretization error, which arises from the required discretization of continuous governing equations, is often the primary source of numerical error. Discretization error is introduced locally into the solution by truncation error. Truncation error represents the higher order terms which are truncated during the discretization process. Depending upon the chosen discretization scheme, such as a finite difference, finite volume, or finite element discretization, and the geometry definition, i.e. the grid, the amount of discretization error in a solution can be quite substantial. Since exact solutions are generally not available for precisely assessing the amount of discretization error present, much research has been conducted to develop and test methods for accurately estimating discretization error. A comprehensive review of discretization error estimators can be found in [1].

As the designs of engineering devices become more reliant upon the accuracy of computational models, it is becoming increasingly necessary to not only accurately quantify discretization error, but also to reduce it. The simplest approach to reducing discretization error is through uniform grid refinement. With uniform grid refinement, driving the grid spacing terms present in the truncation error to zero allows for less discretization error to be introduced into the solution. But, this method for discretization error reduction is often too expensive. With typical engineering problems requiring millions of degrees of freedom, uniform refinement quickly becomes impractical. To this end, methods for grid adaptation

were developed to combat the exponential increase in solution cost associated with uniform grid refinement. Intelligent techniques for local grid refinement, referred to as h-adaptation, and grid movement, referred to as r-adaptation, have allowed engineers to obtain more accurate solutions at a fraction of the cost of uniform grid refinement. h-Adaptation techniques are particularly suitable for unstructured grids where the data structure allows for cell subdivision to be easily implemented. r-Adaptation, on the other hand, can be applied to both structured and unstructured grids since a particular data structure is not required for node relocation within the computational domain. The primary focus of this work is to examine the effectiveness of r-adaptation for reducing the amount of discretization error present in a solution. Several adaptation schemes are applied to problems in both the subsonic and supersonic flow regimes. Discretization error reductions are assessed and comparisons are made.

1.1 Previous Work

Many of the advances in grid adaptation could not have been made without the seminal works of Thompson et al. [2] and Steger and Chaussee [3] who were able to successfully generate smooth, body-fitted grids around an arbitrary number of bodies by solving a set of partial differential equations (PDEs) for the grid node locations. Many researchers attempted to make these schemes adaptive by adding inhomogeneous source terms to the governing grid generation equations in hopes of controlling the location of grid nodes [4, 5, 6]. Anderson [7, 8, 9, 10] in particular was quite successful at accomplishing adaptivity by extending the work of Winslow [11]. For the structured flow solvers that were available at the time, grid quality was important for maintaining a smooth transformation between physical space and the computational space where the governing equations for fluid flow were solved. With this in mind, Brackbill and Saltzman [12] also investigated methods for grid generation which combined both adaptivity and grid quality optimization.

Physically-based adaptation schemes also emerged as alternatives to PDE-based schemes. Gnoffo [13] successfully developed an adaptation scheme based upon a linear spring analogy where the grid can be thought of as a system of springs whose stiffnesses are defined by an adaptation parameter. The resulting spring system is solved for the adapted grid. Nakahashi and Deiwert [14] extended the method to include torsional springs for added grid quality control. Eiseman [15] developed an adaptation scheme based upon a center of mass analogy where, given some weighting or "mass", individual grid nodes are repositioned to the center of mass of the surrounding nodes. McRae [16] and Laflin [17] successfully applied Eiseman's center of mass adaptation scheme to steady and unsteady supersonic flows.

While the mechanism for grid movement is an important aspect of grid adaptation, the selection of an appropriate adaptation driver is crucial to the overall reduction of numerical error. With the goal of more accurately resolving discontinuous flow features, much of the early grid adaptation research utilized solution features such as pressure or density gradients

to drive the adaptive process [7, 13, 18, 19, 20]. Although feature-based adaptation can provide increased resolution to shock waves and other discontinuities, this adaptation driver assumes that error is localized on the flow feature and does not account for how error is transported and diffused throughout the computational domain. Warren et al.[21] and Ainsworth and Oden [22] have found that feature-based adaptation can actually increase the error present in a solution and even cause solution features to set up in the wrong physical location. Zhang et al. [23], Roy [24], and Gu and Shih [25] have shown that a more appropriate adaptation driver is truncation error since truncation error acts as the local source for discretization error in numerical solutions. To this end, Choudhary and Roy [26, 27] extended the work of McRae [16] and Laflin [17] within the framework of truncation error based adaptation and were able to successfully drive adaptation and reduce spatial discretization error for 2-D square domains.

More recently, adjoint methods have emerged as the standard for adaptation aimed at reducing functional discretization error. Venditti and Darmofal [28, 29] pioneered this area of research in CFD by successfully reducing functional discretization error for 1-D and 2-D inviscid flows. While adjoint methods have gained much popularity for functional-based adaptation, Park et al. [30] and Fidkowski and Roe [31] have shown that adaptation based on one functional can actually increase the error in another functional. In this case, multiple adjoint problems must be solved in order to improve all functionals of interest. This deficiency on the part of adjoint methods led Fidkowski and Roe [31] to develop an entropy adjoint approach to functional-based adaptation which can provide accurate functional estimates without having to solve multiple adjoint problems. In a novel work, Derlaga et al. [32] and Derlaga [33] used truncation error based adaptation for improving solution functionals and found that truncation error based adaptation performs on par with or better than adjoint-based adaptation. Although much advancement has been made in the area of grid adaptation, a great deal of work remains to improve the overall efficiency and performance of adaptation schemes.

1.2 Contributions

Multiple r-adaptation techniques are applied within the framework of truncation error based adaptation with the goal of spatial discretization error reduction. These adaptation schemes are directly integrated into an existing flow solver to create an efficient, compact flow solver/adaptation package. A method for point movement along curved boundaries and across geometrically significant regions is presented. Substantial discretization error reductions of up to an order of magnitude are achieved for a variety of subsonic and supersonic inviscid flow problems.

1.3 Outline

The first chapter provides the reader with background information regarding grid adaptation, adaptation drivers, and previous work in this area.

The second chapter reviews the effectiveness of several r-adaptation schemes for reducing discretization error in numerical solutions.

The third chapter offers discussion of the work, draws conclusions, and provides areas for future improvement.

1.4 Attribution

As this thesis is written in a manuscript format, Chapter 2 of this work has multiple authors. To outline the contributions of each author, consider the following:

- William C. Tyson (First Author): The first author served as the main contributor and primary author of the work. The first author implemented the Anderson, Brackbill and Saltzman, and Deformation adaptation schemes in an existing structured adaptation module. This module along with the center of mass adaptation scheme were updated by the first author and directly incorporated into an existing flow solver. The first author also implemented novel boundary conditions for adaptation along curved boundaries as well as mechanisms for weight function and flow solution interpolation between grids.
- Joseph M. Derlaga (Second Author): The second author wrote the majority of the flow solver and provided helpful comments when drafting the manuscript for the work.
- Christopher J. Roy (Third Author): The third author provided valuable guidance and insight throughout the work and also provided helpful comments when drafting the manuscript.
- Aniruddha Choudhary (Fourth Author): The fourth author wrote the first version of the adaptation module and the initial code for the center of mass adaptation scheme.

1.5 Bibliography

[1] Roy, C., "Review of Discretization Error Estimators in Scientific Computing," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Jan 2010.

[2] Thompson, J. F., Thames, F. C., and Mastin, C., "Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies," *Journal of Computational Physics*, Vol. 15, No. 3, Jul 1974, pp. 299 – 319.

[3] Steger, J. L. and Chaussee, D. S., "Generation of Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations," *SIAM J. Sci. and Stat. Comput.*, Vol. 1, No. 4, Dec 1980, pp. 431 – 437.

[4] Thompson, J., Warsi, Z., and Mastin, C., *Numerical Grid Generation: Foundations and Applications*, Elsevier Science Publishing Co., Inc., 1997.

[5] Roache, P. J., Salari, K., and Steinberg, S., "Hybrid adaptive poisson grid generation and grid smoothness," *Communications in Applied Numerical Methods*, Vol. 7, No. 5, Jul 1991, pp. 345 – 354.

[6] Kania, L., "Elliptic adaptive grid generation and area equidistribution," *International Journal for Numerical Methods in Fluids*, Vol. 30, No. 5, Jul 1999, pp. 481 – 491.

[7] Anderson, D. A., "Equidistribution schemes, poisson generators, and adaptive grids," *Applied Mathematics and Computation*, Vol. 24, No. 3, Dec 1987, pp. 211 – 227.

[8] Anderson, D., "An Adaptive grid scheme controlling cell area/volume," *25th AIAA Aerospace Sciences Meeting*, Mar 1987.

[9] Anderson, D. A., "Development of Adaptive Grid Schemes Based on Poisson Grid Generators," Tech. rep., University of Texas at Arlington, 1989.

[10] Anderson, D. A., "Grid cell volume control with an adaptive grid generator," *Applied Mathematics and Computation*, Vol. 35, No. 3, Feb 1990, pp. 209 – 217.

[11] Winslow, A., "Numerical Solution of the Quasi-Linear Poisson Equation," *Journal of Computational Physics*, Vol. 1, 1966, pp. pp. 149 – 172.

[12] Brackbill, J. and Saltzman, J., "Adaptive zoning for singular problems in two dimensions," *Journal of Computational Physics*, Vol. 46, No. 3, Jun 1982, pp. 342 – 368.

[13] Gnoffo, P., "A vectorized, finite-volume, adaptive grid algorithm applied to planetary entry problems," *Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference*, , No. 82-1018, June 1982.

[14] Nakahashi, K. and Deiwert, G. S., "Self-adaptive-grid method with application to airfoil flow," *AIAA Journal*, Vol. 25, No. 4, Apr 1987, pp. 513 – 520.

[15] Eiseman, P. R., "Adaptive grid generation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, No. 1-3, Oct 1987, pp. 321 – 376.

[16] McRae, D., "r-Refinement grid adaptation algorithms and issues," *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, No. 4, Sep 2000, pp. 1161 – 1182.

[17] Laflin, K., *Solver-Independent r-Refinement Adaptation for Dynamic Numerical Simulations*, Ph.D. thesis, North Carolina State University, 1997.

[18] Deiwert, G. S., Andrews, A. E., and Nakahashi, K., "Theoretical analysis of aircraft afterbody flow," *Journal of Spacecraft and Rockets*, Vol. 24, No. 6, Nov 1987, pp. 496 – 503.

[19] Liao, G., Lei, Z., and de la Pena, G., "Adaptive grids for resolution enhancement," *Shock Waves*, Vol. 12, No. 2, Aug 2002, pp. 153 – 156.

[20] Nakahashi, K. and Deiwert, G., "A Practical Adaptive-Grid Method for Complex Fluid-Flow Problems," Technical Memo NASA-TM-85989, NASA Ames Research Center, June 1984.

[21] Warren, G., Anderson, W., Thomas, J., and Krist, S., "Grid convergence for adaptive methods," *10th Computational Fluid Dynamics Conference*, Jun 1991.

[22] Ainsworth, M. and Oden, J., *A Posteriori Error Estimation in Finite Element Analysis*, Wiley Interscience, New York, 2000.

[23] Zhang, X., Trepanier, J.-Y., and Camarero, R., "A posteriori error estimation for finite-volume solutions of hyperbolic conservation laws," *Computer Methods in Applied Mechanics and Engineering*, Vol. 185, No. 1, Apr 2000, pp. 1 – 19.

[24] Roy, C., "Strategies for Driving Mesh Adaptation in CFD (Invited)," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Jan 2009.

[25] Gu, X. and Shih, T., "Differentiating between source and location of error for solution-adaptive mesh refinement," *15th AIAA Computational Fluid Dynamics Conference*, Jun 2001.

[26] Choudhary, A. and Roy, C., "Efficient Residual-Based Mesh Adaptation for 1D and 2D CFD Applications," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Jan 2011.

[27] Choudhary, A. and Roy, C. J., "Structured Mesh r-Refinement using Truncation Error Equidistribution for 1D and 2D Euler Problems," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

[28] Venditti, D. A. and Darmofal, D. L., "Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow," *Journal of Computational Physics*, Vol. 164, No. 1, Oct 2000, pp. 204 – 227.

[29] Venditti, D. A. and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows," *Journal of Computational Physics*, Vol. 176, No. 1, Feb 2002, pp. 40 – 69.

[30] Park, M., Lee-Rausch, B., and Rumsey, C., "FUN3D and CFL3D Computations for the First High Lift Prediction Workshop," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Jan 2011.

[31] Kidkowski, K. and Roe, P., "Entropy-based Mesh Refinement, I: The Entropy Adjoint Approach," *19th AIAA Computational Fluid Dynamics Conference, San Antonio, Texas*, June 22-25, 2009.

[32] Derlaga, J. M., Phillips, T., Roy, C. J., and Borggaard, J., "Adjoint and Truncation Error Based Adaptation for Finite Volume Schemes with Error Estimates," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[33] Derlaga, J. M., *Application of Improved Truncation Error Estimation Techniques to Adjoint Based Error Estimation and Grid Adaptation*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, July 2015.

Chapter 2

Comparison of r-Adaptation Techniques for 2-D CFD Applications

William C. Tyson,[a] Joseph M. Derlaga,[a] Christopher J. Roy,[a]
Aniruddha Choudhary[b]

[a]Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, 24061
[b]National Energy Technology Laboratory, Morgantown, WV 26507

Abstract

Discretization error is often the primary source of error in numerical simulations. Discretization error is introduced into the solution by the required discretization of the continuous equations which govern a particular model. Adaptation can reduce the amount of error present in a solution by increasing the grid resolution or by including higher order terms in the discretization. r-Adaptation schemes provide increased grid resolution to regions of the domain where it is most needed by relocating nodes from other parts of the domain. The distinguishing factor between all r-adaptation schemes is the method used for point relocation. This paper examines the effectiveness of several 2-D r-adaptation schemes for reducing discretization error in numerical solutions. The adaptation schemes examined in this study include an adaptive Poisson grid generator, a variational grid generator, a center of mass based scheme, and a scheme based on deforming maps. These methods are applied to supersonic flow around a 12° downward turn, supersonic flow around a 10° compression turn, supersonic flow over a diamond-shaped airfoil, and subsonic flow over a Karman-Trefftz airfoil. For each case, discretization error is computed using known exact solutions. Discretization error reductions of up to eleven times are achieved on adapted grids relative to the discretization error present on a uniform grid of the same size. Due to the reduced order

of convergence for these test cases resulting from the presence of discontinuities, tremendous cost savings can be achieved over uniform grid refinement even for modest reductions in discretization error.

2.1 Introduction

Often in CFD applications, it is necessary to not only quantify but also reduce numerical errors. Numerical errors are typically categorized in the following manner: round-off error, iterative error, and discretization error. With modern computing, round-off and iterative error are generally not the leading contributors to the overall error in a solution. Discretization error, defined as the difference between the exact solution to the discrete equations and the exact solution to the continuous governing equations, is often the primary source of numerical error. Discretization error may be reduced through uniform grid refinement but at the cost of increased computation time. When large problems can sometimes take weeks to run, uniform grid refinement is not a viable option to increase solution accuracy. Grid adaptation, on the other hand, provides an avenue to increase solution accuracy without the exponential increase in computational cost associated with uniform grid refinement. When conducting grid adaptation, typically two design choices must be made: the mechanism for altering the grid and the indicator which drives the adaptive process. Historically, two distinct adaptation mechanisms for altering the grid have been studied: h-adaptation schemes and r-adaptation schemes. h-Adaptation schemes increase grid resolution through local cell subdivision rather than uniformly across the entire mesh. r-Adaptation schemes attempt to increase solution accuracy by locally increasing or decreasing the grid resolution for a fixed number of nodes via mesh movement. For this study, r-adaptation techniques are examined for structured grids.

Within the realm of structured r-adaptation, many philosophies exist regarding the appropriate driver for grid adaptation. The most common techniques are gradient, curvature, discretization error (DE), and truncation error (TE) based adaptation schemes [1]. Gradient-based adaptation drives the relocation procedure using the gradient of one or more flowfield quantities such as pressure or density. These types of methods are attractive for resolving shocks and other discontinuous flow features. Similarly, curvature-based adaptation drives node relocation according to the second derivative of one or more flowfield variables. Gradient-based and curvature-based schemes fall under what is typically referred to as feature-based adaptation. Alternatively, TE-based and DE-based based techniques relocate points into regions of the domain where error is high in an attempt to locally resolve the solution with greater accuracy.

Much research has been conducted to determine which adaptation technique is most appropriate for CFD applications. While feature-based adaptation schemes provide higher resolution of discontinuous flow features, these methods are not guaranteed to reduce numerical errors and can even introduce more error into the solution [2, 3]. Since reducing

discretization error is the ultimate goal of almost every adaptation scheme, it would seem reasonable that discretization error is the ideal driver for mesh adaptation. Zhang et al. [4], Roy [1], and Gu and Shih [5] have shown that this is not the case. Adaptation based on discretization error does not account for how discretization error is produced. On the other hand, truncation error, which is formally defined as the difference between the discrete governing equations and the continuous governing equations, is oftentimes a more appropriate adaptation driver since discretization error is introduced locally into the solution through truncation error. Therefore, by locally reducing truncation error, the local production of discretization error should also be reduced. For r-adaptation schemes, truncation error reduction is achieved by relocating nodes into regions where truncation error is high in an attempt to equidistribute it across the domain.

When conducting a CFD simulation, structured grids are often preferred over unstructured grids due to the logical ordering of information and because of their favorable grid quality. Some argue that it is desirable to maintain some amount of favorable grid quality, such as smoothness and orthogonality, in computational meshes especially in viscous boundary layers. With this in mind, adaptation schemes have been developed whose primary objective is to maintain and even optimize grid quality while still improving solution accuracy [6]. However, it has been found that global smoothness and orthogonality are not always necessary to accurately resolve flow behavior [7]. Therefore, it should be noted that, even for structured grids, grid quality alone is not a sufficient metric and must be simultaneously considered with the solution of the given problem.

Often, the distinguishing factor between r-adaptation schemes is how nodes are repositioned within the domain. Node relocation may be performed in a variety of ways. Baker [8], McRae[7], and Hawken et al. [9] provide extensive reviews of the many techniques used for node relocation. Most of these methods are centered around the fundamental principle of equidistributing a weight function across the domain where the weight function is typically some measure of the local truncation error. With this technique, nodes are pulled to regions of the domain where the weight function is high. Variations of these methods also include grid quality and other metrics to drive the adaptation procedure. In order to achieve the best discretization error reduction possible, it is important not only to understand how these methods work but also to be able to quantify their performance. The aim of this paper is to review and compare several well known 2-D r-adaptation schemes. The performance of each scheme is evaluated based upon the amount of discretization error reduction achieved. The benefit of using r-adaptation schemes is also illustrated in multiple grid studies which compare error levels on uniform and adapted grids. The adaptation schemes examined in this work include an adaptive Poisson grid generator, a variational grid generator, a center of mass based scheme, and a scheme based on deforming maps. These techniques are applied to supersonic, inviscid flow around a 12° downward turn, around a 10° compression turn, over a diamond-shaped airfoil, and to subsonic, inviscid flow over a Karman-Trefftz airfoil.

2.2 Adaptation Driver

2.2.1 Generalized Truncation Error Expression

When performing adaptation, it is imperative that an appropriate driver for adaptation be selected. Historically, solution features such as pressure gradients across a shock wave have been used. While feature-based adaptation can provide sharp resolution of discontinuities, this type of adaptation is not guaranteed to reduce numerical errors and can even introduce more error into the solution [2, 3] because it does not account for how discretization error is generated, transported, and dissipated. Rather, feature-based adaptation assumes that error is localized at a feature which may not necessarily be the case. This heuristic approach to adaptation can lead to under-resolution in areas of the domain which contribute most to the discretization error and over-resolution in areas which contribute little. The next logical candidate to drive adaptation might be to adapt on discretization error. But, not only can discretization error be difficult to quantify, but Roy [1] has shown that greater reductions in discretization error may be achieved if truncation error is used to flag regions of the domain for adaptation rather than discretization error.

To illustrate why truncation error may be viewed as an appropriate driver for adaptation, the continuous governing equations, $L(\cdot)$, and a consistent discretized form of the equations, $L_h(\cdot)$, can first be related using the Generalized Truncation Error Expression [10], given by

$$L_h(I^h u) = I^h L(u) + \tau_h(u) \tag{2.1}$$

where u is any continuous function and $\tau_h(\cdot)$ represents the truncation error. Based on Eq. 2.1, the truncation error may be viewed as higher order terms which are truncated during the discretization process of the continuous governing equations, $L(\cdot)$. For model problems, the truncation error can be shown to be a function of continuous solution derivatives and cell size [11]. But, while operating on a continuous solution space, evaluating the truncation error results in a discrete space [12, 13]. In order for Eq. 2.1 to be consistent, one must be able to represent a continuous function in a discrete space and vice versa. This is achieved via a restriction or prolongation operator, I_a^b, which allows for the transition between a continuous space and a discrete space; the subscript, a, denotes the starting space and the superscript, b, denotes the resultant space. In Eq. 2.1, a subscript or superscript h denotes a discrete space on a mesh with a characteristic size, h, and an empty subscript or superscript represents a continuous space. It is important to note that there is no restriction on the form of the continuous and discrete operators, $L(\cdot)$ and $L_h(\cdot)$; the governing equations can be in strong form (ODEs and PDEs) or weak form (integral equations).

2.2.2 Error Transport Equations

Now, having related the continuous governing equations, $L(\cdot)$, to the discrete governing equations, $L_h(\cdot)$, through the truncation error, $\tau_h(\cdot)$, the truncation error can now be related to the discretization error. First, consider the following definitions. For the exact solution to the continuous governing equations, $\tilde{u}$, and the exact solution to the discrete governing equations, u_h, the following holds

$$L(\tilde{u}) = 0 \tag{2.2a}$$

$$L_h(u_h) = 0 \ . \tag{2.2b}$$

Plugging the exact solution to the discrete equations, u_h, into Eq. 2.1 and noting that the left hand side is identically zero by Eq. 2.2b, Eq. 2.1 may be simplified to the following

$$0 = I^h L(I_h u_h) + \tau_h(I_h u_h) \tag{2.3}$$

where I_h prolongs the exact discrete solution to a continuous space. Next, the continuous governing equations, Eq. 2.2a, can be subtracted from both sides of Eq. 2.3 to form

$$0 = I^h L(I_h u_h) - I^h L(\tilde{u}) + \tau_h(I_h u_h) \ . \tag{2.4}$$

Finally, by defining the discretization error as the difference between the exact solution to the discrete equations and the exact solution to the continuous equations, $I_h \varepsilon_h = I_h u_h - \tilde{u}$, and by requiring the the governing equations be linear or linearized [13, 10], the continuous operators in Eq. 2.4 can be combined and the definition of discretization error may be inserted to form the continuous form of the error transport equations

$$I^h L(I_h \varepsilon_h) = -\tau_h(I_h u_h) \ . \tag{2.5}$$

An equivalent discrete form of Eq.2.5 can also be derived and are given by

$$L_h(\varepsilon_h) = -\tau_h(\tilde{u}) \ . \tag{2.6}$$

The error transport equations provide a great deal of information regarding how discretization error behaves and how it is related to the truncation error. Eq. 2.5 and Eq. 2.6 demonstrate that truncation error acts as the local source for discretization error in numerical solutions and that discretization error is convected and diffused in the same manner as the solution. Since truncation error acts as the source of discretization error, it can be viewed as an appropriate driver for adaptation. By targeting regions of the domain for adaptation where truncation error is high, the amount of discretization error introduced into the solution can be reduced.

2.3 Adaptation Schemes

2.3.1 Adaptive Poisson Grid Generation

One of the most common methods of grid generation in 2-D is elliptic grid generation. With this method of grid generation, the boundary nodes of the mesh are specified and used to solve for the interior point distribution by solving an elliptic PDE. Thompson et al. [14] derive the grid generation equations by minimizing the variation in the following functional

$$I = \iint \left[(\nabla \xi)^2 + (\nabla \eta)^2 \right] \, dA \tag{2.7}$$

where $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$ represent the coordinates of the grid in a logically Cartesian computational space and the integral quantity, I, is a measure of global smoothness. The Euler-Lagrange equations for this system are given by the following

$$\nabla^2 \xi = 0 \tag{2.8a}$$
$$\nabla^2 \eta = 0 \ . \tag{2.8b}$$

Eq. 2.8a and Eq. 2.8b are simply Laplace equations for the node locations in computational space. Solving these equations yields the smoothest mesh possible for a given boundary point distribution. While global smoothness is often an important characteristic of a mesh, this system lacks direct control of the interior points. Historically, interior point control is attained with this type of system by the introduction of inhomogeneous source terms. This yields Poisson equations for the computational coordinates

$$\nabla^2 \xi = P \tag{2.9a}$$
$$\nabla^2 \eta = Q \tag{2.9b}$$

where P and Q are control functions. One can now see that with an appropriate selection of the control functions, P and Q, this grid generation system may become adaptive. This type of adaptation has been popular due to its ease of implementation. Minimal changes have to be made to convert a standard grid generator into an adaptive grid generator. However, much care must be taken when determining the form of P and Q because in this system a one-to-one mapping is no longer guaranteed and grid crossover can easily become a problem.

Several methods have been proposed which seek to create an adaptive grid generator from Poisson's equation. The difficulty with deriving such a generator is determining the exact form of the control functions which will produce the desired adaptation. In this study,

the control functions proposed by Anderson [15] are used for mesh adaptation. To see how Eq. 2.9a and Eq. 2.9b can be made adaptive, consider the following form of the grid generation equations proposed by Winslow [16]

$$\nabla \cdot (D\nabla\xi) = 0 \tag{2.10a}$$
$$\nabla \cdot (D\nabla\eta) = 0 \tag{2.10b}$$

where D is a diffusion coefficient which controls the flux of the computational coordinate into and out of a given volume [15]. Anderson [15] points out that upon expansion it can be seen that this is a system of Poisson equations

$$\nabla^2\xi = -\nabla\xi \, \frac{\nabla D}{D} \tag{2.11a}$$
$$\nabla^2\eta = -\nabla\eta \, \frac{\nabla D}{D} \tag{2.11b}$$

where the control functions, P and Q, take the following form

$$P = -\nabla\xi \, \frac{\nabla D}{D} \tag{2.12}$$
$$Q = -\nabla\eta \, \frac{\nabla D}{D} \, . \tag{2.13}$$

This system is made adaptive by applying the equidistribution principle to the computation of the diffusion coefficient such that

$$DW = constant \tag{2.14}$$

where $W = W(x, y)$ is a weight function or indicator function that flags which regions of the domain require adaptation. The selection of the form of this function is crucial to how the adaptation scheme behaves and ultimately the amount of discretization error improvement that can be obtained.

In most situations, it is not possible to solve Eq. 2.11a and Eq. 2.11b directly for the mesh in physical space. Therefore, it is necessary to invert the equations and pose them in a logically Cartesian computational space. In computational space, the governing equations become

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = (\alpha x_\xi - \beta x_\eta)\frac{D_\xi}{D} - (\beta x_\xi - \gamma x_\eta)\frac{D_\eta}{D} \tag{2.15a}$$
$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = (\alpha y_\xi - \beta y_\eta)\frac{D_\xi}{D} - (\beta y_\xi - \gamma y_\eta)\frac{D_\eta}{D} \tag{2.15b}$$

where the nonlinear coefficients, α, β, and γ are

$$\alpha = x_\eta^2 + y_\eta^2 \tag{2.16}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \tag{2.17}$$

$$\gamma = x_\xi^2 + y_\xi^2 \ . \tag{2.18}$$

Eq. 2.15a and Eq. 2.15b are discretized using second order accurate finite differences and solved using a symmetric Gauss-Seidel relaxation method to determine the node locations.

2.3.2 Variational Adaptation

Often when equidistribution of a weight function is the sole principle driving an adaptation scheme, the resulting mesh can be highly skewed even to the point where some flow solvers will not run on them. It can also be shown that skewness and non-smooth transformations can be the leading terms in the truncation error [1]. Therefore, in most cases, it is desirable to retain some amount of smoothness and orthogonality in computational meshes. This can be achieved using variational methods. The goal of variational grid generators is to determine the mesh which optimizes one or more measures of grid quality across the domain.

The variational scheme examined in this paper is that of Brackbill and Saltzman [6]. Their adaptation scheme aims to optimize the following grid quality metrics: smoothness, grid line orthogonality, and weighted cell volume variation. The governing equations for this system are determined in a similar manner to Eq. 2.8a and Eq. 2.8b by minimizing the variation of an integral quantity of interest across the domain. Integral measures of smoothness, orthogonality, and weighted cell volume variation are given by the following expressions

$$I_s = \iint \left[(\nabla\xi)^2 + (\nabla\eta)^2 \right] \, dA \tag{2.19}$$

$$I_o = \iint (\nabla\xi \cdot \nabla\eta)^2 \, J^3 \, dA \tag{2.20}$$

$$I_v = \iint W J \, dA \tag{2.21}$$

where $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$ are the grid locations in a logically Cartesian computational space, J is the Jacobian of the mapping between computational space and physical space, I_s is a measure of the global smoothness of the mapping, I_o is a volume weighted measure of the orthogonality of the mapping, I_v is the weighted variation of cell volume, and W is a weight function. Through minimizing the variation of each of these integrals, the respective grid

quality metric for that integral may be optimized across the mesh. For example, minimizing the variation of Eq. 2.19 will produce the mesh with the smoothest mapping. In their work, Brackbill and Saltzman investigated the optimization of a combination of these grid quality metrics [6]. This was accomplished by linearly combining Eq. 2.19 - Eq. 2.21 in the following manner

$$I = I_s + \lambda_o I_o + \lambda_v I_v \tag{2.22}$$

where λ_o and λ_v are positive, nonzero coefficients. The combined grid quality measure, I, takes the form of Eq. 2.22 because it is necessary to combine the integral measures of grid quality in this manner to obtain a unique solution [6]. Optimizing solely on orthogonality or weighted cell volume variation will not yield a unique solution; some amount of smoothness is required.

In order to obtain a grid which optimizes the combination of smoothness, orthogonality, and weighted cell volume variation defined by Eq. 2.22, the Euler-Lagrange equations which minimize the variation in the combined grid quality measure, I, must be determined. It can be shown that the Euler-Lagrange equations for the combined grid quality measure, I, can be obtained by linearly combining the Euler-Lagrange equations for the individual grid quality measures in the same manner as Eq. 2.22. Once the Euler-Lagrange system is determined, the equations must be inverted and posed in a logically Cartesian space in order to solve for the grid node locations.

The Euler-Lagrange equations for the optimization of global smoothness of the mapping are given by the following

$$b_{s1}x_{\xi\xi} + b_{s2}x_{\xi\eta} + b_{s3}x_{\eta\eta} + a_{s1}y_{\xi\xi} + a_{s2}y_{\xi\eta} + a_{s3}y_{\eta\eta} = 0 \tag{2.23a}$$
$$a_{s1}x_{\xi\xi} + a_{s2}x_{\xi\eta} + a_{s3}x_{\eta\eta} + c_{s1}y_{\xi\xi} + c_{s2}y_{\xi\eta} + c_{s3}y_{\eta\eta} = 0 \tag{2.23b}$$

where the coefficients $a_{s(\cdot)}$, $b_{s(\cdot)}$, and $c_{s(\cdot)}$ are given in Appendix A.1.

The Euler-Lagrange equations for the global orthogonality of the mapping are

$$b_{o1}x_{\xi\xi} + b_{o2}x_{\xi\eta} + b_{o3}x_{\eta\eta} + a_{o1}y_{\xi\xi} + a_{o2}y_{\xi\eta} + a_{o3}y_{\eta\eta} = 0 \tag{2.24a}$$
$$a_{o1}x_{\xi\xi} + a_{o2}x_{\xi\eta} + a_{o3}x_{\eta\eta} + c_{o1}y_{\xi\xi} + c_{o2}y_{\xi\eta} + c_{o3}y_{\eta\eta} = 0 \tag{2.24b}$$

where the coefficients $a_{o(\cdot)}$, $b_{o(\cdot)}$, and $c_{o(\cdot)}$ are given in Appendix A.2.

The Euler-Lagrange equations for weighted cell volume variation are given by the following

$$b_{v1}x_{\xi\xi} + b_{v2}x_{\xi\eta} + b_{v3}x_{\eta\eta} + a_{v1}y_{\xi\xi} + a_{v2}y_{\xi\eta} + a_{v3}y_{\eta\eta} = -\frac{J^2}{2W}\frac{\partial W}{\partial x} \tag{2.25a}$$

$$a_{v1}x_{\xi\xi} + a_{v2}x_{\xi\eta} + a_{v3}x_{\eta\eta} + c_{v1}y_{\xi\xi} + c_{v2}y_{\xi\eta} + c_{v3}y_{\eta\eta} = -\frac{J^2}{2W}\frac{\partial W}{\partial y} \tag{2.25b}$$

where the coefficients $a_{v(\cdot)}$, $b_{v(\cdot)}$, and $c_{v(\cdot)}$ are given in Appendix A.3. It is important to note that the weight function, $W(x, y)$, defined across the domain governs how the volume of each cell should vary. Similar to adaptive Poisson grid generation, the selection of the form of $W(x, y)$ determines how the adaptation will perform. The Jacobian of the transformation, J, in Eq. 2.25a and Eq. 2.25b is given by

$$J = x_\xi y_\eta - x_\eta y_\xi \ . \tag{2.26}$$

The Euler-Lagrange equations for the system are determined by combining Eq. 2.23a - 2.25b where the coefficients for each individual set of equations are combined in the following manner

$$a_{(\cdot)} = a_{s(\cdot)} + \lambda_o a_{o(\cdot)} + \lambda_v a_{v(\cdot)} \tag{2.27a}$$

$$b_{(\cdot)} = b_{s(\cdot)} + \lambda_o b_{o(\cdot)} + \lambda_v b_{v(\cdot)} \tag{2.27b}$$

$$c_{(\cdot)} = c_{s(\cdot)} + \lambda_o c_{o(\cdot)} + \lambda_v c_{v(\cdot)} \ . \tag{2.27c}$$

Each term in Eq. 2.23a - 2.25b is discretized using second order accurate finite differences. The updated node locations are solved using a point Jacobi relaxation technique.

Although the optimization of grid quality seems attractive, several authors [17, 18] have pointed out a deficiency with this method in that the functionals for smoothness, orthogonality, and weighted cell volume variation are not dimensionally consistent. Therefore, from problem to problem, the coefficients, λ_o and λ_v, do not scale and must be continuously updated. Also, the choices for λ_o and λ_v appear to be somewhat arbitrary making it difficult to determine appropriate values for a given application. For this study, with weighted cell volume variation as the highest priority, the coefficients are selected as $\lambda_v = 10$ and $\lambda_o = 1$. Lastly, Brackbill and Saltzman [6] found that the combination of grid quality metrics does not produce a mesh which simultaneously optimizes each individual metric. Therefore, the solution of Eq. 2.23a - 2.25b, while optimizing a given combination of grid quality metrics, can suffer from a deterioration of one or more of the grid quality measures.

2.3.3 Center of Mass

The formulation of the center of mass approach to mesh movement used in this study is based on the work of Laflin [19] and McRae [7]. This is a physically based scheme which relocates nodes to the center of mass of what is referred to as the local cell cluster. An example of a local cell cluster in physical space may be found in Fig. 2.1 where node V is the node which is to be relocated. In order to prevent mesh crossover, node relocation is performed in a time-dependent parametric domain given in Fig. 2.2 rather than in physical space. The time-dependency of the parametric domain refers to the state of the domain at the start of each adaptation cycle. In the work of Benson and McRae [20], the parametric domain is time-independent in that the mapping between it and the physical domain remains fixed. However, to prevent mesh crossover and to save computer memory associated with storing the initial grid, Laflin [19] reinitializes the parametric domain at the beginning of each adaptation cycle to a logically Cartesian grid. Once a particular node is repositioned in parametric space, its updated location in physical space is determined by mapping the local cell cluster back to physical space.

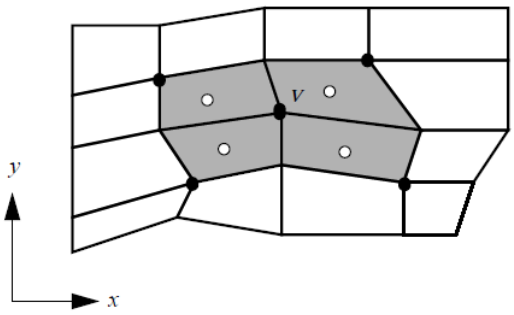


Figure 2.1: Local Cell Cluster: Physical Space (Reproduced from Ref. [19])

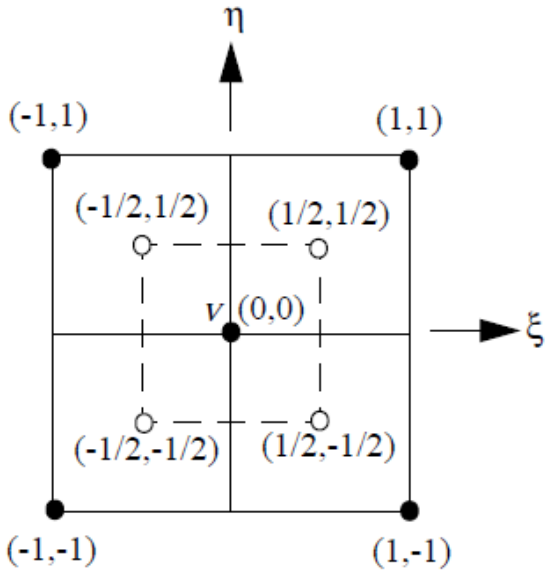


Figure 2.2: Local Cell Cluster: Parametric Space (Reproduced from Ref. [19])

To further illustrate the node relocation procedure, first consider the definition of the center of mass of a body, $\hat{r}_{CoM}$, given by [19]

$$\hat{r}_{CoM} = \frac{\int_{\Omega_v} \hat{m}(\hat{r})\hat{r}\, dV}{\int_{\Omega_v} \hat{m}(\hat{r})\, dV} \ . \tag{2.28}$$

Here, the domain of integration, Ω_v, is the volume of the local cell cluster in the parametric domain given in Fig. 2.2. Since each local cell cluster is composed of N_v number of cells, where in 2-D $N_v = 4$, the center of mass can be further defined as [19]

$$\hat{r}_{CoM} = \frac{\sum_{i=1}^{N_v} \int_{\Omega_i} \hat{m}(\hat{r})\hat{r}\, dV}{\sum_{i=1}^{N_v} \int_{\Omega_i} \hat{m}(\hat{r})\, dV} \tag{2.29}$$

where the domain of integration, Ω_i, is now the volume of each cell within the local cell cluster. Now, using the mean-value theorem, the following substitutions can be made [19]

$$\int_{\Omega_i} \hat{m}(\hat{r})\hat{r}\, dV = \hat{V}_i \bar{\hat{m}}_i \hat{r}_i \tag{2.30}$$

$$\int_{\Omega_i} \hat{m}(\hat{r})\, dV = \hat{V}_i \bar{\hat{m}}_i \tag{2.31}$$

where $\hat{V}_i$ is the volume of cell i in the parametric domain, $\bar{\hat{m}}_i$ is the average value of mass in cell i in the parametric domain, and $\hat{r}_i$ is the center of mass of cell i. Requiring that the mass in cell i be invariant between the parametric and physical domains, it can be shown that [19]

$$\hat{V}_i \bar{\hat{m}}_i = V_i \bar{m}_i \tag{2.32}$$

where V_i is the volume of cell i in the physical domain and $\bar{m}_i$ is the average value of mass in cell i in the physical domain. If the weight function which is used to drive adaptation is related to the product of mass times volume for cell i in the following manner, $W_i = V_i \bar{m}_i$, then the center of mass of the local cell cluster can be found using

$$\hat{r}_{CoM} = \frac{\sum_{i=1}^{N_v} W_i \hat{r}_i}{\sum_{i=1}^{N_v} W_i} \ . \tag{2.33}$$

Since the parametric domain is reinitialized at the beginning of each adaptation step, Eq. 2.33 can be simplified to the following expressions for the updated node locations in computational

space [19]

$$\xi_V^{new} = \frac{W_{(1/2,1/2)} + W_{(1/2,-1/2)} - W_{(-1/2,1/2)} - W_{(-1/2,-1/2)}}{W_{(1/2,1/2)} + W_{(1/2,-1/2)} + W_{(-1/2,1/2)} + W_{(-1/2,-1/2)}} \tag{2.34a}$$

$$\eta_V^{new} = \frac{W_{(1/2,1/2)} - W_{(1/2,-1/2)} + W_{(-1/2,1/2)} - W_{(-1/2,-1/2)}}{W_{(1/2,1/2)} + W_{(1/2,-1/2)} + W_{(-1/2,1/2)} + W_{(-1/2,-1/2)}} \tag{2.34b}$$

where $W_{(\cdot,\cdot)}$ is the value of the weight function located at the center of a given cell in the local cell cluster. Finally, in 2-D, the node locations in physical space may be updated in the following manner [19]

$$\Delta r_v = r_\xi \Delta\xi + r_\eta \Delta\eta + r_{\xi\eta} \Delta\xi \Delta\eta \tag{2.35}$$

where r_ξ, r_η, and $r_{\xi\eta}$ are metric derivatives and $\Delta\xi$ and $\Delta\eta$ are the incremental changes of the node location in the parametric domain.

2.3.4 Deformation Method

The deformation method for grid adaptation was primarily developed by Liao and Anderson [21] and is based upon the work of Moser [22] on volume elements of a Riemannian manifold. A concise explanation of the method can be found in [23]. First, consider a weight or density function in 2-D, $W(\xi, \eta, t)$. It is important to note that this definition of a weight function behaves opposite from the conventional definition of a weight function in that the function is large in regions where fewer points are needed. The weight function is used to deform the original grid through a family of transformations, $\phi(\xi, \eta, t)$, so that the weight function explicitly defines the Jacobian of the transformation, J, and therefore the variation of cell volume [23]

$$J(\phi) = W(\phi, t) \ . \tag{2.36}$$

If the weight or density function is some measure of solution error, then by the principle of error conservation, it can be shown that from the original grid to the adapted grid, the following relation holds for a given cell where the domain of integration is the cell in the adapted grid [23]

$$\frac{\partial}{\partial t} \int \frac{1}{W} \, dV = 0 \ . \tag{2.37}$$

Grisham et al. [23] go on to point out that if the quantity $\frac{1}{W}$ is thought of as a density, ρ, Eq. 2.37 is simply a statement of conservation of mass. In strong, conservation form, this can be represented by the following

$$\frac{\partial}{\partial t}\left(\frac{1}{W}\right) + \nabla \cdot \left(\frac{\vec{v}}{W}\right) = 0 \ . \tag{2.38}$$

Assuming the quantity $\frac{\vec{v}}{W}$ is irrotational, Eq. 2.38 may be rewritten as Poisson's equation in terms of a potential function, ε, subject to Neumann boundary conditions

$$\nabla^2 \varepsilon = -\frac{\partial}{\partial t}\left(\frac{1}{W}\right) \ . \tag{2.39}$$

The solution to Eq. 2.39 is obtained using second order finite differences. Node locations are updated via Euler-explicit time integration [23]

$$x^{n+1} = x^n + v_x \Delta t \tag{2.40a}$$
$$y^{n+1} = y^n + v_y \Delta t \tag{2.40b}$$

where the mesh velocities, v_x and v_y, are defined by

$$\vec{v} = W \nabla \varepsilon \ . \tag{2.41}$$

In this method, the time parameter, t, can explicitly represent time if it is desired for the mesh to move in conjunction with the flow solution in a time accurate manner. For steady-state problems, the time parameter, t, can be thought of as "pseudo-time" or simply a numerical tool to acquire the steady-state grid.

2.4 Adaptation Boundary Conditions

For this work, multiple boundary conditions are implemented to offer some freedom when selecting how the boundary points of the mesh should be allowed to move. The boundary conditions implemented are a Dirichlet boundary condition, a Neumann boundary condition, and a mathematically consistent boundary condition for each adaptation scheme.

2.4.1 Dirichlet Boundary Condition

The Dirichlet boundary condition is simply a fixed point boundary condition. As the adaptation scheme is solved on the interior of the mesh, the boundary points remain fixed. While this boundary condition requires minimal effort to implement, it restricts the amount of mesh movement that can be achieved. In order to attain the greatest amount of improvement in discretization error, the boundary points must be allowed to freely move along the boundary.

2.4.2 Neumann Boundary Condition

Typically, a Neumann boundary condition refers to prescribing the derivative of a given quantity at the grid boundary. With respect to grid adaptation, a Neumann boundary condition prescribes how opposite family grid lines intersect the boundary surface, for example, how grid lines of constant ξ intersect the minimum η grid line. This type of boundary condition can be especially useful when conducting adaptation for viscous flows where orthogonal grid lines at walls is desired.

For structured, r-adaptation, a Neumann boundary condition can either be implemented by forcing the interior grid lines to move to satisfy the derivative constraint while keeping the boundary points fixed or by allowing the boundary points to slide along the boundary surface. Requiring the interior points to move to satisfy the boundary condition can be difficult to enforce. Also, similar to the Dirichlet boundary condition, fixing the boundary points limits the amount of improvement which can be achieved by the adaptation scheme. Therefore, for grid adaptation targeted at discretization error improvement, it is better to allow points to move along the boundary to satisfy the boundary derivative constraint.

The Neumann boundary condition implemented in this work was developed by Zhang et al. [24]. This boundary condition moves points along the boundary curve so that grid lines intersect the boundary orthogonally. To move points along a given boundary curve, the first interior mesh line adjacent to the boundary is projected orthogonally across the boundary curve. This projected curve along with the boundary curve and the first interior mesh line are used to form an auxiliary mesh. An illustration of an auxiliary mesh can be seen in Fig. 2.3 where the solid dots represent nodes along the boundary curve and the first interior mesh line and the open dots represent the nodes along the projected curve. Eq. 2.8a and

Eq. 2.8b are solved on the auxiliary mesh subject to Dirichlet boundary conditions such that the boundary points move along the boundary surface and grid lines intersect the boundary orthogonally. It is important to note that since the computation of this boundary condition depends upon the position of the first interior mesh line of the grid, this boundary condition must be lagged behind the interior adaptation scheme.

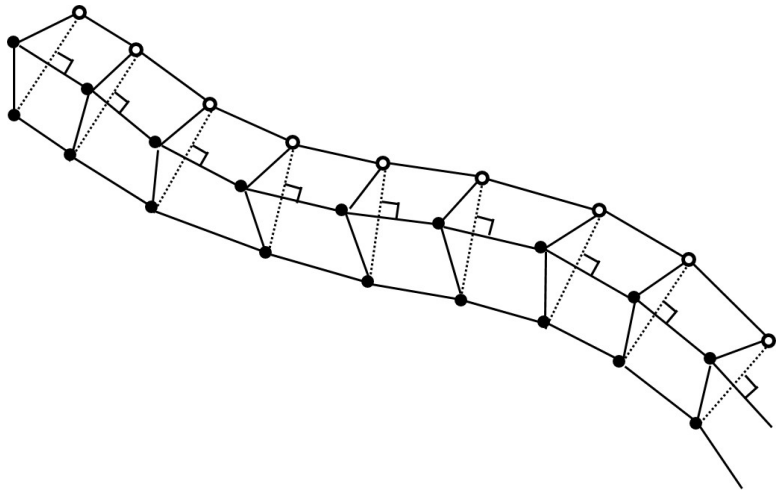


Figure 2.3: Neumann Boundary Condition: Auxiliary Mesh

2.4.3 Adaptation Scheme Consistent Boundary Conditions

For each adaptation scheme used in this work, there exists a mathematically consistent boundary condition. Ideally, adaptation should be performed with the boundary condition that is consistent with a particular scheme, but it is sometimes better to use a different boundary condition if the adaptation fails to converge or the grid crosses over. For this study, each test case is run with the adaptation scheme consistent boundary condition unless otherwise noted. In cases where this boundary condition hinders the performance of a given adaptation scheme, the boundary condition is changed to a Neumann boundary condition to improve results.

For Anderson's adaptation scheme [15] (adaptive Poisson grid generation) and Brackbill and Saltzman's adaptation scheme [6] (variational adaptation), the consistent boundary condition is deemed a Neumann boundary condition. For center of mass [19, 7], the interior scheme is solved on the boundary by reflecting the weight function in the first interior line of cells into ghost cells. Similar to center of mass, the interior scheme for the Deformation method [21, 23] is solved on the boundary such that boundary points could only move tangential to the boundary surface.

2.5 Geometry Preservation

2.5.1 Boundary Preservation

When allowing points to move along the boundary surface during the adaptation process, it is essential to preserve the geometric integrity of the body being studied. If care is not taken when relocating boundary points, the geometry of the body can degrade and the resulting grid will be rendered useless. In this work, to maintain the geometry, each boundary is parameterized as

$$x = x(s) \tag{2.42a}$$
$$y = y(s) \tag{2.42b}$$

where the parameterization variable, s, is taken as $s = 0$ at the start of a boundary curve and $s = 1$ at the end of a boundary curve. The parameterization variable, s, is computed at each point along a boundary in the following manner

$$s_k = \frac{\sum_{i=2}^{k} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{\sum_{i=2}^{N} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \tag{2.43}$$

where s_k is the parametric position of the k^{th} boundary point and N is the total number of points along a given boundary. It is important to note that the numerator of Eq. 2.43 represents the arc length up to the k^{th} point while the denominator represents the total arc length of the boundary curve. Once a boundary is parameterized, a piecewise cubic Hermite interpolating polynomial (PCHIP) [25] is fit to both x and y coordinates to define the functional form of Eq. 2.42a and Eq. 2.42b. During the adaptation process, the parameterization variable, s, is updated along each boundary and the corresponding functional form of Eq. 2.42a and Eq. 2.42b for that boundary from the initial, unadapted grid is used to project the boundary points back to the correct location in space in order to maintain the body geometry.

Now, depending upon the body being studied, only using this technique for geometry preservation can fail. Sometimes maintaining the body geometry requires that a boundary point always be located at a set physical location in space, such as the trailing edge of an airfoil or the point which defines a sharp corner. By evaluating a curve fit over the entire boundary, sharp corners can become rounded and, in the case of an airfoil, the chord length can change. To combat this issue, fixed points in space are defined along each boundary prior to adaptation. Eq. 2.42a and Eq. 2.42b are then defined in regions between the fixed points

of a given boundary such that

$$x = \begin{cases} x_1(s) & s_1 < s < s_2 \\ \vdots & \\ x_{f-1}(s) & s_{f-1} < s < s_f \end{cases} \tag{2.44a}$$

$$y = \begin{cases} y_1(s) & s_1 < s < s_2 \\ \vdots & \\ y_{f-1}(s) & s_{f-1} < s < s_f \end{cases} \tag{2.44b}$$

where f is the number of fixed points along a given boundary and $s_{(\cdot)}$ is the parametric position of a fixed point. It is important to note that the endpoints of the boundary curve are also considered fixed points. Similar to Eq. 2.42a and Eq. 2.42b, Eq. 2.44a and Eq. 2.44b are defined on the initial, unadapted grid. As adaptation progresses and points are allowed to freely move along the boundary curve, the closest boundary point to a fixed point is relocated to the prescribed physical space location of that fixed point. The parametric position, s, of each boundary point is updated and Eq. 2.44a and Eq. 2.44b are evaluated appropriately.

2.5.2 Interior Preservation

Depending upon the formulation of the weight function, the adaptation driver, and the node relocation scheme, the grid movement between each adaptation cycle can often be quite substantial even to the point where the grid crosses over. Since the grid is relatively constrained at the boundaries, grid crossover normally occurs on the interior of the domain. When grid crossover occurs, grid lines overlap creating cells with negative volume. In this case, the flow solver will no longer be able to run on the grid. Therefore, care must be taken to ensure that either grid crossover does not occur or that when it does measures are in place to fix the grid. In this study, to help prevent grid crossover from one iteration to the next, under-relaxation of the grid update is implemented

$$x^{k+1} = x^k + \omega\left(x^{k+1} - x^k\right) \tag{2.45a}$$

$$y^{k+1} = y^k + \omega\left(y^{k+1} - y^k\right) \tag{2.45b}$$

where x^{k+1} and y^{k+1} are the updated x and y coordinates of the grid, x^k and y^k are the x and y coordinates from the previous iteration, and ω is the under-relaxation factor which is less than one. For this study, an under-relaxation factor, ω, between 0.1 and 0.3 is used. For cases where grid crossover still occurs, the updated grid is reset to the grid at the beginning of the adaptation cycle and adaptation is stopped.

2.6 Weight Function Formulation

2.6.1 Truncation Error Based Weight Function

Since truncation error is selected to drive the adaptation, it is necessary to formulate a weight function in terms of truncation error such that nodes are relocated to regions of the domain where truncation error is high. To holistically adapt the mesh for the entire governing system of equations, the truncation error from each governing equation must be combined in a consistent manner which accounts for the orders of magnitude difference between each equation's truncation error. This is accomplished by defining the local weight function as the average of the absolute value of each equation's truncation error normalized by the L_2 norm of that equation's truncation error on the initial grid. For this study, the weight function at a given grid point takes the following form

$$W_{i,j} = \frac{1}{N} \sum_{k=1}^{N} \frac{|\tau_h(\tilde{u})^k_{i,j}|}{||\tau_h(\tilde{u})^k||_{2,initial}} \tag{2.46}$$

where N is the number of governing equations.

2.6.2 Weight Function Smoothing

When constructing the weight function in this manner, there is often some amount of high frequency noise present. To prevent the adaptation schemes from diverging or causing mesh crossover, it is necessary to apply a smoothing algorithm to the weight function prior to its application. Weight function smoothing is accomplished by applying the following elliptic smoother

$$W^{smooth}_{i,j} = \frac{W_{i-1,j} + W_{i+1,j} + 8W_{i,j} + W_{i,j+1} + W_{i,j-1}}{12} \ . \tag{2.47}$$

Although some weight function smoothing is required, it is also important that the weight function not be over-smoothed either. If the weight function is over-smoothed, information about the problem is lost, and the amount of discretization error improvement will likely be diminished. Therefore, some care must be taken when selecting the amount of weight function smoothing for a particular problem.

2.6.3 Weight Function Interpolation

When the weight function, $W = W(x, y)$, is computed at a particular grid node, it is stored such that its (i, j) location in an array corresponds to the (i, j) location in the grid where

the weight function was computed. As the adaptation scheme progresses and the grid nodes are relocated, the stored values of the weight function are no longer consistent with the node locations of the grid. In order to maintain an accurate representation of $W = W(x, y)$, the values of the weight function must be interpolated from one grid to another. In this work, a bilinear interpolation of the form

$$W(\xi, \eta) = a + b\xi + c\eta + d\xi\eta \tag{2.48}$$

is used to interpolate between grids. First, for a particular point in the new, adapted grid, the cell in the original grid in which the point is located must be determined. An example of this may be seen in Fig. 2.4 where the dashed lines represent a cell in the original grid and the point p is the point to be interpolated.

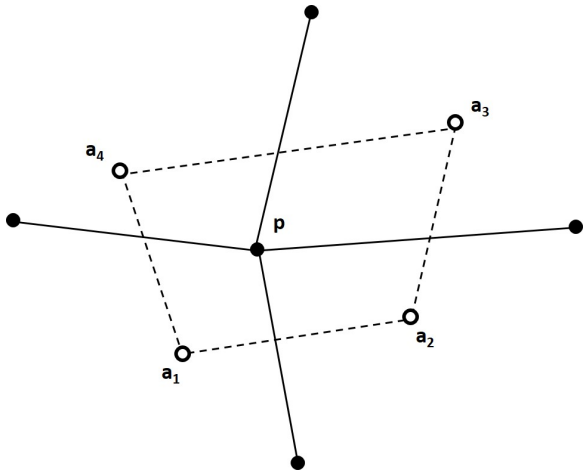


Figure 2.4: Bilinear Interpolation of Weight Function: Old Grid (Dashed), New Grid (Solid)

Next, the (x, y) locations of the points a_1 through a_4 are used to map the quadrilateral cell in the old grid to a unit square using the following bilinear mapping functions

$$x = b_1 + b_2\xi + b_3\eta + b_4\xi\eta \tag{2.49a}$$
$$y = c_1 + c_2\xi + c_3\eta + c_4\xi\eta \ . \tag{2.49b}$$

Once the mapping is determined, the (ξ, η) location of the point p can also be computed. Lastly, having computed the (ξ, η) location of point p and knowing the weight function values on the original grid at points a_1 through a_4, the weight function at point p may be interpolated using Eq. 2.48.

2.7 Adaptation Monitors

When conducting adaptation, it is important to monitor the convergence of the adaptation scheme to know when to stop adaptation. However, an appropriate adaptation monitor must be selected that is consistent with the adaptation scheme being used. For this investigation, three adaptation monitors are used to track the convergence of adaptation process. The first adaptation monitor is a truncation error based monitor given by

$$\frac{||\tau^k_{h_{i,j}}||_{2,adapted}}{||\tau^k_{h_{i,j}}||_{2,initial}} \tag{2.50}$$

which compares the L_2 norm of truncation error on the adapted grid to the L_2 norm of truncation error on the initial grid. The L_2 norm of truncation error is computed in 2-D using

$$L_p = \left[\frac{1}{A}\sum_{i=1}^{i_{max}}\sum_{j=1}^{j_{max}}|\phi_{i,j}|^p A_{i,j}\right]^{1/p} \tag{2.51}$$

where p is set to 2, $\phi_{i,j}$ represents the truncation error in a given cell, $A_{i,j}$ is the area of a given cell, and A is the total area of the domain.

The second adaptation monitor, taken from previous work by Choudhary [26], is weight function equidistribution error given by

$$WEE_{i,j} = \left|\frac{\sigma - |W_{i,j}|A_{i,j}}{\sigma}\right| \tag{2.52}$$

where σ is the equidistributed value of the quantity weight function times cell area across the entire computational domain.

The final adaptation monitor used is the L_2 norm of grid movement where the L_2 norm is computed using Eq. 2.51 without the area weighting and grid movement is defined as

$$\Delta s = \sqrt{(x^{k+1} - x^k)^2 + (y^{k+1} - y^k)^2}\ . \tag{2.53}$$

The superscript k in Eq. 2.53 denotes the grid at a given outer iteration of the adaptation scheme. An outer iteration of the adaptation scheme will be defined in the following section.

For all test cases in this study, a combination of these adaptation monitors is used to determine adaptive convergence. In general, the adaptation process is deemed converged if both the L_2 norm of weight function equidistribution error and the L_2 norm of grid movement are less than 0.1. But, in cases where the norms of truncation error converged before weight function equidistribution error, the adaptation process is stopped prematurely to prevent increased computational cost with diminishing returns.

2.8 Numerical Methods

2.8.1 Flow Solver: SENSEI

For this study, all simulations are conducted using SENSEI, a 3-D, structured, Euler/Navier-Stokes solver developed by Derlaga et al. [27]. The flowfield for each test case is determined by solving the Euler equations on 2-D structured grids. The Euler equations in weak, conservation form are given by

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{Q}\,d\Omega + \oint_{\partial\Omega}\left(\vec{F}_i - \vec{F}_v\right)\,dS = \int_{\Omega}\vec{S}\,d\Omega \tag{2.54}$$

where $\vec{Q}$ is the vector of conserved variables, $\vec{F}_i$ is the vector of inviscid fluxes, $\vec{F}_v$ is the vector of viscous fluxes, $\vec{S}$ is a generic source term, and Ω is the fixed control volume over which conservation of mass, momentum and energy must be satisfied. For the 2-D Euler equations, the vector of conserved variables, $\vec{Q}$, the fluxes, $\vec{F}_i$ and $\vec{F}_v$, and the source term, $\vec{S}$, are given by

$$\vec{Q} = \begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho e_t\end{bmatrix}, \quad \vec{F}_i = \begin{bmatrix}\rho V_n\\ \rho u V_n + p n_x\\ \rho v V_n + p n_y\\ \rho h_t V_n\end{bmatrix}, \quad \vec{F}_v = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \quad \vec{S} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{2.55}$$

where n_x and n_y are the components of the outward pointing normal, $\hat{n} = n_x\hat{i} + n_y\hat{j}$, on the surface of the control volume, $\partial\Omega$, and V_n is velocity normal to the surface given by

$$V_n = \vec{V}\cdot\hat{n}\ . \tag{2.56}$$

The system of equations is closed using the equation of state for a perfect gas such that the total energy, e_t, and the total enthalpy, h_t are given by

$$e_t = \frac{p}{\rho(\gamma-1)} + \frac{\vec{V}\cdot\vec{V}}{2}, \quad h_t = \frac{\gamma p}{\rho(\gamma-1)} + \frac{\vec{V}\cdot\vec{V}}{2} \tag{2.57}$$

where γ is the ratio of specific heats for a perfect gas which for air is $\gamma = 1.4$.

The Euler equations are discretized using a second order, cell-centered, finite volume discretization. The inviscid fluxes are computed using Roe's flux difference splitting scheme [28] and Van Leer's flux vector splitting scheme [29]. MUSCL extrapolation [30] is used to reconstruct the fluxes to the face in order to obtain second order spatial accuracy. The boundary

conditions are enforced weakly through the fluxes using a ghost cell implementation. Numerical solutions are marched in time to a steady-state using an implicit time stepping scheme [31] given by the following

$$\left[\frac{\Omega}{\Delta t}I + \frac{\partial \vec{R}}{\partial \vec{Q}}\right]^n \Delta \vec{Q}^n = -\vec{R}^n \tag{2.58}$$

where Ω is the volume of a given cell, Δt is the time step, I is the identity matrix, $\frac{\partial \vec{R}}{\partial \vec{Q}}$ is the Jacobian matrix, $\Delta \vec{Q}^n$ is a forward difference of the conserved variable vector given by $\Delta \vec{Q}^n = \vec{Q}^{n+1} - \vec{Q}^n$, and $\vec{R}^n$ is the residual evaluated at time level n. During the solution of Eq. 2.58 to steady state, the update to the conserved variable vector, $\Delta \vec{Q}^n$, can sometimes result in non-physical values of density and pressure. In this case, it is necessary to limit the density and pressure to small but physical values to prevent the solver from diverging. To avoid the computational cost of frequently converting from conserved variables to primitive variables and vice versa, SENSEI only stores the primitive variables. Therefore, Eq. 2.58 is solved by multiplying the left hand side by a transformation matrix which converts the conserved variable vector, $\vec{Q}$, to the primitive variable vector, $\vec{q}$, such that

$$\left[\frac{\Omega}{\Delta t}\frac{\partial \vec{Q}}{\partial \vec{q}} + \frac{\partial \vec{R}}{\partial \vec{q}}\right]^n \Delta \vec{q}^{\,n} = -\vec{R}^n \tag{2.59}$$

where $\frac{\partial \vec{Q}}{\partial \vec{q}}$ is the transformation matrix and $\vec{q}$ is the primitive variable vector which for 2-D is given by $\vec{q} = [\rho, u, v, p]^T$.

It is important to note that currently the flow solution is interpolated from one grid to another in the same manner as the weight function as outlined in the previous section. While a bilinear interpolation is better than reinitializing the solution, it is not a conservative interpolation. Therefore, from one grid to another, the iterative residuals will increase. A better solution might be to reconstruct the solution on the original grid then restrict it back to the adapted grid. This improved interpolation is left for future work.

2.8.2 Adaptation Module: SAM

The adaptation schemes investigated in this work are coded using the Fortran 03/08 coding standard. The modern Fortran features and stylistic practices outlined in [27] are used to ensure readability and maintainability of the code. The code is arranged in an object-oriented manner to allow for increased modularity and easy addition of new adaptation schemes in the future. The adaptation routines are incorporated into an existing structured adaptation module (SAM) [26, 32]. Previous versions of SAM were designed to be completely separate

from SENSEI. This implementation increased the overall computation time by requiring an excessive amount of input/output and bash scripting in order for SAM and SENSEI to properly communicate. The current implementation is incorporated directly within the flow solver. This setup greatly reduces the overhead cost by eliminating the input/output communication between SENSEI and SAM and by eliminating the unnecessary allocation and deallocation of memory associated with calling SENSEI multiple times.

To simplify the implementation of SAM within SENSEI and to minimize the footprint of SAM on the main SENSEI code base, all the routines needed by SAM are compiled into a separate library. Other than a few minor setup routines, the only evidence of SAM within the main routines of SENSEI is a single subroutine call which passes in the current grid and flow solution and returns the adapted grid and the flow solution interpolated onto the adapted grid. To better understand the flow of information within SAM, consider the following outline of a single call to SAM:

Adaptation Cycle

1. Compute/estimate truncation error on initial grid
2. Compute norms of truncation error
3. Create truncation error based weight function
4. Compute norms of weight function equidistribution error
5. Store initial truncation error and weight function equidistribution error norms for normalization if necessary
6. Check adaptive convergence using adaptation monitors
   (a) If converged, exit SAM
7. Adapt Grid
   (a) Relocate grid nodes using selected adaptation scheme
   (b) Under-relax grid update
   (c) Check for mesh crossover and reset grid if necessary
   (d) Interpolate weight function onto adapted grid
   (e) Check for convergence of adaptation scheme for the given weight function
8. Interpolate flow solution onto adapted grid
9. Compute norms of grid movement
10. Check adaptive convergence using adaptation monitors

(a) If converged, exit SAM

(b) If not converged:

i. Return to SENSEI and recompute flow solution if necessary

ii. Return to Step 1.

An outer adaptation cycle of SAM, defined by steps 1 - 10, refers to one complete call of the adaptation routines. An inner adaptation cycle, defined by steps 7a - 7e, refers to the convergence of the selected adaptation scheme for one particular weight function. Since truncation error is a function of the grid, the weight function changes immediately once the grid nodes are relocated. Therefore, running the inner adaptation cycles to convergence is unnecessary. In practice, a minimal number of inner adaptation cycles are needed to move the grid significantly. For this study, the maximum number of inner adaptation cycles was set to 10 for all test cases.

2.9 Results

2.9.1 Test Case Selection & Evaluation

To test these adaptation schemes, four inviscid test cases are selected which encompass both the subsonic and supersonic flow regimes. The test cases include a supersonic expansion fan, an oblique shock wave, supersonic flow over a diamond airfoil, and subsonic flow over a Karman-Trefftz airfoil. These tests cases are explicitly selected because there exists a known exact solution for each case. With the knowledge of an exact solution, exact truncation error is formulated and used to drive adaptation. Exact solutions also permit the direct computation of discretization error improvements on adapted grids making the comparison of the various adaptation schemes straightforward. It is important to note that while the use of exact truncation error and discretization error is appropriate for the current study, practical applications will require truncation error and discretization error estimation [12, 13]. Exact truncation error for each case is computed by inserting the exact solution, $\tilde{u}$, into Eq. 2.1 and simplifying using Eq. 2.2a such that

$$\tau_h(\tilde{u}) = L_h(I^h\tilde{u}) \tag{2.60}$$

where $L_h(\cdot)$ is the discrete residual and I^h is an operator which restricts the exact solution to a discrete space.

In the following sections, test case descriptions and results are presented. For each case, truncation error and discretization error improvements obtained with each adaptation scheme are reported and compared. Truncation error improvement for the k^{th} governing equation is defined as the L_p norm of truncation error on the uniform or initial grid divided by L_p norm of truncation error on the adapted grid

$$Truncation\ Error\ Improvement = \frac{||\tau_h^k||_{p,initial}}{||\tau_h^k||_{p,adapted}} \ . \tag{2.61}$$

Likewise, discretization error improvement for the k^{th} primitive variable is defined as the L_p norm of discretization error on the uniform or initial grid divided by the L_p norm of discretization error on the adapted grid

$$Discretization\ Error\ Improvement = \frac{||\varepsilon_h^k||_{p,initial}}{||\varepsilon_h^k||_{p,adapted}} \ . \tag{2.62}$$

2.9.2 Supersonic Expansion Fan

Case Description

The first test case is a 2-D supersonic expansion fan generated by a Mach 1.2 flow around a 12° downward turn. Three instances of this case are examined and will be denoted as the "edge case", the "corner case", and the "full domain". The computational domain for the edge case and corner case can be found in Fig. 2.5a given by the blue square and black square, respectively. The grid for both the edge and corner cases is a 65x65 node grid extending from $0\,m$ to $1\,m$ in both the x and y directions. Additionally for the corner case, grids ranging in size from a 17x17 node grid to a 129x129 node grid are studied. The computational domain for the full domain case can be found in Fig. 2.5b. For the full domain case, the grid extends completely around the 12° downward turn. Grids ranging in size from a 17x9 node grid to a 257x129 node grid are examined.

The inflow conditions for all expansion fan cases are Mach 1.2 flow with a static pressure of $p_{\infty} = 100\ kPa$ and a static temperature of $T_{\infty} = 273\ K$. The boundary conditions for the corner case and the full domain case can be found in Fig. 2.5c and Fig. 2.5d, respectively. It is important to note that an exact solution boundary condition simply refers to evaluating the exact solution in a ghost cell in order to properly evaluate the inviscid flux at the boundary face. The boundary conditions for the edge case are identical to the corner case except the supersonic inflow boundary condition is replaced with an exact solution boundary condition. For the inviscid wall boundary condition, ghost cell values of the primitive variables are set such that the velocity normal to the wall is zero. For the supersonic outflow boundary condition, all primitive variables in the ghost cells are extrapolated from the interior.

The exact solution for the corner case and the full domain case can be found in Fig. 2.5e and Fig. 2.5f, respectively. The exact solution is computed using the Prandtl-Meyer solution for 2-D supersonic expansion waves given by [33, 34]

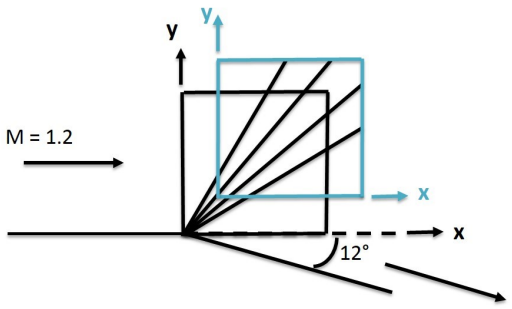
$$\nu \equiv \sqrt{\frac{\gamma+1}{\gamma-1}}\tan^{-1}\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)} - \tan^{-1}\sqrt{M^2-1} \tag{2.63}$$

where M is the local Mach number, γ is the ratio of specific heats for air, and ν is the local Prandtl-Meyer angle. The Prandtl-Meyer angle, and therefore the Mach number, is determined everywhere in the flow by noting that

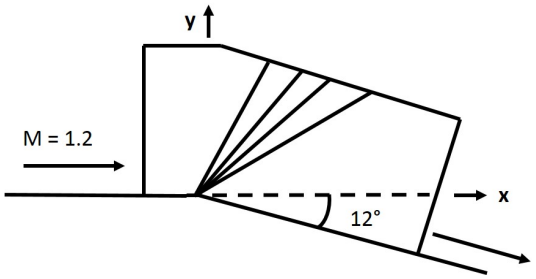
$$\theta \mp \nu = constant \tag{2.64}$$

on c^+ and c^- characteristic lines, respectively, where θ is the local flow angle. The static pressure, p_{∞}, and the static temperature, T_{∞}, may be computed everywhere by noting that there is no loss in the total conditions through the expansion fan.

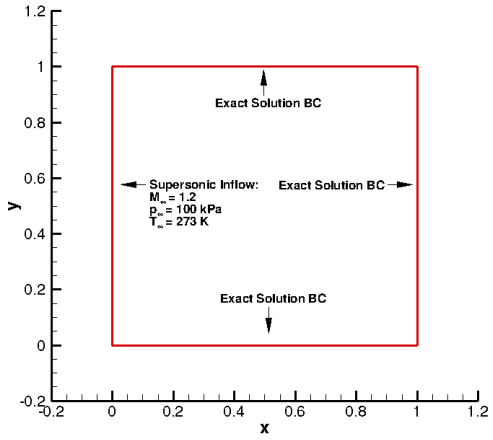
Numerical solutions are obtained using a fully upwinded MUSCL scheme with Roe's flux difference splitting scheme [28]. Stability of the second order reconstruction is attained using the van Albada limiter [35].



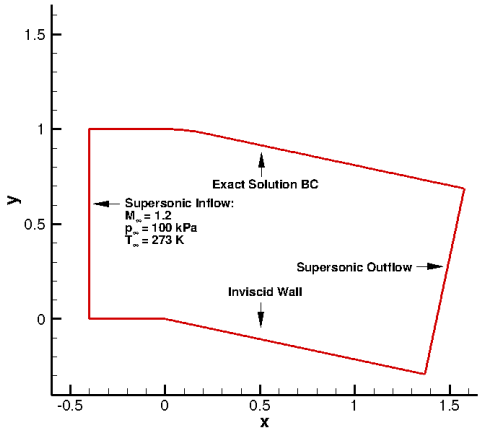
(a) Edge Case & Corner Case: Geometry



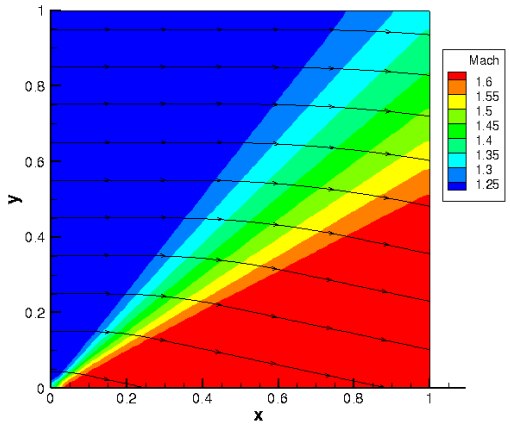
(b) Full Domain: Geometry



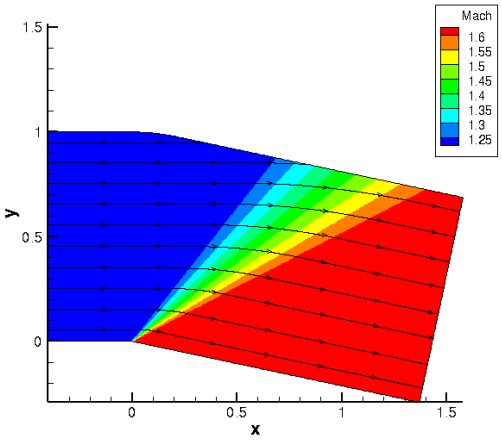
(c) Corner Case: Boundary Conditions



(d) Full Domain: Boundary Conditions



(e) Corner Case: Exact Solution - Mach



(f) Full Domain: Exact Solution - Mach

Figure 2.5: Expansion Fan: Case Description

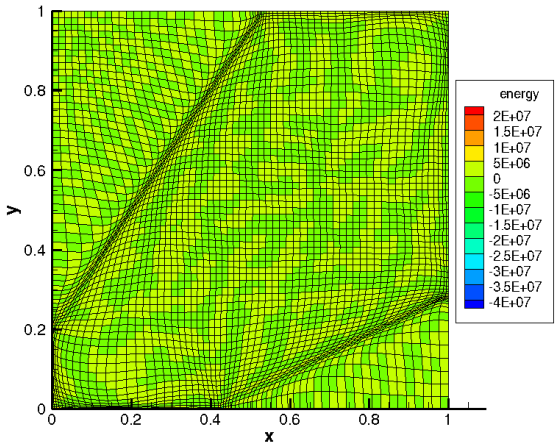
Results - Edge Case

Truncation error and discretization error improvements for the edge case of the supersonic expansion fan achieved by each adaptation scheme may be found in Table 2.1. In the best case, truncation error is decreased by a factor of ten when using the center of mass approach [19, 7]. For the same adaptation scheme, discretization error is reduced by a factor of six. Anderson's adaptation scheme [15] also performs well by achieving a six time improvement in discretization error. The Brackbill and Saltzman [6] and Deformation [21] adaptation schemes still achieve some amount of discretization error reduction, but at best it is half the reduction seen with the other two methods.

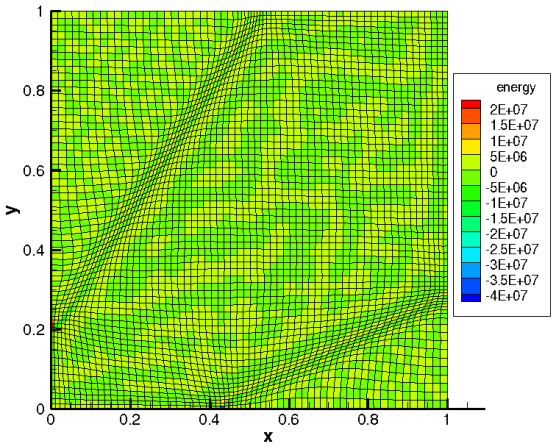
Qualitative results of truncation error and discretization error reductions for the edge case of the supersonic expansion fan are presented in Figures 2.6 and 2.7, respectively. Truncation error results are only presented for the energy equation while discretization error results are only presented for pressure. Although results are only presented for energy truncation error and pressure discretization error, the remaining variables exhibit qualitatively similar behavior. For this case, the majority of truncation error is located at the initial and final waves of the expansion fan. Since truncation error is high in these regions of the domain, the adaptation schemes pull nodes to the beginning and end of the expansion fan as illustrated by the adapted grids in Fig. 2.6a - 2.6d. The Anderson and center of mass adaptation schemes produce very similar adapted grids. The Brackbill and Saltzman adaptation scheme appears to adapt to the edges of the expansion fan appropriately but not to the extent of center of mass and Anderson's adaptation scheme. Better selections for the coefficients λ_o and λ_v could possibly improve the results for this adaptation scheme, but as previously mentioned, a systematic selection of these coefficients is unclear. The Deformation adaptation scheme seems to adapt to the final wave of the expansion fan quite well but struggles to resolve the leading wave. This could possibly be due to using too large of a pseudo time step during the adaptation process.

Table 2.1: Expansion Fan (Edge): Improvement of L_2 Norm of Error

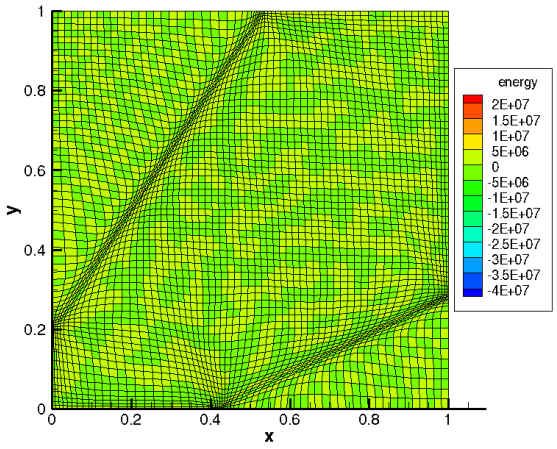
	Truncation Error Improvement				Discretization Error Improvement			
Adaptation Scheme	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
Anderson	8.44	8.90	7.75	8.62	5.95	6.09	5.28	6.13
Brackbill & Saltzman	4.30	4.71	4.12	4.51	2.81	2.81	2.84	2.81
Center of Mass	9.95	10.46	9.25	10.10	6.30	6.21	6.55	6.29
Deformation	3.87	4.81	2.59	3.95	2.09	1.94	2.99	2.00



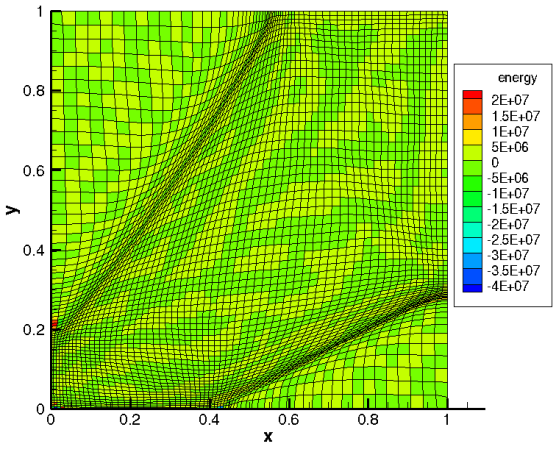
(a) Anderson



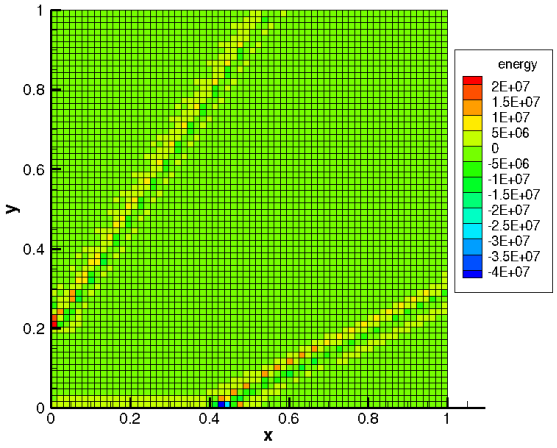
(b) Brackbill & Saltzman



(c) Center of Mass

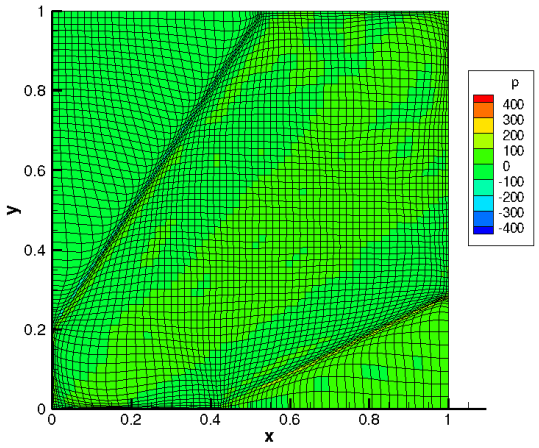


(d) Deformation

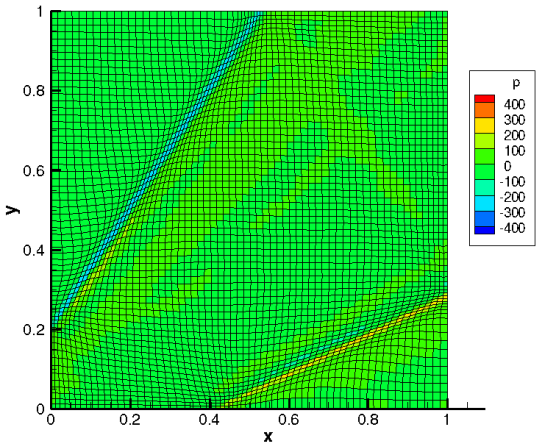


(e) Uniform

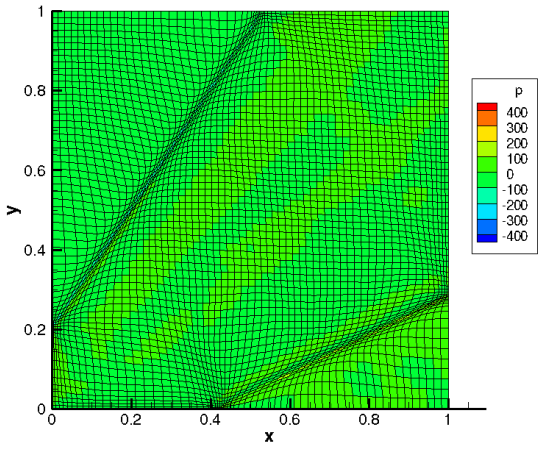
Figure 2.6: Expansion Fan (Edge): TE Energy



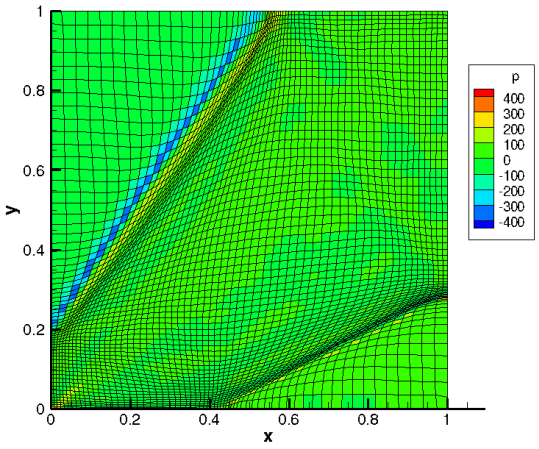
(a) Anderson



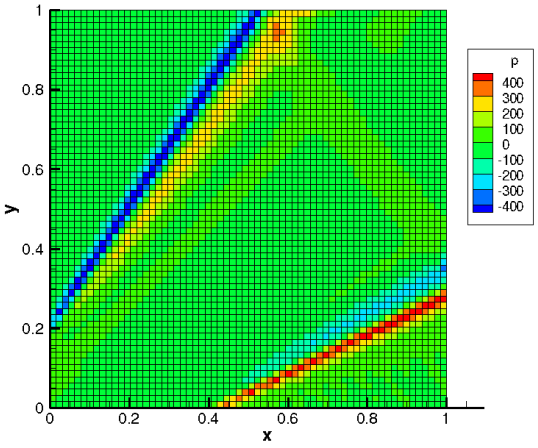
(b) Brackbill & Saltzman



(c) Center of Mass



(d) Deformation Method



(e) Uniform

Figure 2.7: Expansion Fan (Edge): DE Pressure

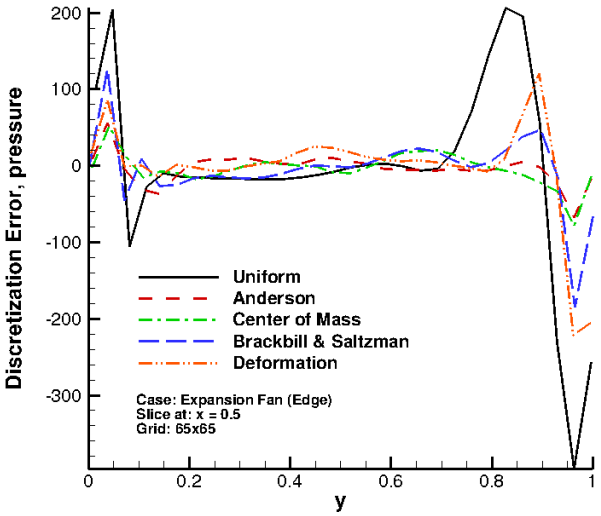


Figure 2.8: Expansion Fan (Edge): DE Pressure at x = 0.5 m

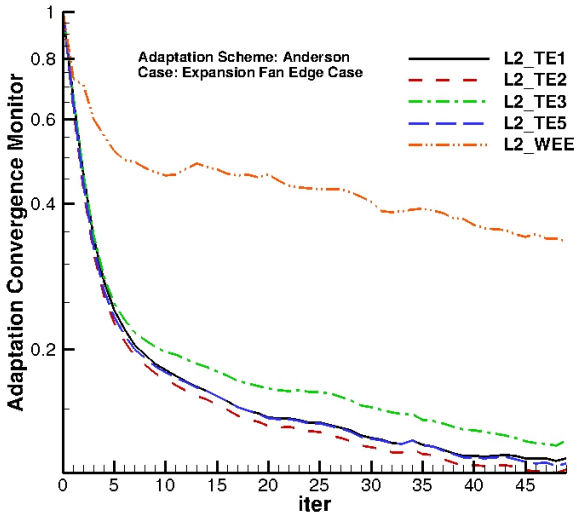


Figure 2.9: Expansion Fan (Edge): Adaptive Convergence - Anderson

Discretization error comparisons for pressure along a line through the domain at $x = 0.5\,m$ is shown in Fig. 2.8. The spikes in discretization error in this figure at approximately $y = 0.05\,m$ and $y = 0.95\,m$ correspond to the edges of the expansion fan. It can be seen that the Anderson and center of mass adaptation schemes best reduce these spikes in discretization error. Fig. 2.9 illustrates the adaptive convergence of the Anderson adaptation scheme. The L_2 norms of truncation error and the weight function equidistribution error are plotted against the number of outer adaptation cycles. For this case, the adaptive convergence is relatively smooth and monotonic. Although it may appear that these monitors could converge further, truncation error norms do not continue to converge and no increased benefit is found by adapting more than 50 outer adaptation cycles for this case.

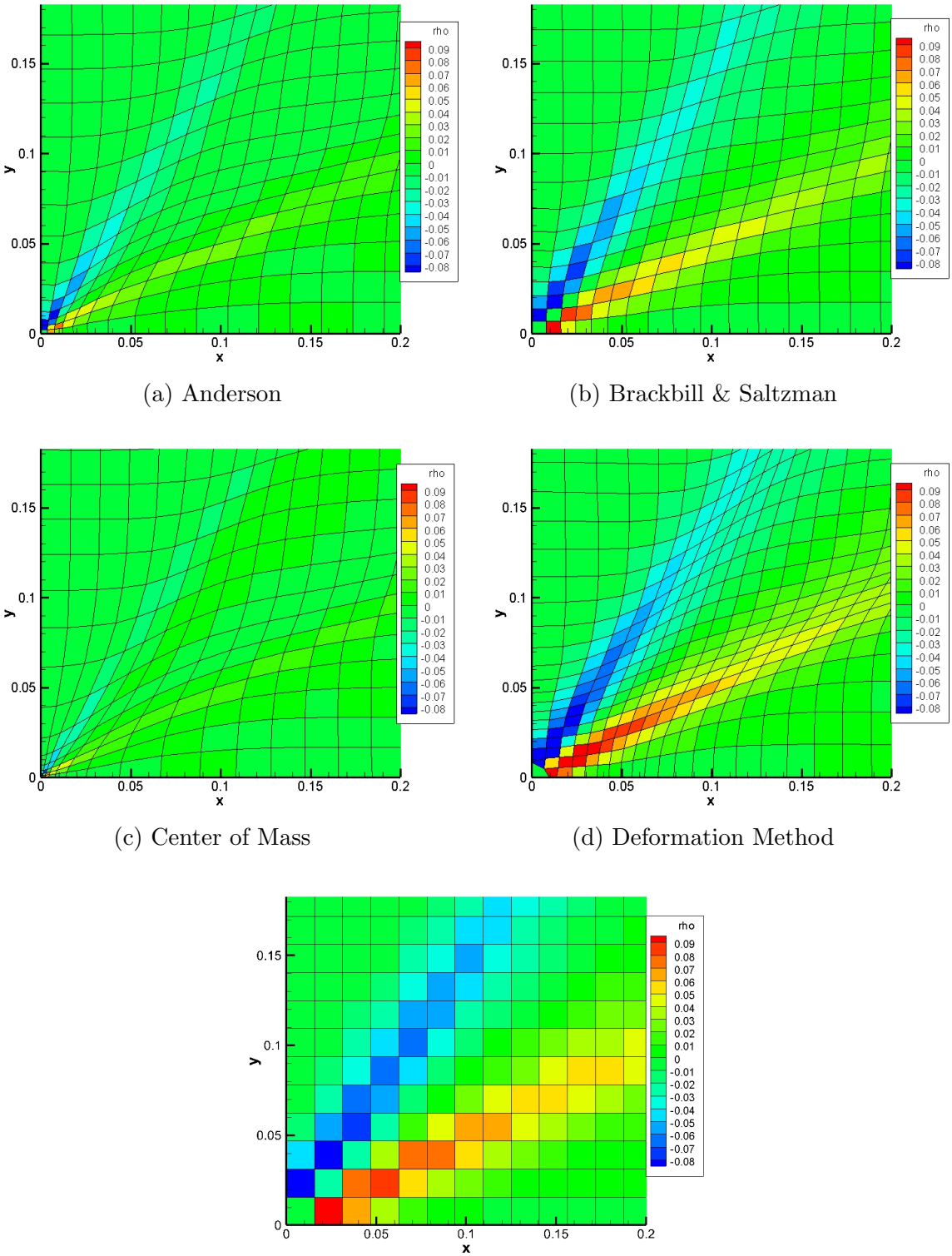
Results - Corner Case

Truncation and discretization error improvements for the corner case of the supersonic expansion fan achieved by each adaptation scheme may be found in Table 2.2. Similar to the edge case, the greatest truncation error reduction is achieved using the center of mass approach with about an eight time reduction. Discretization error for this adaptation scheme is reduced by a factor of approximately five for each primitive variable. Although Anderson's adaptation scheme does not perform as well here as with the edge case, over a three time improvement in discretization error is achieved. The Brackbill and Saltzman and Deformation adaptation schemes achieve a peak reduction factor in discretization error of 1.87 and 1.49, respectively, again, falling short of the other two adaptation schemes.

Most of the discretization error for this case is present in the bottom left corner of the domain at the root of the expansion fan. With this in mind, discretization error is only plotted for each adaptation scheme in the bottom left corner of the domain to better illustrate the results for each adaptation method. Discretization error in density is given in Fig. 2.10. It can be seen in Fig. 2.10a and Fig. 2.10c how well the Anderson and center of mass schemes reduce discretization error in this region relative to the Brackbill and Saltzman, Fig. 2.10b, and Deformation, Fig. 2.10d, adaptation schemes. The grids produced by the Anderson and center of mass schemes again look very similar except that center of mass is able to pull more points into the corner of the domain to better resolve the region around the discontinuity at the root of the expansion fan. On the contrary, the Brackbill and Saltzman and Deformation adaptation schemes are unable to relocate enough points to more accurately resolve the root of the expansion fan and achieve significant discretization error reductions.

Table 2.2: Expansion Fan (Corner): Improvement of L_2 Norm of Error

	Truncation Error Improvement				Discretization Error Improvement			
Adaptation Scheme	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
Anderson	4.66	5.19	3.66	4.84	3.35	3.26	3.60	3.29
Brackbill & Saltzman	2.40	2.51	2.25	2.42	1.82	1.80	1.87	1.80
Center of Mass	6.87	7.90	6.22	7.61	4.98	5.06	4.88	5.00
Deformation	2.71	2.76	2.06	2.68	1.45	1.45	1.49	1.45



(a) Anderson

(b) Brackbill & Saltzman

(c) Center of Mass

(d) Deformation Method

(e) Uniform

Figure 2.10: Expansion Fan (Corner): DE Density

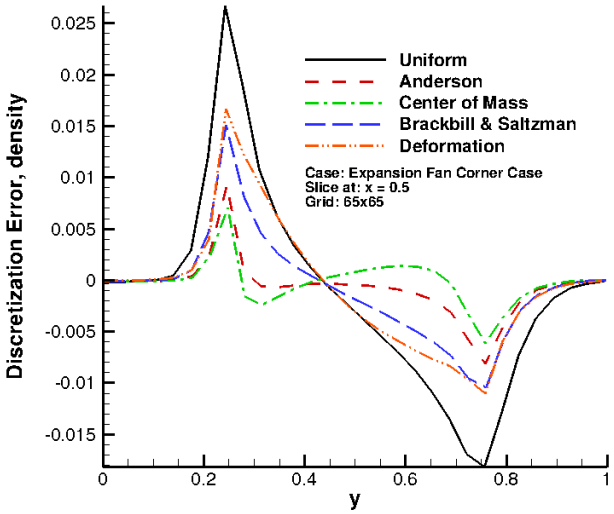


Figure 2.11: Expansion Fan (Corner): DE in Density at x = 0.5m

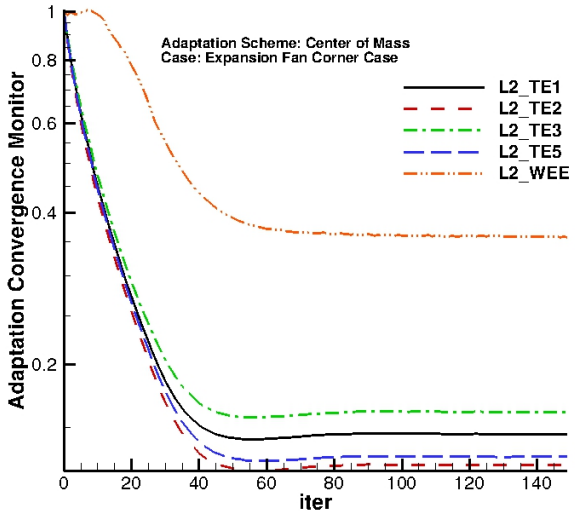


Figure 2.12: Expansion Fan (Corner): Adaptive Convergence - Center of Mass

Discretization error comparisons for density along a line through the domain at $x = 0.5\,m$ may be found in Fig. 2.11. Similar to the edge case, the spikes in discretization error in this figure at approximately $y = 0.25\,m$ and $y = 0.75\,m$ correspond to the edges of the expansion fan. Again, the Anderson and center of mass adaptation schemes are able to reduce these spikes in discretization error the most. Figure 2.12 illustrates the adaptive convergence of the center of mass adaptation scheme for this case. The L_2 norm of truncation error and the weight function equidistribution error are plotted against the number of outer adaptation cycles. Here adaptation is performed for 150 outer adaptation cycles. Although the adaptive process has not converged to the prescribed convergence criteria given in Section 2.7, past 50 outer adaptation cycles the adaptation is converged and no added benefit is achieved by further adaptation. In this case, a finer grid might be necessary to achieve the set convergence tolerance of the adaptation process.

Based upon the results for the edge case and corner case of the supersonic expansion fan, it is clear that the Anderson and center of mass adaptation schemes outperform the Brackbill and Saltzman and Deformation adaptation schemes. In order to better determine which adaptation scheme performs best for reducing discretization error, it is important to note that subsequent test cases and results will only examine the Anderson and center of mass adaptation schemes.

Grid Study

A grid study is conducted for the corner case of the 2-D expansion fan. The L_2 norm of discretization error in pressure is compared on uniform and adapted grids for five uniformly refined grid levels ranging from a 9x9 node grid to 129x129 node grid. Each adapted grid in this grid study is generated by independently adapting the corresponding uniform grid at each grid level. As shown in Fig. 2.13, the order of convergence between the uniform and adapted grids is quite similar. This is surprising considering the adaptation schemes used in this grid study provide no guarantee that the order of convergence on adapted grids will be the same as consistently refined unadapted grids. One possible explanation for this is that upon examination of the adapted grids it can be seen that the coarse adapted grids are qualitatively similar to grids generated by removing points from the finest adapted grid. Therefore, if the adapted grids are similar enough to a set of consistently refined grids based upon the finest adapted grid, it is possible to see a similar convergence to a set of uniform, unadapted grids. The order of convergence using discretization error in pressure for the three finest grid levels of the center of mass adapted grids is computed to be $\hat{p} = 0.68$. The same rate is also achieved on the three finest uniform grids. This convergence rate is consistent with what is expected for problems containing linear discontinuities such as this expansion fan case. Banks et al. [36] show that for problems with linear discontinuities the observed order of accuracy will reduce to

$$\frac{p_f}{p_f + 1} \leq \hat{p} \leq 1 \tag{2.65}$$

where p_f is the formal order of accuracy of the discretization for smooth problems. The range on the observed order of accuracy given in Eq. 2.65 accounts for any added dispersion introduced by the choice of limiter. With a formal order of accuracy of $p_f = 2$, the lower bound on the expected order of accuracy for this case is $\hat{p} = 2/3$. Fig. 2.13 also illustrates the tremendous savings which can be achieved when using r-adaptation as a means of reducing discretization error, especially for problems with linear or sub-linear convergence rates. For center of mass, to achieve the same level of discretization error as the coarsest adapted grid, a uniform grid which is three grid levels finer would be required. This uniform grid would have 64 times the number of cells of the coarsest adapted grid for this 2-D problem. In 3-D, the number of cells would increase by 512 times. This equates to huge computational savings especially when a lower level of discretization error may be desired. Similar trends can also be seen with the other primitive variables for this case.

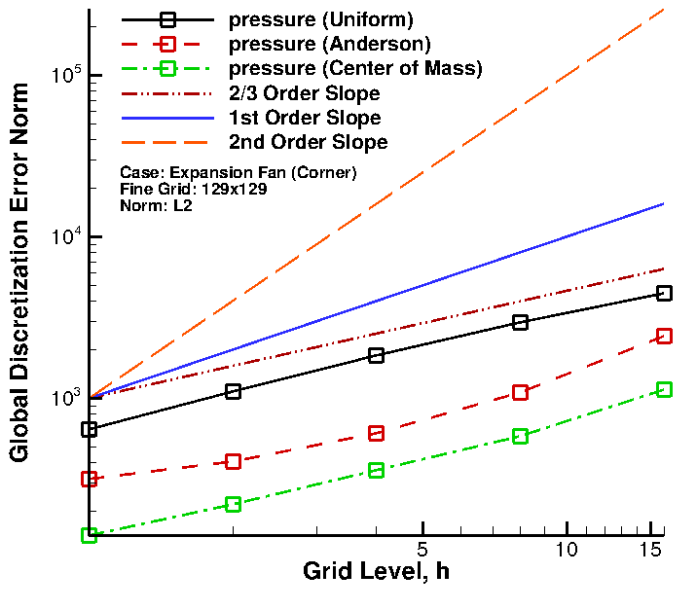


Figure 2.13: Expansion Fan (Corner): Global Discretization Error - Pressure

Results - Full Domain

Truncation error and discretization error improvements for the full domain case of the supersonic expansion fan can be seen in Table 2.3 for Anderson's adaptation scheme and in Table 2.4 for the center of mass adaptation scheme. Improvement in the L_2 norm of truncation error and discretization error are reported for five grid levels ranging from a 17x9 node grid to a 257x129 node grid. Both adaptation schemes achieve approximately a 3x improvement in discretization error norms for all primitive variables on the finest grid. But, while center of mass outperformed Anderson's adaptation scheme for both the edge and corner cases of the expansion fan, Anderson's adaptation scheme obtains slightly greater improvements for this case. This can especially be seen on the 33x17 node grid where center of mass only achieves about a 1.5x improvement whereas Anderson's scheme achieves about a 5x improvement for all primitive variables.

Table 2.3: Expansion Fan (Full Domain): Improvement of L_2 Norm of Error (Anderson)

	Adaptation Scheme: Anderson							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
257 x 129 (1)	1.86	2.09	1.43	1.96	3.49	3.16	3.18	3.50
129 x 65 (2)	2.50	2.83	1.65	2.92	3.76	3.70	3.21	3.79
65 x 33 (4)	1.65	1.95	1.04	1.73	4.21	3.61	2.99	4.21
33 x 17 (8)	2.12	2.17	2.33	2.17	5.02	5.26	4.94	5.13
17 x 9 (16)	2.86	2.86	2.22	3.09	2.82	2.94	2.39	2.91

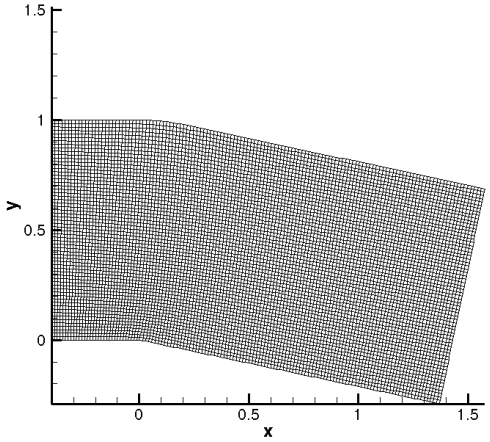
Table 2.4: Expansion Fan (Full Domain): Improvement of L_2 Norm of Error (Center of Mass)

	Adaptation Scheme: Center of Mass							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
257 x 129 (1)	1.38	1.42	1.22	1.41	2.91	2.66	2.83	2.87
129 x 65 (2)	1.35	1.46	0.96	1.36	2.97	2.88	2.92	2.93
65 x 33 (4)	1.48	1.52	1.41	1.51	3.91	3.73	3.69	3.91
33 x 17 (8)	1.32	1.28	1.43	1.32	1.47	1.21	2.06	1.55
17 x 9 (16)	1.28	1.41	0.96	1.31	2.52	2.33	2.42	2.58

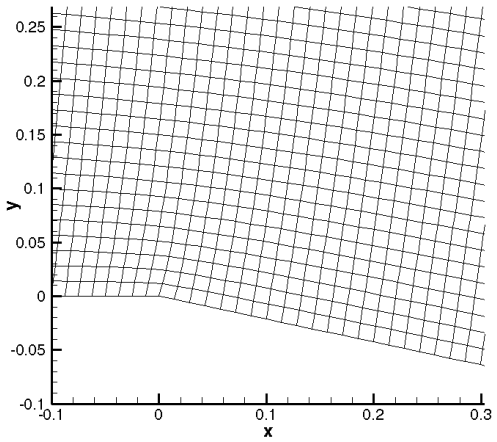
Plots of the initial and adapted grids for the 129x65 node grid can be seen in Fig. 2.14. When viewing the entire grid, the Anderson adapted grid and the center of mass adapted grid look nearly identical. Upon further examination of Fig. 2.14d and Fig. 2.14f, it can be

seen that the Anderson adapted grid provides much more clustering near the root of the expansion fan than the center of mass adapted grid. During the adaptation process, points were allowed to freely move across the location of the corner so that each adaptation scheme could relocate points as needed. To maintain the geometry of the turn, points were snapped to the corner location using the method described in Section 2.5.1. For this case, the center of mass adaptation scheme struggled to adapt across the corner point of the turn when using the scheme consistent boundary condition. Therefore, the Neumann boundary condition outlined in Section 2.4 was used for all center of mass adapted grids.

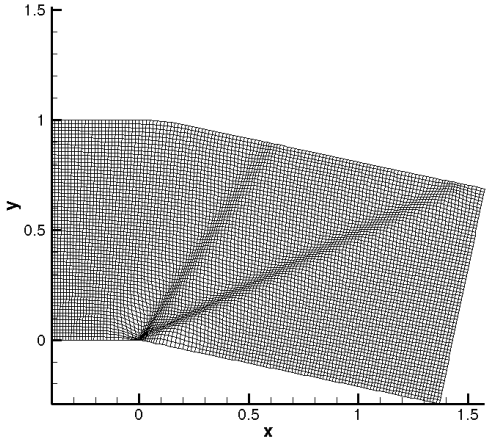
Truncation error and discretization error across the entire computational domain are plotted for this case in Fig. 2.15. Truncation error is only plotted for the continuity equation while discretization error is only plotted for density. Similar to the corner case, the truncation error is highest at the root of the expansion fan where the corner of the turn represents a singularity in the flow. Fig. 2.15b illustrates that this is also the location of highest discretization error. Discretization error generated at the root of the expansion fan is then transported along the leading and trailing edges of the fan. By increasing the grid resolution near the corner of the turn and the edges of the fan, both the Anderson and center of mass adaptation schemes are able to effectively eliminate the discretization error everywhere except immediately at the root of the expansion fan.



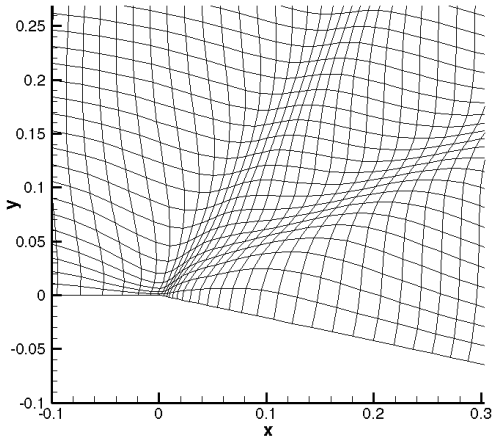
(a) Uniform



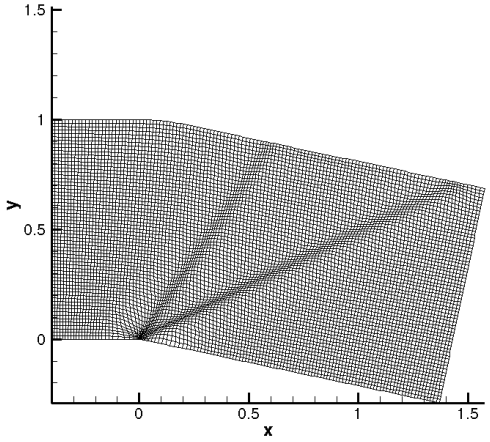
(b) Uniform: Inset



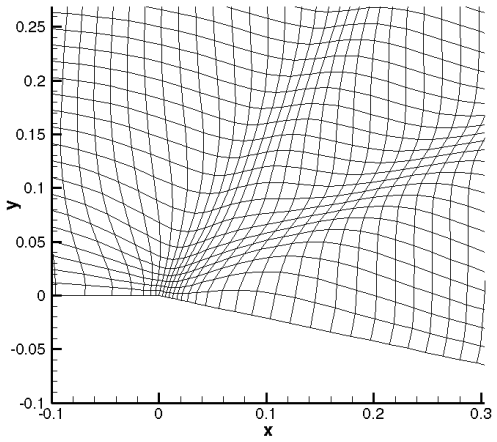
(c) Anderson



(d) Anderson: Inset

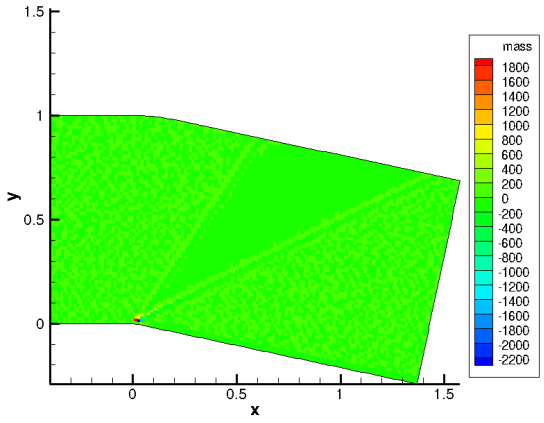


(e) Center of Mass

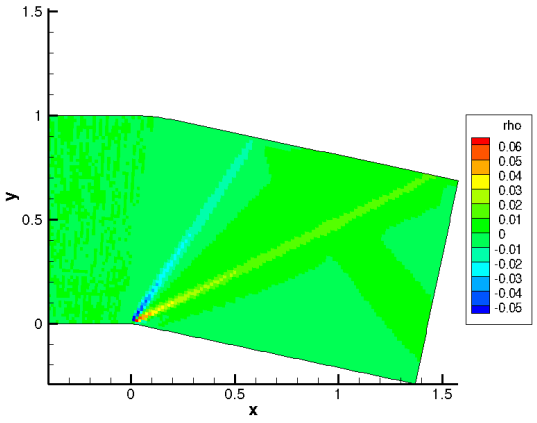


(f) Center of Mass: Inset

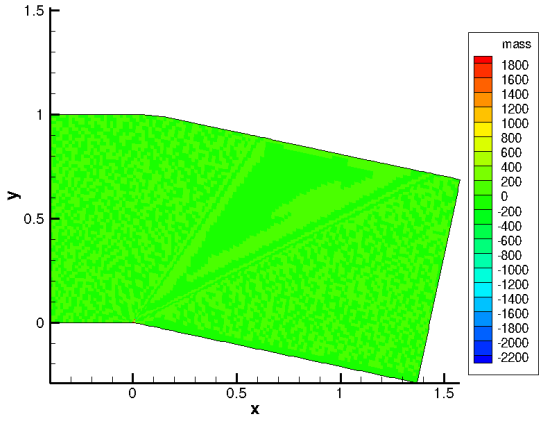
Figure 2.14: Expansion Fan (Full Domain): Initial & Adapted Grids (Grid Level: 129x65)



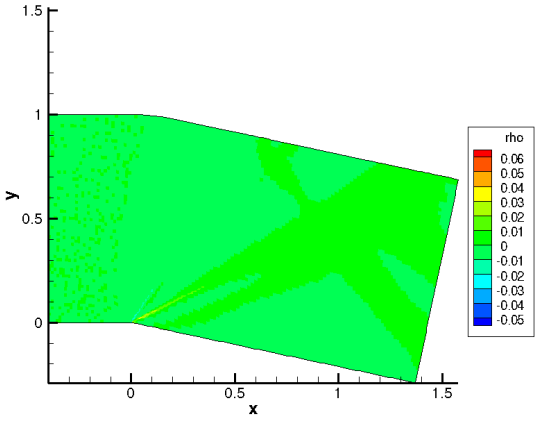
(a) Uniform: TE Mass



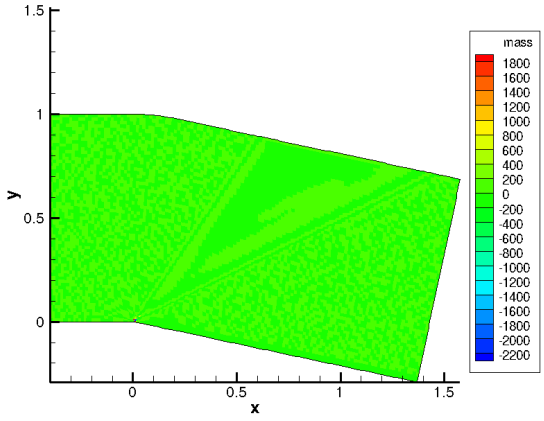
(b) Uniform: DE Density



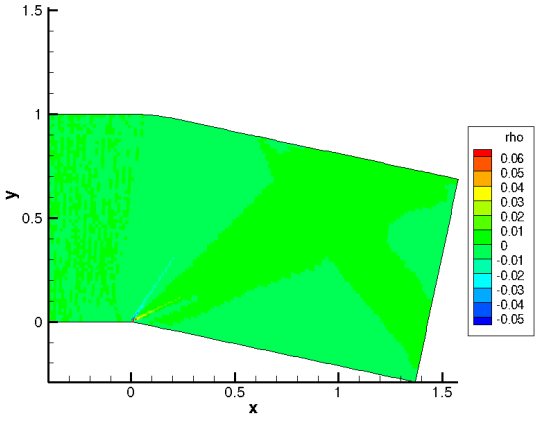
(c) Anderson: TE Mass



(d) Anderson: DE Density



(e) Center of Mass: TE Mass



(f) Center of Mass: DE Density

Figure 2.15: Expansion Fan (Full Domain): Truncation Error & Discretization Error (Grid Level: 129x65)

The convergence of the L_2 norm of discretization error in density for the initial and adapted grids is given in Fig. 2.16. The discretization error norm convergence for the remaining primitive variables exhibits a similar behavior. Interestingly enough, the discretization error norm on the uniform grid converges at a rate of $\hat{p} = 0.80$ whereas the norm for the corner case converged at a rate of $\hat{p} = 0.68$. This could possibly be due to better alignment of the initial grid for this case with the discontinuous flow features. Also, the convergence rate observed on the adapted grids is no longer constant across the entire range of grids. This nonlinear convergence could be attributed to the coarse adapted grids no longer being in the asymptotic range or to how the boundary conditions are implemented. For example, when the nearest point to the corner of the turn is snapped to the corner location, the resulting grid quality can become very poor in that region of the domain. If the adaptive convergence criteria is met at a point in the adaptive process when this poor grid quality is present, the resulting flow solution could have more error than it would with an otherwise smooth grid. But, at least for the three finest grids, the adapted grids appear to have a similar rate of convergence as the uniform grids. Like the corner case, Fig. 2.16 illustrates the significant savings that can be achieved with mesh adaptation. The 33x17 node Anderson adapted grid has less discretization error than a uniform grid over three grid levels finer.

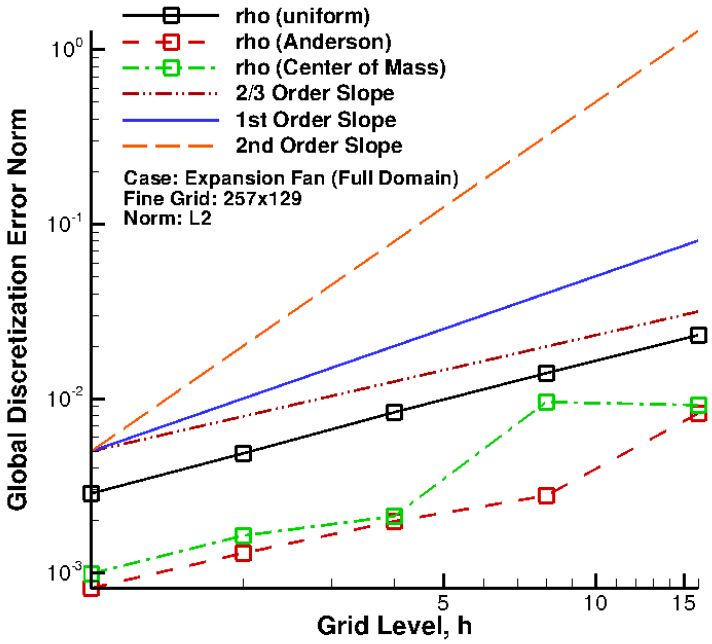


Figure 2.16: Expansion Fan (Full Domain): Global Discretization Error - Density

2.9.3 Oblique Shock Wave

Case Description

The second test case is a 2-D oblique shock generated by Mach 2.0 flow around a 10° compression turn. The computational domain for this case can be found in Fig. 2.17a. A series of grids ranging in size from a 17x9 node grid to a 257x129 node grid are examined. The inflow conditions are Mach 2.0 flow with a static pressure of $p_{\infty} = 100\ kPa$ and a static temperature of $T_{\infty} = 300\ K$. The boundary conditions for this case can be found in Fig. 2.17b. All boundary conditions labeled in Fig. 2.17b are implemented in the same manner as described for the 2-D supersonic expansion fan cases.

The exact solution for the oblique shock can be seen in Fig. 2.17c. The exact solution is computed using the Rankine-Hugoniot shock jump conditions for a 2-D oblique shock wave which are [33, 34]

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma+1)M_{n_1}^2}{2+(\gamma-1)M_{n_1}^2} \tag{2.66a}$$

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma+1}(M_{n_1}^2 - 1) \tag{2.66b}$$

$$\frac{M_{n_2}}{M_{n_1}} = \left[\frac{\rho_1}{\rho_2}\right]^{1/2}\left[\frac{p_2}{p_1}\right]^{-1/2} \tag{2.66c}$$

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)\left(\frac{\rho_1}{\rho_2}\right) \tag{2.66d}$$

where γ is the ratio of specific heats for air, M_{n_1} is the upstream Mach number normal to the shock wave, and M_{n_2} is the downstream Mach number normal to the shock wave. The upstream and downstream normal Mach numbers, M_{n_1} and M_{n_2}, are given by

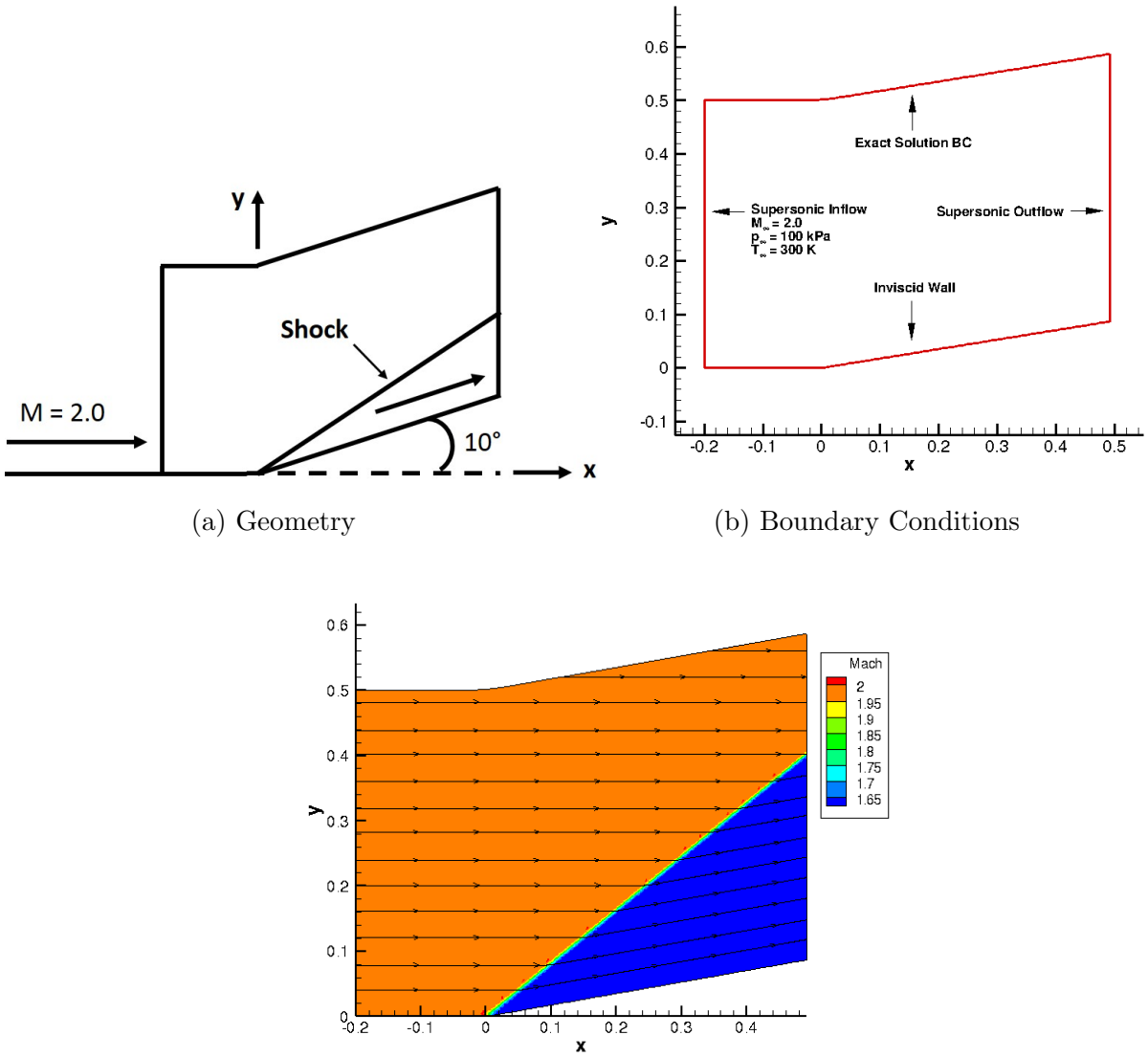
$$M_{n_1} = M_1 \sin(\beta) \tag{2.67a}$$

$$M_{n_2} = M_2 \sin(\beta - \theta) \tag{2.67b}$$

where θ is the turn angle and β is the wave angle. The wave angle can be computed as a function of the upstream Mach number, M_1, by solving the following transcendental equation for β

$$\tan(\theta) = 2\cot(\beta)\left[\frac{M_1^2\sin^2(\beta)-1}{M_1^2(\gamma+\cos(2\beta))+2}\right] . \tag{2.68}$$

Numerical solutions are obtained using a fully upwinded MUSCL scheme with Roe's flux difference splitting scheme [28]. Stability of the second order reconstruction is attained using the van Albada limiter [35].



(a) Geometry

(b) Boundary Conditions

(c) Exact Solution - Mach Number

Figure 2.17: Oblique Shock: Case Description

Results

Truncation error and discretization error improvements for the oblique shock case may be found in Table 2.5 for Anderson's adaptation scheme and in Table 2.6 for the center of mass adaptation scheme. Here, the L_1 norm is used instead of the L_2 norm since the L_1 norm is more appropriate for problems which contain discontinuities. The L_1 norm of truncation error and discretization error are reported for a series of grids ranging from a 17x9 node grid to a 257x129 node grid. The L_1 norm of truncation error improves significantly with a maximum improvement of approximately a 11x for both adaptation schemes on the finest grid. For discretization error, Anderson's adaptation scheme achieves between a 3x and 4x improvement in the L_1 norm for all primitive variables. But, unlike the expansion fan case, center of mass grossly outperforms the Anderson adaptation scheme by obtaining between a 10x and 11x improvement in the L_1 norm of discretization error for all the primitive variables on the finest grid.

Table 2.5: Oblique Shock: Improvement of L_1 Norm of Error (Anderson)

	Adaptation Scheme: Anderson							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
257 x 129 (1)	7.50	5.30	11.51	5.23	3.05	3.98	3.31	3.13
129 x 65 (2)	7.20	5.10	10.43	5.08	4.21	4.22	4.39	4.61
65 x 33 (4)	5.93	4.63	8.00	4.58	5.62	6.15	5.68	6.27
33 x 17 (8)	6.76	5.12	7.55	5.15	8.67	6.98	9.81	9.21
17 x 9 (16)	2.62	2.56	3.11	2.54	4.01	3.47	5.07	4.50

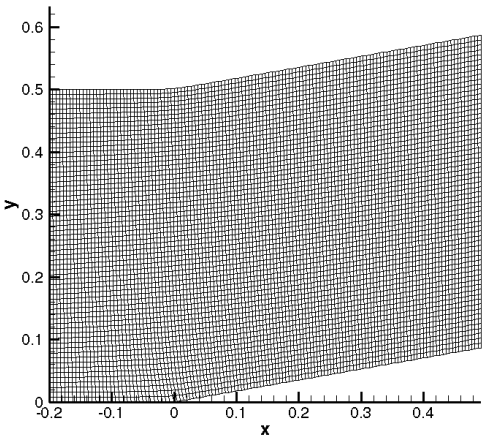
Table 2.6: Oblique Shock: Improvement of L_1 Norm of Error (Center of Mass)

	Adaptation Scheme: Center of Mass							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
257 x 129 (1)	9.09	6.21	11.24	6.28	10.86	11.84	10.68	11.44
129 x 65 (2)	8.22	5.72	10.48	5.51	10.04	10.40	9.91	10.69
65 x 33 (4)	7.60	5.48	9.14	5.52	11.08	11.49	11.00	11.95
33 x 17 (8)	5.75	4.68	6.58	4.81	12.62	11.11	16.10	13.69
17 x 9 (16)	3.14	2.83	4.03	2.88	5.76	5.44	8.38	6.33

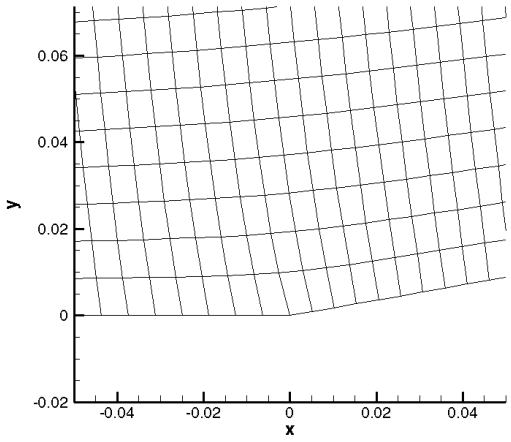
The uniform and adapted grids for the 129x65 node grid level may be seen in Fig. 2.18. The Anderson and center of mass adaptation schemes again produce qualitatively very similar grids. For this test case, the majority of the adaptation occurs right at the shock location

due to the large amounts of truncation error generated in that region. Fig. 2.18b, Fig. 2.18d and Fig. 2.18f show a better illustration of the root of the shock wave for the uniform grid, the Anderson adapted grid, and the center of mass adapted grid, respectively. The center of mass adaptation scheme is able to provide more resolution to the root of the shock than Anderson's adaptation scheme. This could possibly be a reason why center of mass is able to achieve significantly greater discretization error norm reductions. Also, while these adaptation schemes were not designed to align to flow features, both adaptation schemes appear to align to the shock wave to some degree. Compared to Anderson's adaptation scheme, center of mass seems to obtain better alignment with the shock likely because a Neumann boundary condition is not used which would require the grid lines at the boundary remain orthogonal. Moreover, upon examination of the grid cells in the shock region, it can be seen that both the center of mass and Anderson adaptation schemes suffer from poor grid quality. But, it is important to note that the highly skewed cells in this region do not hinder the adaptation schemes from achieving significant discretization error reductions.

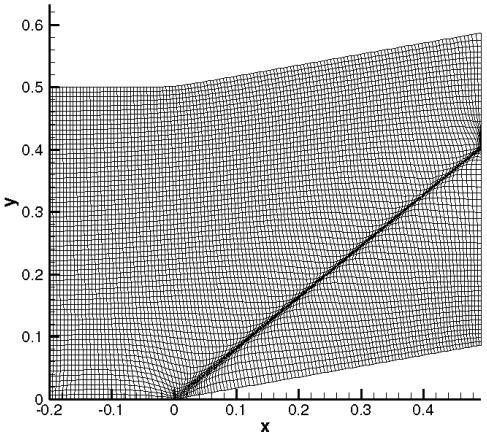
Truncation error and discretization error plotted across the computational domain may be seen in Fig. 2.19. Truncation error is plotted for the energy equation while discretization error is plotted for pressure. As previously mentioned, Fig. 2.19a illustrates the large amounts of truncation error generated at the shock location. As a result, the majority of the discretization error is also located at the shock as can be seen in Fig. 2.19b. With the unaligned uniform starting grid, discretization error in pressure at the shock is present in a wide band of cells. As the adaptation schemes attempt to resolve the shock, the pressure discretization error is confined to a very thin region.



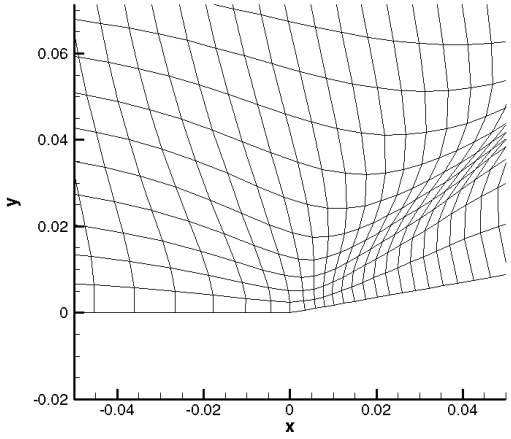
(a) Uniform



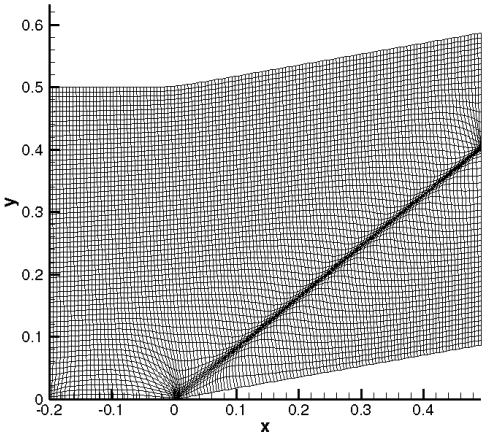
(b) Uniform: Inset



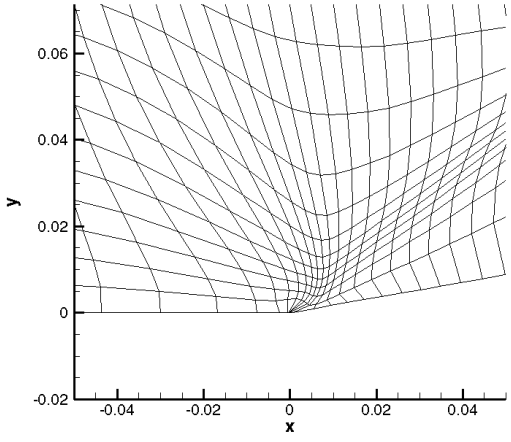
(c) Anderson



(d) Anderson: Inset

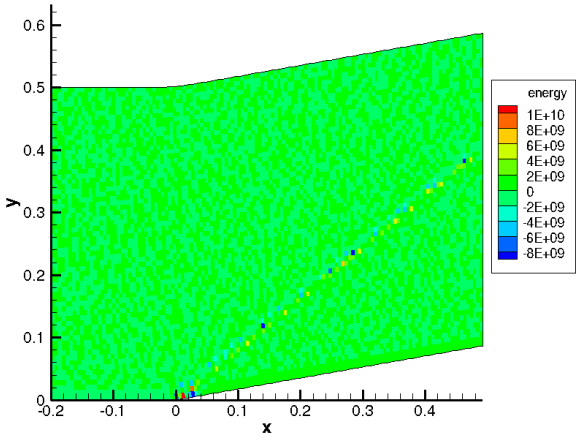


(e) Center of Mass

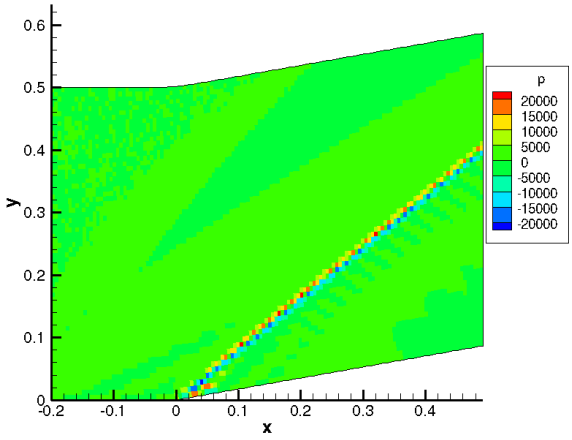


(f) Center of Mass: Inset

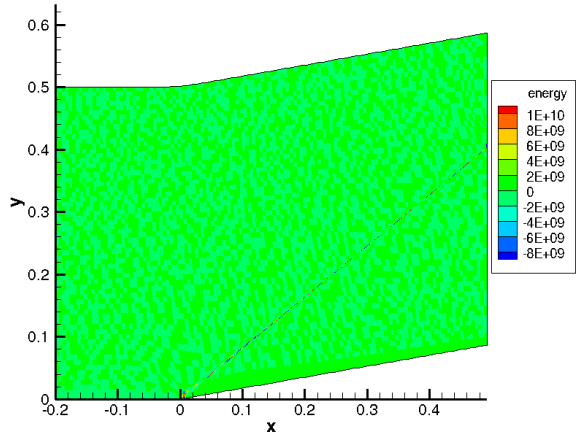
Figure 2.18: Oblique Shock: Initial & Adapted Grids (Grid Level: 129x65)



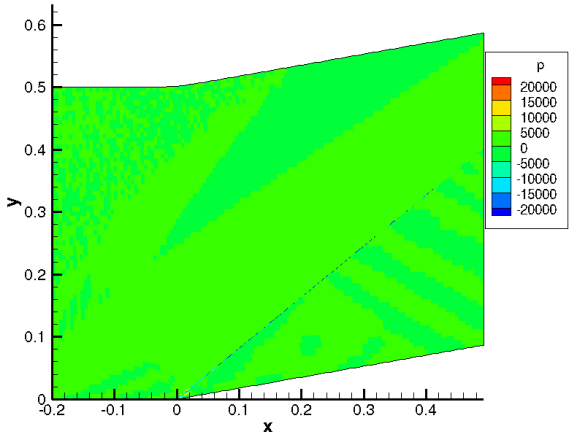
(a) Uniform: TE Energy



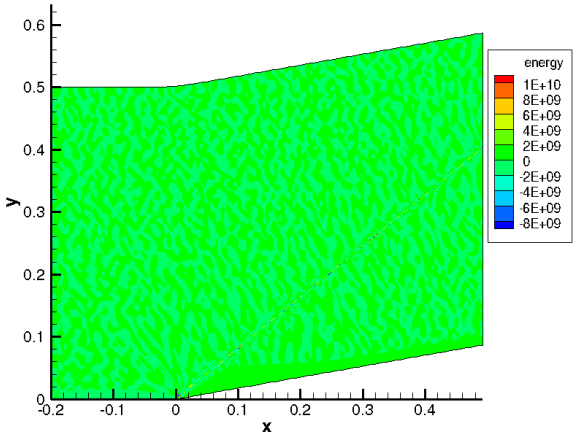
(b) Uniform: DE Pressure



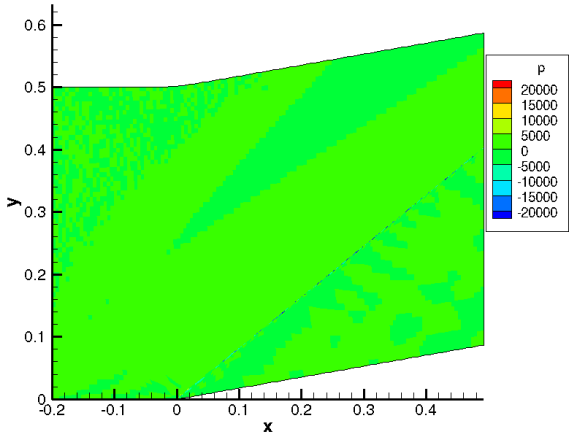
(c) Anderson: TE Energy



(d) Anderson: DE Pressure



(e) Center of Mass: TE Energy



(f) Center of Mass: DE Pressure

Figure 2.19: Oblique Shock: Truncation Error & Discretization Error (Grid Level: 129x65)

The convergence of the L_1 norm of pressure discretization error for the initial and adapted grids is given in Fig. 2.20. The uniform grids converge at a first order rate due to the presence of the shock wave. Both the Anderson and center of mass adapted grids exhibit a superconvergent property between the 17x9 node and 33x17 node grid levels. The coarseness of these grids could indicate that they are not within the asymptotic range. Between the 33x17 node and 257x129 node grid levels, both adaptation schemes exhibit a relatively constant rate of convergence. While the center of mass adapted grids appear to be converging at first order, the Anderson adapted grids converge at a sub-linear rate. For this case, a significant savings in computational cost can be gained with the center of mass adaptation scheme. The discretization error on the 33x17 node adapted grid is significantly less than on the uniform grid three grid levels finer.

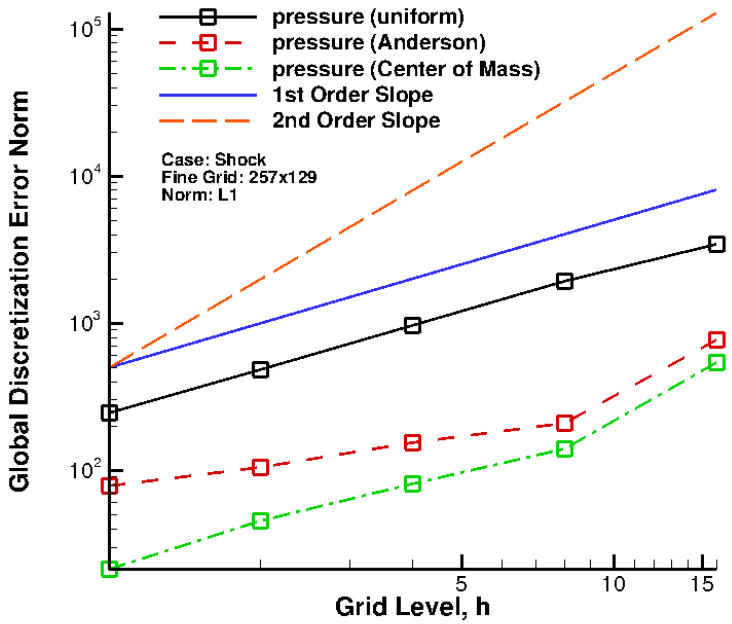


Figure 2.20: Oblique Shock: Global Discretization Error - Pressure

2.9.4 Diamond Airfoil

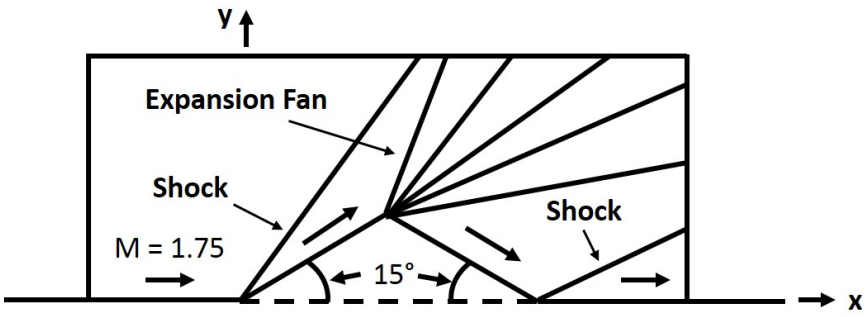
Case Description

The third test case is supersonic flow over a diamond airfoil at 0° angle of attack. The computational domain for this case can be found in Fig. 2.21a. The airfoil is symmetric about the x-axis and has an initial turn angle of 15°. The flow is characterized by an oblique shock emanating from the leading edge, a supersonic expansion fan at the top of the airfoil, and another oblique shock at the trailing edge. A series of grids ranging in size from a 33x17 node grid to a 257x129 node grid are examined.

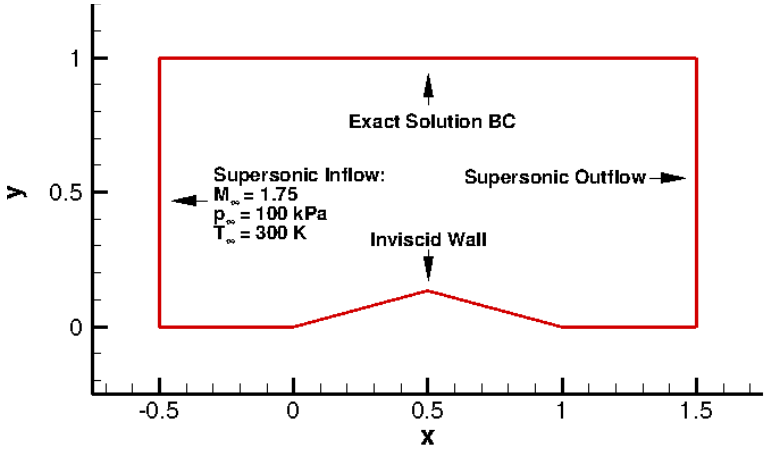
The inflow conditions are Mach 1.75 flow with a static pressure of $p_{\infty} = 100\ kPa$ and a static temperature of $T_{\infty} = 300\ K$. The boundary conditions for this case can be found in Fig. 2.21b. For completeness, it must be noted that in Fig. 2.21b the boundary condition upstream and downstream of the airfoil surface along the bottom of the domain should in fact be a symmetry plane. But, since a symmetry boundary condition and an inviscid wall boundary condition are implemented in a similar manner, for simplicity, an inviscid wall boundary condition is used along the entire bottom boundary of the domain.

The exact solution for the diamond airfoil can be seen in Fig. 2.21c. The exact solution is computed using both the Prandtl-Meyer solution for an expansion fan and the Rankine-Hugoniot shock jump conditions as given in the previous sections. It is important to note that in order for this case to have an exact solution everywhere in the computational domain the grid cannot extend into a region where any of the waves intersect.

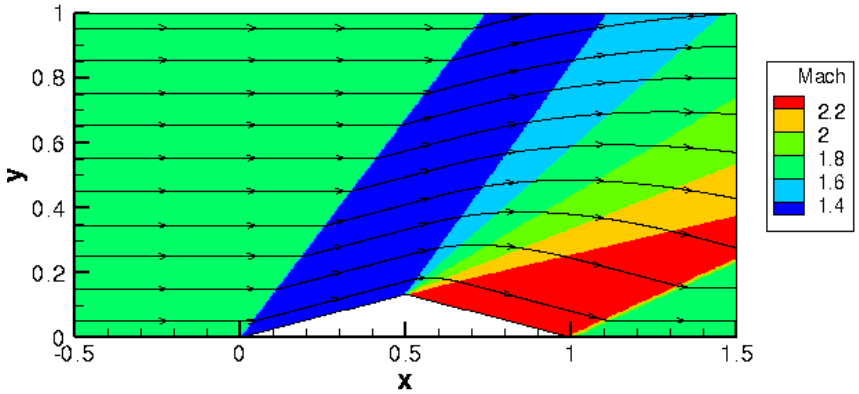
Numerical solutions are obtained using a fully upwinded MUSCL scheme with Roe's flux difference splitting scheme [28]. Stability of the second order reconstruction is attained using the van Albada limiter [35].



(a) Diamond Airfoil: Geometry



(b) Diamond Airfoil: Boundary Conditions



(c) Exact Solution - Mach Number

Figure 2.21: Diamond Airfoil: Case Description

Results

Truncation error and discretization error improvements for the diamond airfoil case can be seen in Table 2.7 for Anderson's adaptation scheme and in Table 2.8 for the center of mass adaptation scheme. The L_1 norm of truncation error and discretization error are reported for a series of grids ranging from a 33x17 node grid to a 257x129 node grid. For both adaptation schemes, the L_1 norm of truncation error improves between 2x and 3x for the continuity, x-momentum, and energy equations and about 8x for the y-momentum equation. For discretization error, the Anderson and center of mass adaptation schemes achieve between a 2x and 4x improvement in the L_1 norm for all primitive variables on the finest grid. Center of mass obtains the largest improvement for this case with about a 4x improvement in v-velocity discretization error.

Table 2.7: Diamond Airfoil: Improvement of L_1 Norm of Error (Anderson)

	Adaptation Scheme: Anderson							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
257 x 129 (1)	3.44	2.28	7.53	2.37	1.85	2.40	3.31	2.58
129 x 65 (2)	3.17	2.18	5.57	2.29	2.66	3.16	3.56	3.18
65 x 33 (4)	2.45	1.78	4.20	1.87	1.98	2.58	2.40	2.19
33 x 17 (8)	2.11	1.90	2.16	1.96	1.91	2.12	2.40	1.87

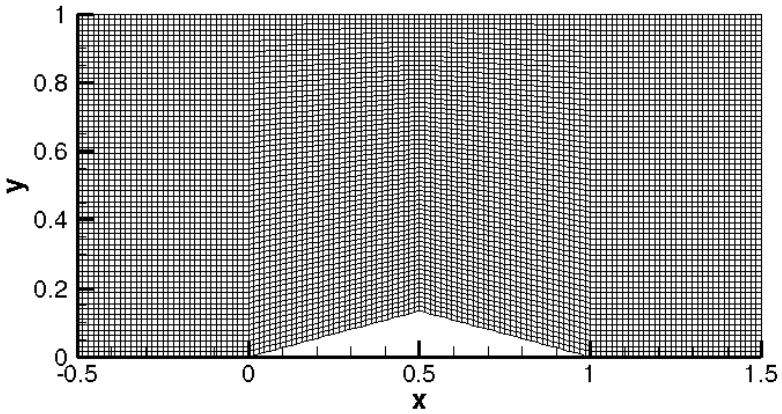
Table 2.8: Diamond Airfoil: Improvement of L_1 Norm of Error (Center of Mass)

	Adaptation Scheme: Center of Mass							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
257 x 129 (1)	3.85	2.41	8.08	2.52	2.33	3.43	3.92	3.33
129 x 65 (2)	3.10	2.10	6.81	2.16	2.24	3.25	3.26	2.84
65 x 33 (4)	2.51	1.90	4.51	1.95	1.93	2.31	2.30	1.91
33 x 17 (8)	2.01	1.75	2.32	1.86	1.63	1.70	1.94	1.51

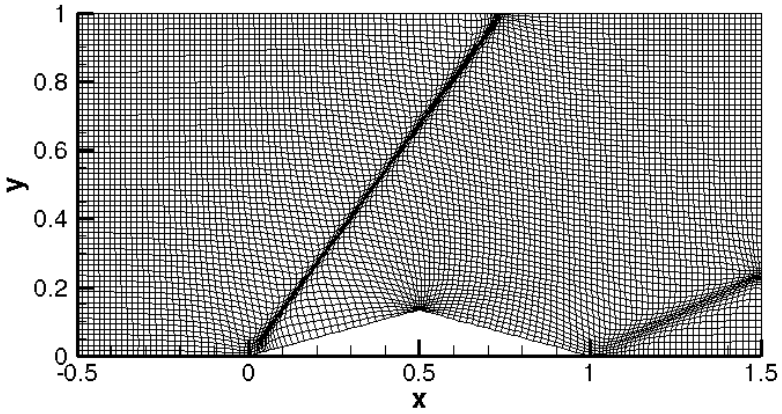
The uniform and adapted grids for the 129x65 node grid level may be seen in Fig. 2.22 - Fig. 2.25. As with the expansion fan case and the oblique shock case, the Anderson and center of mass adaptation schemes again produce qualitatively very similar grids. Although there are several discontinuities present, the majority of the adaptation surprisingly occurs at the leading edge shock location. A great deal of adaptation still occurs at the trailing edge shock location but not to the extent of the leading edge shock. An expanded view of the region near the leading edge shock, the region near the expansion fan, and the region near

the trailing edge shock can be seen in Fig. 2.23, Fig. 2.24, and Fig. 2.25, respectively. At the leading edge shock, center of mass is again able to provide more resolution than the Anderson adaptation scheme. At the root of the expansion fan, both adaptation schemes pull about the same number of points to the corner location to better resolve the singularity. And, similar to the leading edge shock, the center of mass adaptation scheme is able to relocate a larger number of points to the trailing edge shock than the Anderson adaptation scheme. For this case, center of mass struggled to produce results using the scheme consistent boundary condition. Therefore, a Neumann boundary condition was used instead.

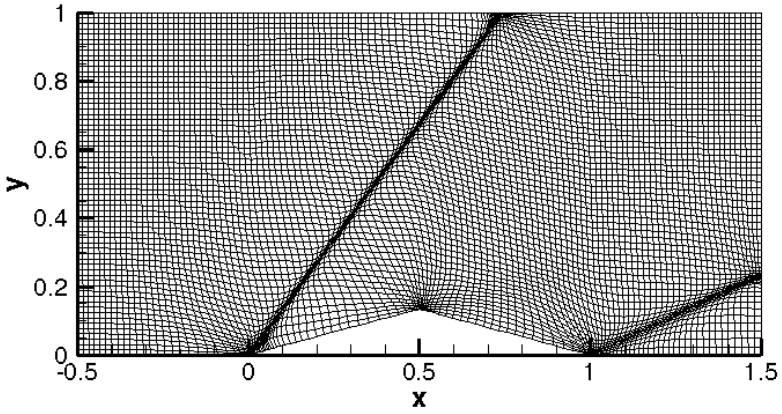
Truncation error and discretization error plotted across the computational domain may be seen in Fig. 2.26 for the 129x65 node grid level. Truncation error is plotted for the y-momentum equation while discretization error is plotted for v-velocity. Similar to the previous two cases, the discretization error at the shock is confined to a very thin region while the discretization error in the expansion fan is relatively localized to the root of the fan. Examination of Fig. 2.26a can help to further explain the behavior of the adaptation. For this case, the truncation error generated by the leading edge shock dominates the truncation error elsewhere in the domain. This is why the adaptation schemes respond more to this discontinuity than to the expansion fan or the trailing edge shock. It is also quite interesting to compare the behavior of the adaptation to the expansion fan for this case to the results presented in Section 2.9.2. Although the inflow and outflow Mach numbers of the expansion fan are different, the adaptation results presented in Section 2.9.2 are much more pronounced. The adaptation schemes not only adapt to the root of the fan but also to the leading and trailing edges. For the diamond airfoil, the adaptation schemes only appear to adapt to the root of the fan. This can be directly attributed to the presence of the shock waves. In the formulation of the truncation error based weight function, the expansion fan is viewed as a less important feature in comparison to the shock waves. This interaction between discontinuities could be a problem for this type of adaptation. The discontinuities which produce the highest truncation error are viewed with the highest priority while other discontinuities are neglected ultimately hindering the total amount of discretization error improvement which can be obtained.



(a) Uniform

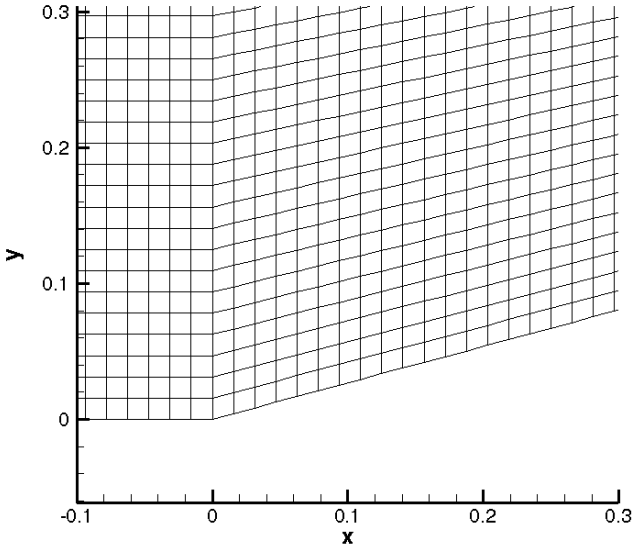


(b) Anderson

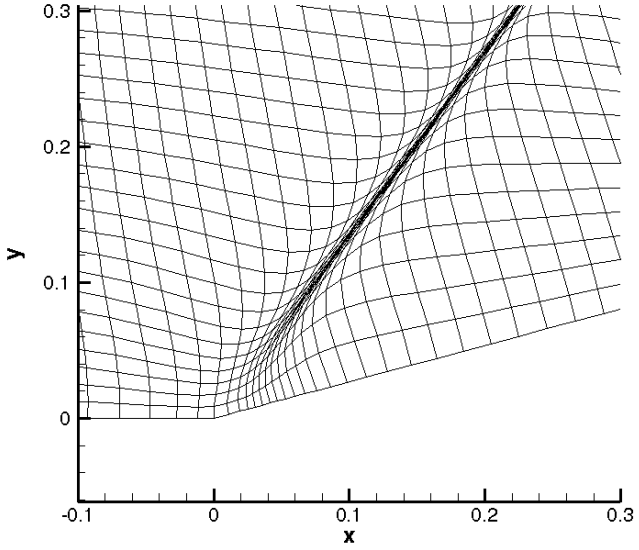


(c) Center of Mass

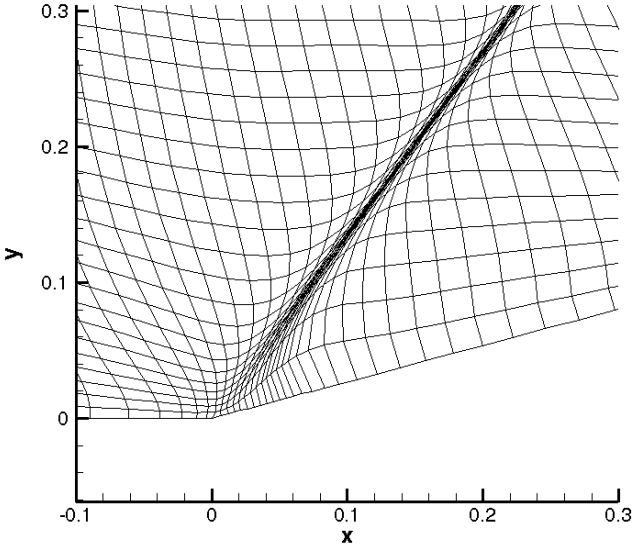
Figure 2.22: Diamond Airfoil: Initial & Adapted Grids (Grid Level: 129x65)



(a) Uniform

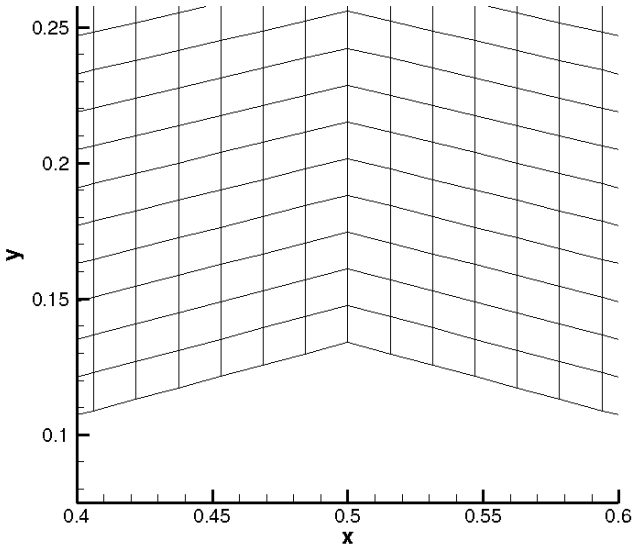


(b) Anderson

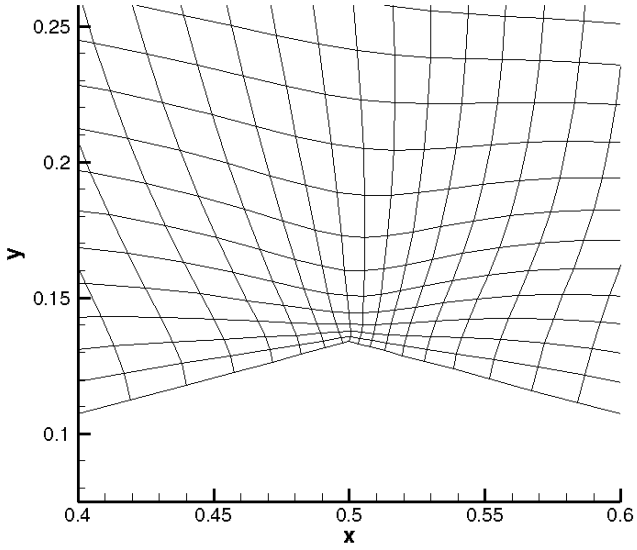


(c) Center of Mass

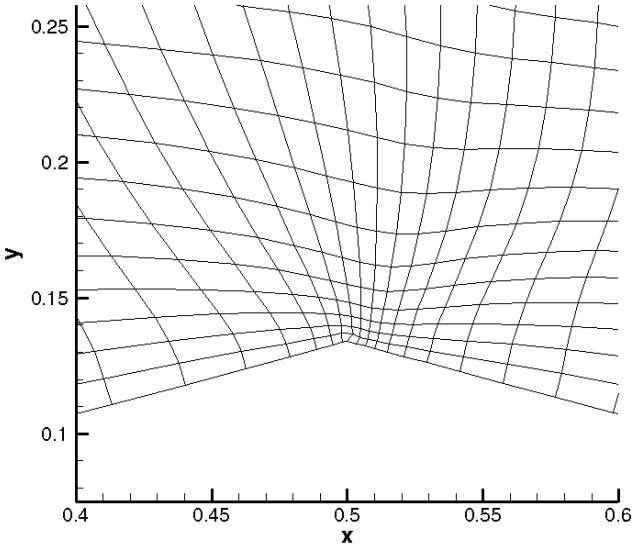
Figure 2.23: Diamond Airfoil: Leading Edge Shock Location (Grid Level: 129x65)



(a) Uniform

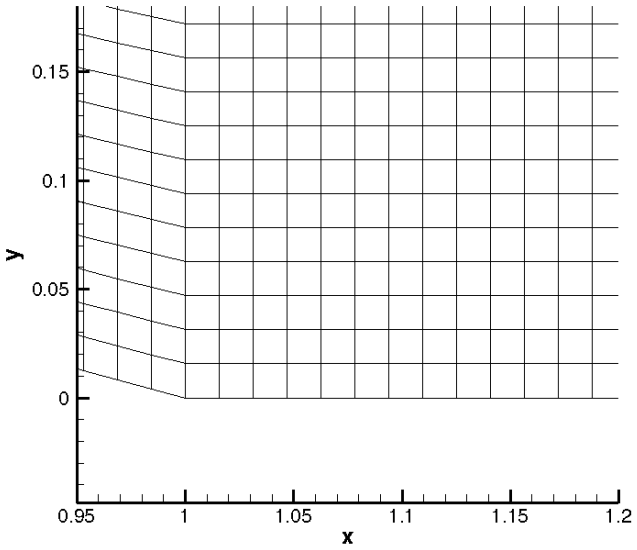


(b) Anderson

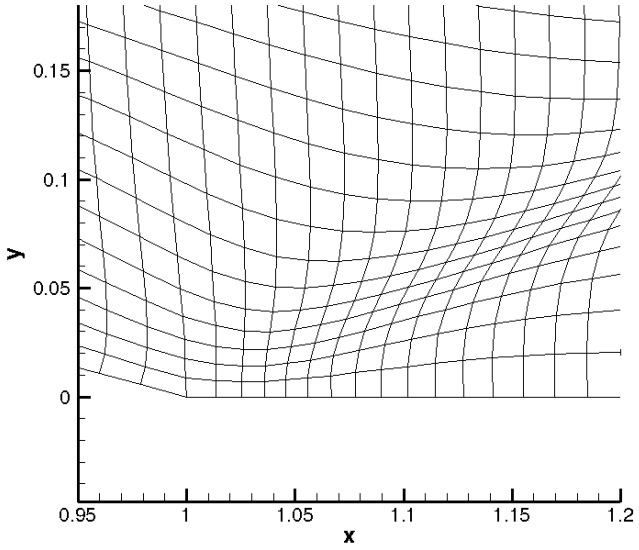


(c) Center of Mass

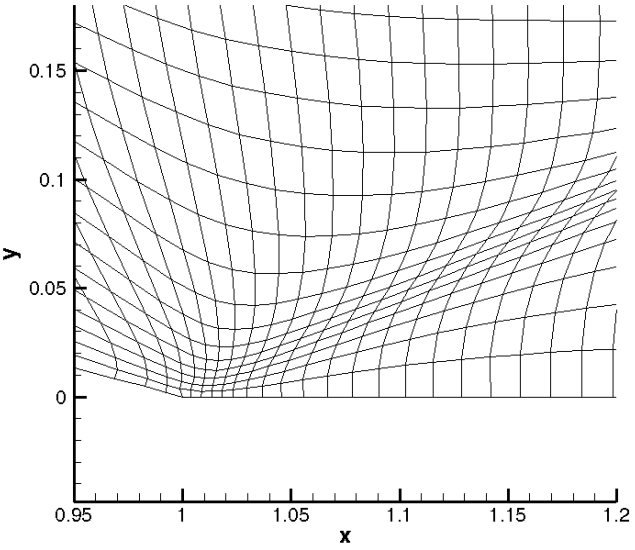
Figure 2.24: Diamond Airfoil: Expansion Fan Location (Grid Level: 129x65)



(a) Uniform

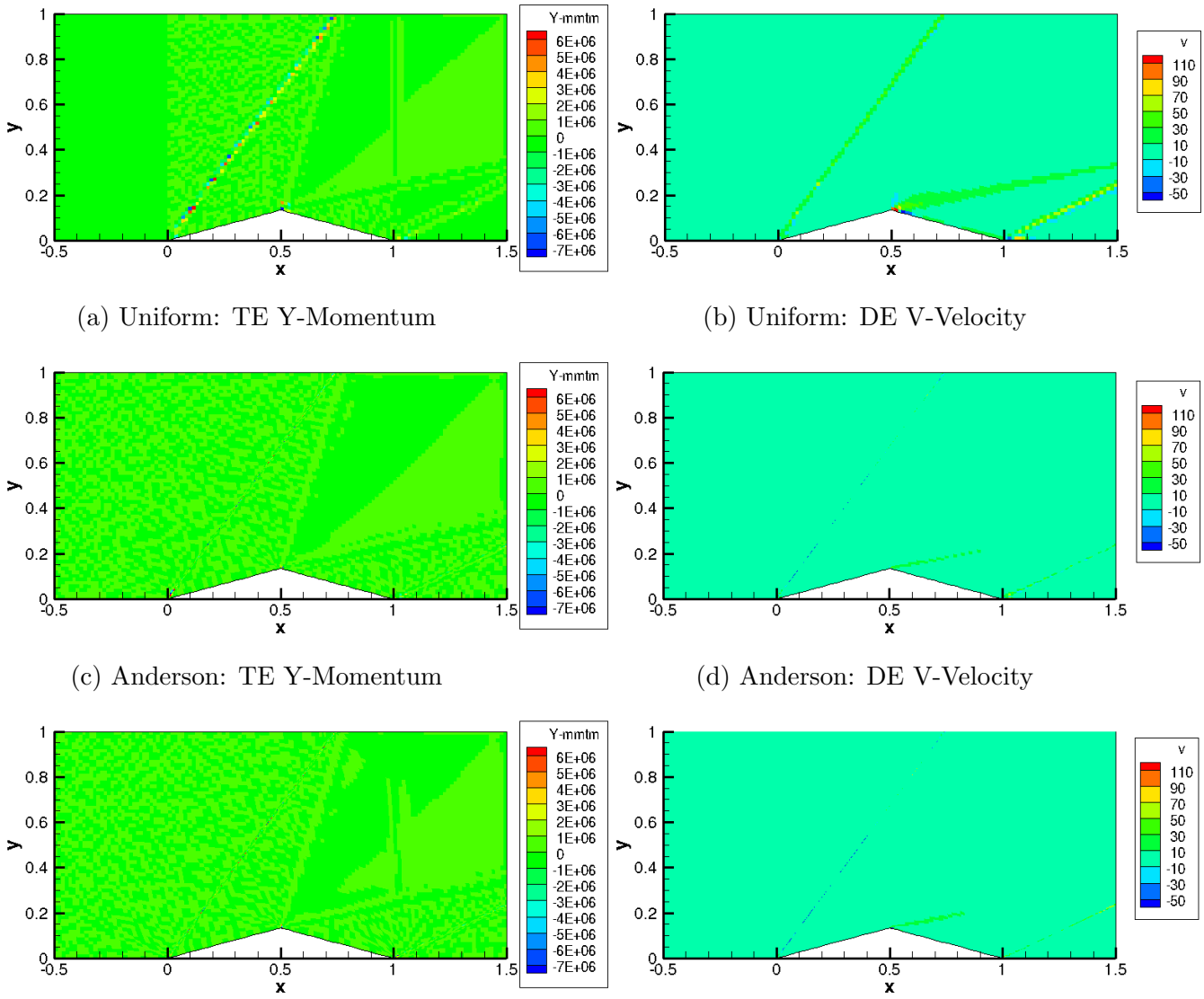


(b) Anderson



(c) Center of Mass

Figure 2.25: Diamond Airfoil: Trailing Edge Shock Location (Grid Level: 129x65)



(a) Uniform: TE Y-Momentum

(b) Uniform: DE V-Velocity

(c) Anderson: TE Y-Momentum

(d) Anderson: DE V-Velocity

(e) Center of Mass: TE Y-Momentum

(f) Center of Mass: DE V-Velocity

Figure 2.26: Diamond Airfoil: Truncation Error & Discretization Error (Grid Level: 129x65)

The convergence of the L_1 norm of v-velocity discretization error for the initial and adapted grids is given in Fig. 2.27. The uniform grids converge at a first order rate due to the presence of the shock waves and the expansion fan. Both the Anderson and center of mass adaptation schemes converge near first order between the two coarsest grids then exhibit a superconvergent property between the 65x33 node and the 257x129 node grids. On the 33x17 node and 65x33 node grid levels, the discretization error present on the adapted grids is less than the next finest uniform grid level. While significant computational savings are not really gained in this case, more significant computational savings could be obtained on the finer grids if the superconvergent property is maintained on the adapted grids. To really evaluate the improvement on the 129x65 node and 257x129 node grid levels, the flow solution on the next finest uniform grid would need to be computed.

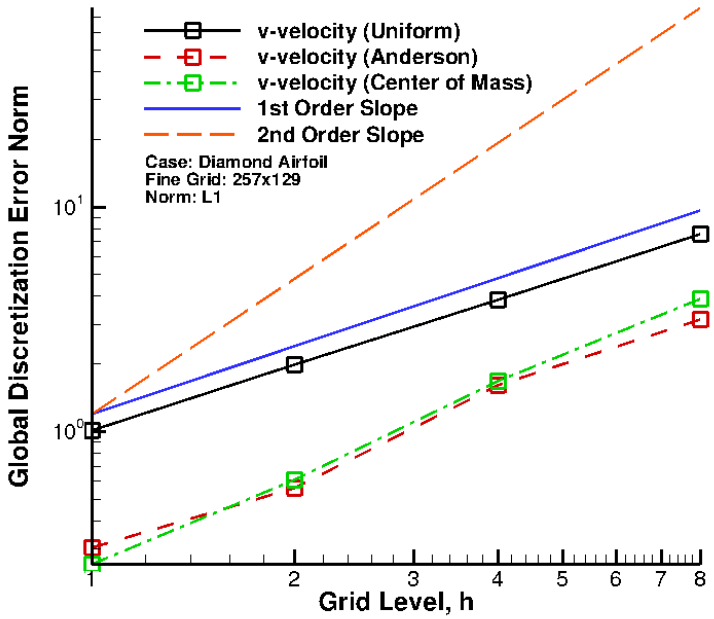


Figure 2.27: Diamond Airfoil: Global Discretization Error - V-Velocity

2.9.5 Karman-Trefftz Airfoil

Case Description

The final case examined is subsonic flow over an airfoil generated by the Karman-Trefftz airfoil transformation [37]. This transformation maps lifting flow over a circular cylinder with circulation to flow over an airfoil. An illustration of this process may be seen in Fig. 2.28. For this case, a series of C-grids ranging in size from a 33x17 node grid to a 257x129 node grid are examined.

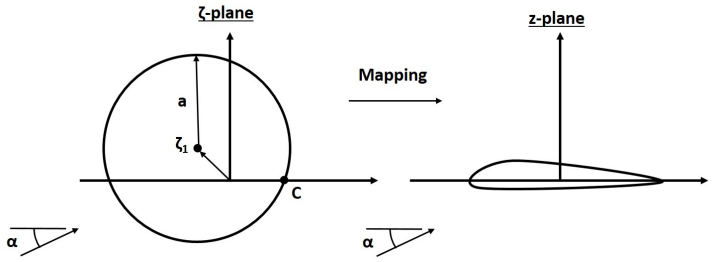


Figure 2.28: Karman-Trefftz Airfoil: Mapping

The freestream conditions are Mach 0.2 flow at a 2° angle of attack with a static pressure of $p_{\infty} = 100\ kPa$ and a static temperature of $T_{\infty} = 347\ K$. The boundary conditions for this case can be found in Fig. 2.29a. For the farfield boundary condition, the ghost cells are set to the freestream conditions and are not altered during the solution process. It is important to note here that it is assumed the outer boundary of the grid is far enough away from the airfoil that the flow is undisturbed by the presence of the airfoil. For the periodic boundary condition, the ghost cells at each iteration of the flow solver are updated with the solution from the adjacent boundary. This boundary condition effectively treats the boundary cells as interior cells.

The exact solution for the Karman-Trefftz airfoil can be seen in Fig. 2.29b. The exact solution is first computed by solving for the flow over a circular cylinder with circulation in the complex ζ-plane. The complex velocity, $\tilde{W}(\zeta) = \tilde{u} - i\tilde{v}$, for flow over a circular cylinder with circulation is given by the following

$$\tilde{W}(\zeta) = V_{\infty}e^{-i\alpha} - \frac{V_{\infty}a^2e^{i\alpha}}{(\zeta - \zeta_1)^2} - \frac{i\Gamma}{2\pi(\zeta - \zeta_1)} \tag{2.69}$$

where α is the angle of attack, a is the radius of the cylinder, ζ_1 is the location of the center of the cylinder, and Γ is the circulation. The flow is then mapped to flow over an airfoil in the complex z-plane using the following mapping

$$\frac{z - nC}{z + nC} = \frac{(\zeta - C)^n}{(\zeta + C)^n} \tag{2.70}$$

where C is a mapping constant and n is given by

$$n = 2 - \frac{\tau}{\pi} \ . \tag{2.71}$$

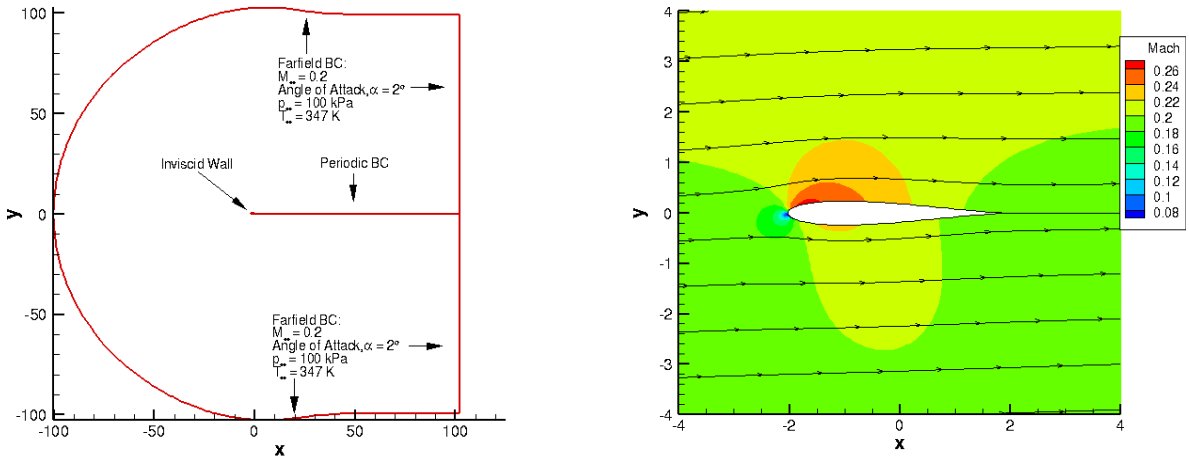
The parameter τ in Eq. 2.71 is a prescribed trailing edge angle for the airfoil given in radians. The geometric parameters for the Karman-Trefftz airfoil used in this study are given in Table 2.9.

Table 2.9: Karman-Trefftz Airfoil: Geometry

Trailing Edge Angle, τ	0.1745°
Mapping Constant, C	1.0
Center of Circle, ζ_1	-0.1 + i 0.0

It is important to note that while this exact incompressible flow solution exactly satisfies conservation of mass and momentum, the assumption of constant density results in a small distributed source term for the total energy equation such that the temperature variations exactly balance the pressure variations to produce a constant density field. This source term is similar to a manufactured solution source term used for code verification [10].

Numerical solutions for this case are obtained using a fully upwinded MUSCL scheme with Van Leer's flux vector splitting scheme [29]. Stability of the second order reconstruction is attained using the van Albada limiter [35].



(a) Boundary Conditions (b) Exact Solution - Mach Number

Figure 2.29: Karman-Trefftz Airfoil: Case Description

Results

Truncation and discretization error improvements for the Karman-Trefftz airfoil case may be found in Table 2.10 for Anderson's adaptation scheme and in Table 2.11 for the center of mass adaptation scheme. The L_2 norm of truncation error and discretization error are reported for a series of grids ranging from a 33x17 node grid to a 257x129 node grid. For this case, the norms of truncation error error show minimal improvements if any for both adaptation schemes. Anderson's adaptation scheme is only able to attain modest improvements for the x-momentum and y-momentum equations. For the continuity equation, truncation error improvements are never obtained. Center of mass does not show improvements until the two finest grids, and even so, the maximum improvement is only a 2x improvement on the finest grid. But, while norms of truncation error do not show significant improvements, substantial improvements in the norms discretization error are achieved on the two finest grid levels for both adaptation schemes. For the 257x129 node grid, the maximum improvement in the L_2 norm of discretization error obtained by Anderson's adaptation scheme is a 7x improvement in density discretization error while the maximum improvement obtained by center of mass is a 8x improvement in u-velocity discretization error.

Table 2.10: Karman-Trefftz Airfoil: Improvement of L_2 Norm of Error

	Adaptation Scheme: Anderson							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
[a]257 x 129 (1)	0.18	1.68	1.07	0.20	7.10	6.70	5.00	4.26
129 x 65 (2)	0.82	3.27	2.76	0.93	2.46	5.76	4.74	2.34
65 x 33 (4)	0.86	2.09	1.50	1.47	1.05	1.61	1.54	0.71
33 x 17 (8)	0.37	0.89	0.62	0.89	0.26	0.42	0.50	0.18

[a]Not fully converged. Converged to 1E-5 (X-Mtm) and 1E-6 (Mass, Y-Mtm, Energy)

Table 2.11: Karman-Trefftz Airfoil: Improvement of L_2 Norm of Error

	Adaptation Scheme: Center of Mass							
	Truncation Error Improvement				Discretization Error Improvement			
Grid Level (h)	Mass	X-mtm	Y-mtm	Energy	Density	U-Velocity	V-Velocity	Pressure
[a]257 x 129 (1)	1.43	2.13	2.09	1.55	4.08	8.17	4.87	5.45
129 x 65 (2)	1.02	1.15	1.01	1.23	2.82	3.34	3.17	2.08
65 x 33 (4)	0.68	0.87	0.48	1.15	0.70	0.95	1.53	0.69
33 x 17 (8)	0.43	0.70	0.50	0.42	0.34	0.45	0.62	0.49

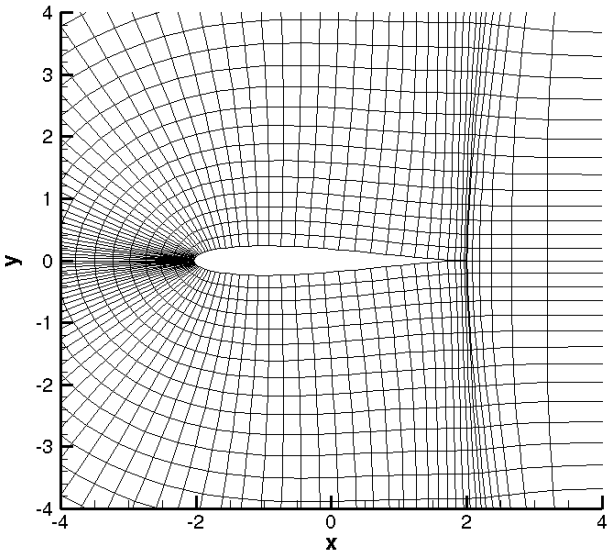
[a]Not fully converged. Converged to 1E-8 (Mass,X-Mtm) and 1E-9 (Y-Mtm,Energy)

While truncation error improvements can often be good indicators that discretization error improvements will be achieved, the results for this case demonstrate that improvements in a

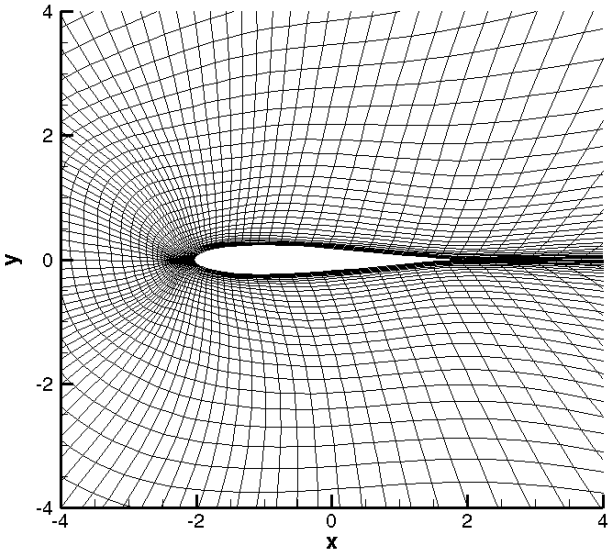
norm of truncation error are not always required to obtain discretization error improvements. For instance, Anderson's adaptation scheme increases mass truncation error by a factor of five for the finest grid but reduced density discretization error by a factor of seven. To better explain why this is the case, it is necessary to examine the initial and adapted grids for this case given in Fig. 2.30 and Fig. 2.31 for the 129x65 node grid. As can be seen in Fig. 2.30a, Fig. 2.31a, and Fig. 2.31b, the initial grid spacing off the surface of the airfoil is relatively large. These coarse cells do not provide adequate resolution of the flowfield. As the adaptation schemes relocate nodes closer to the airfoil as can be seen in Fig. 2.31c and Fig. 2.31e, the flowfield becomes more resolved, especially around the leading edge of the airfoil where large gradients in the solution are present. Since truncation error is not only a function of the grid spacing but also the solution gradients, it is possible that the increased resolution obtained with the adapted grids in high gradient regions effectively uncovers truncation error that would not necessarily be present if the grid were to remain coarse thus giving the appearance that truncation error increases as the grid adapts.

Unlike the previous test cases, the Anderson and center of mass adaptation schemes produce qualitatively different adapted grids for this case. Anderson's adaptation scheme not only provides more resolution at the leading edge, but also provides more regularity in the grid in the trailing edge region. Center of mass, as seen in Fig. 2.31f, struggles in the wake region immediately downstream of the trailing edge. This behavior is due to the current implementation of the boundary condition. Presently, the boundary condition is computed on both sides of the wake cut then points on one side of the wake cut are relocated along the boundary to ensure point matching. In order to fix this issue, the requirement of point-to-point matchup in the wake region would need to be relaxed. For this case, center of mass was unable to produce significant improvements using the scheme consistent boundary condition. Therefore, a Neumann boundary condition was used instead. During the adaptation process, points were allowed to freely move from the wake region up onto the airfoil and vice versa. Measures were taken to not only maintain the trailing edge location but also to maintain the same number of points on both sides of the wake cut.

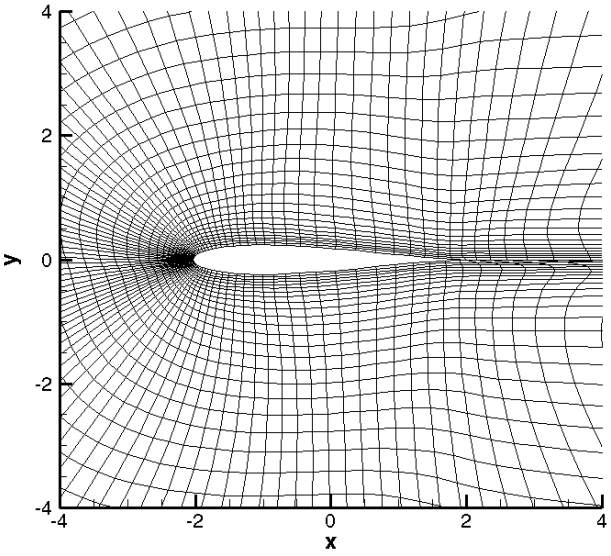
Truncation error and discretization error plotted across the computational domain may be seen in Fig. 2.32 and Fig. 2.33. Truncation error is plotted for the x-momentum equation while discretization error is plotted for u-velocity. The majority of x-momentum truncation error is generated at the leading edge where the grid has a large curvature and u-velocity has a large gradient caused by the deceleration of the flow at the stagnation point. Based upon Fig. 2.33a and Fig. 2.33b, the majority of the discretization error is generated at the leading edge and is convected downstream along the airfoil surface. Anderson's adaptation scheme, given in Fig. 2.33c and Fig. 2.33d, is able to provide enough resolution around the surface of the airfoil to significantly reduce the discretization error. Center of mass, while not providing as much resolution as the Anderson adaptation scheme, is still able to improve the solution to some extent. Also, the struggles encountered in the wake region did not seem to have a significant impact on the discretization error for this scheme.



(a) Uniform

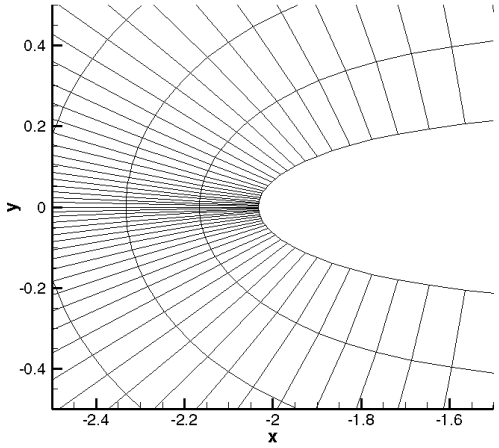


(b) Anderson

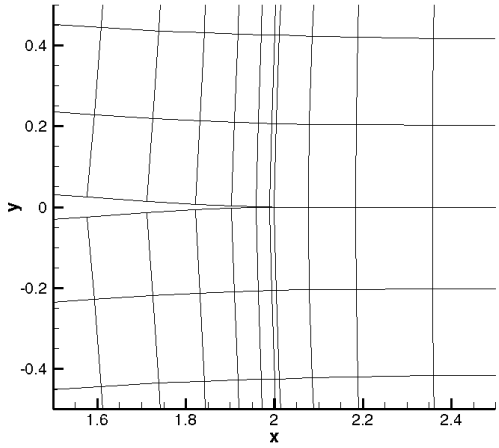


(c) Center of Mass

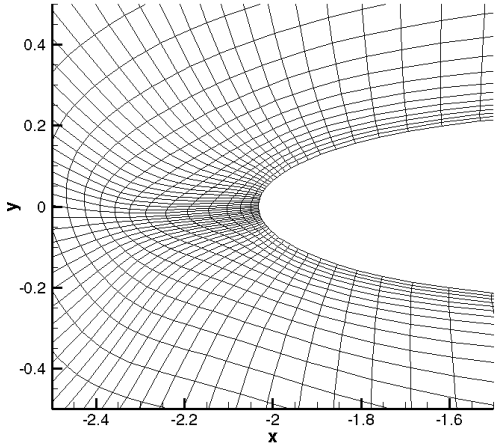
Figure 2.30: Karman-Trefftz Airfoil: Grids (Grid Level: 129x65)



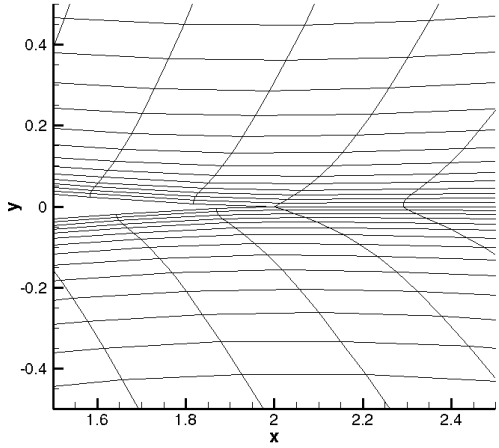
(a) Uniform: Leading Edge



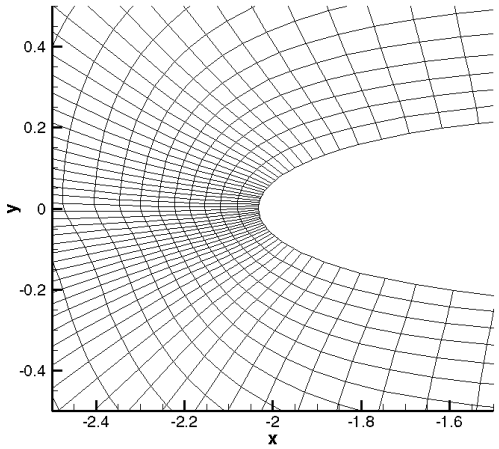
(b) Uniform: Trailing Edge



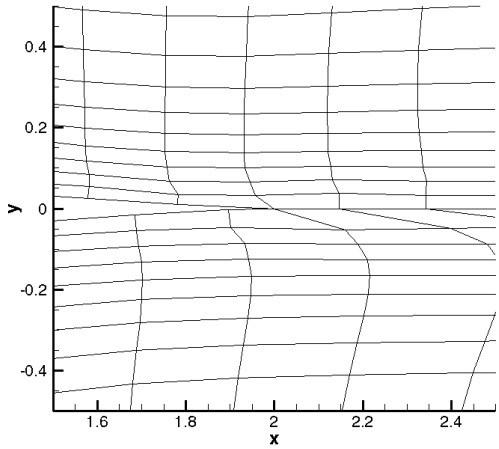
(c) Anderson: Leading Edge



(d) Anderson: Trailing Edge

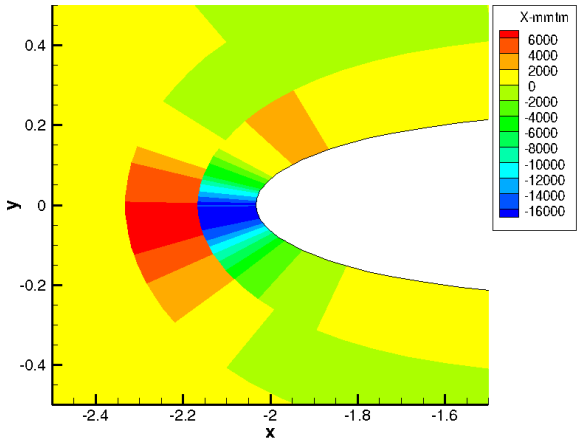


(e) Center of Mass: Leading Edge

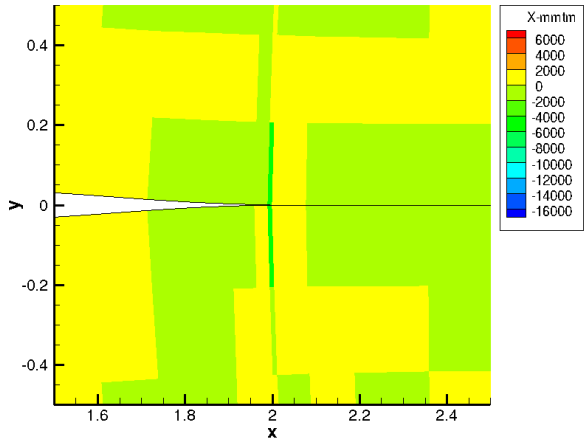


(f) Center of Mass: Trailing Edge

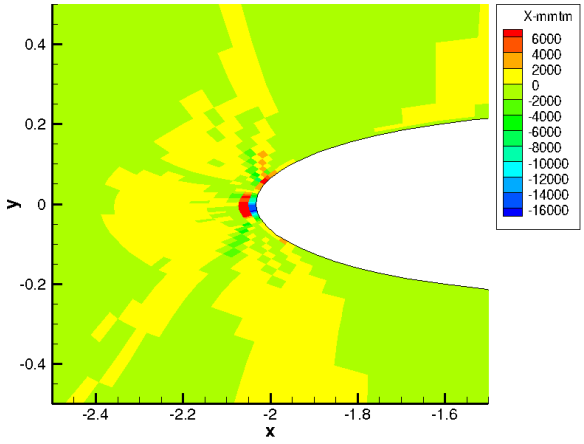
Figure 2.31: Karman-Trefftz Airfoil: Grids - Zoomed (Grid Level: 129x65)



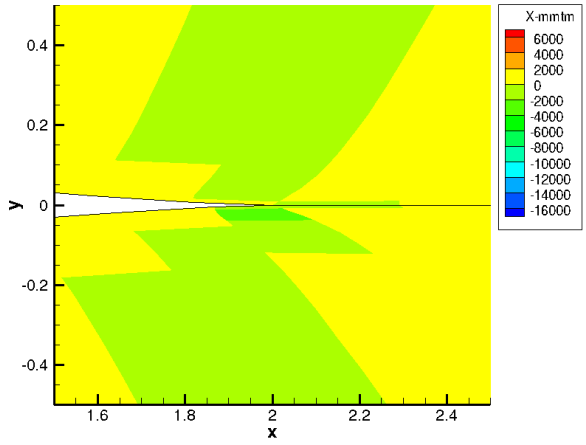
(a) Uniform: Leading Edge



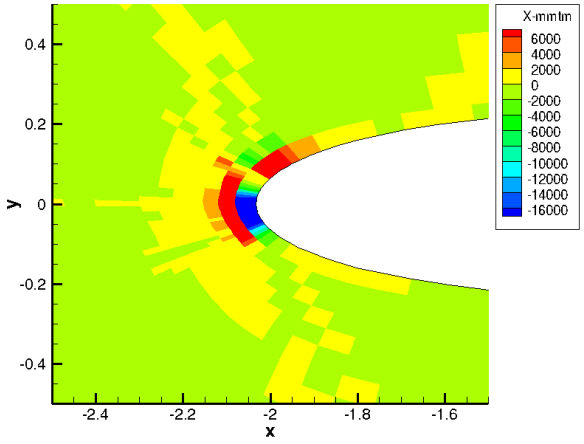
(b) Uniform: Trailing Edge



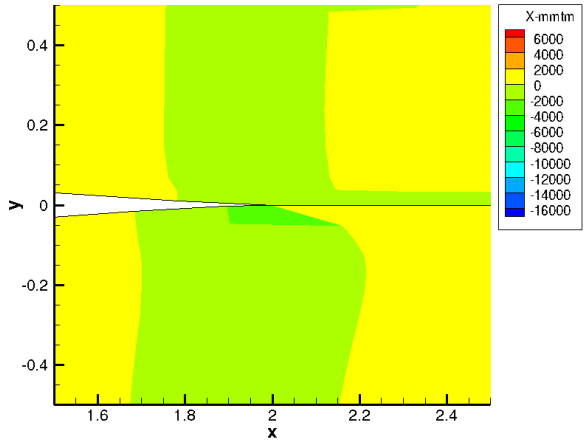
(c) Anderson: Leading Edge



(d) Anderson: Trailing Edge

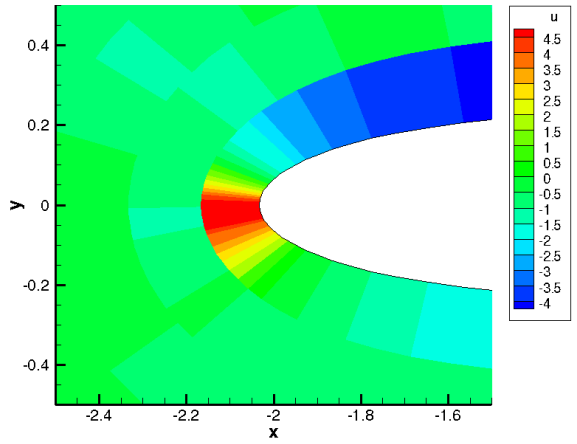


(e) Center of Mass: Leading Edge

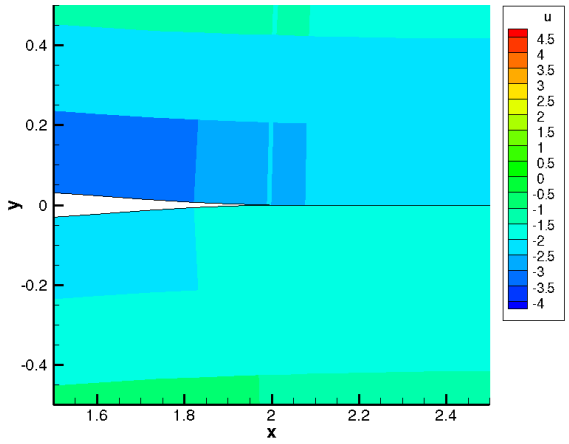


(f) Center of Mass: Trailing Edge

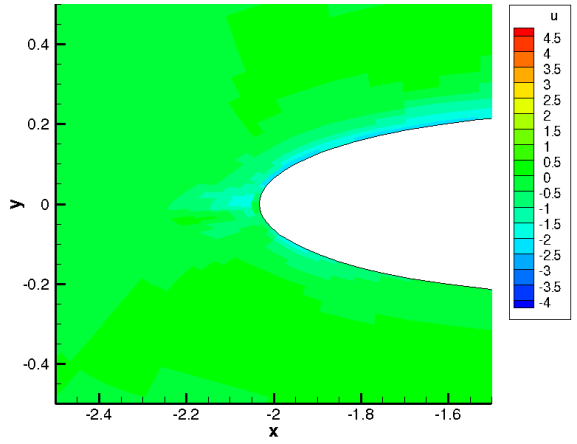
Figure 2.32: Karman-Trefftz Airfoil: TE X-Momentum (Grid Level: 129x65)



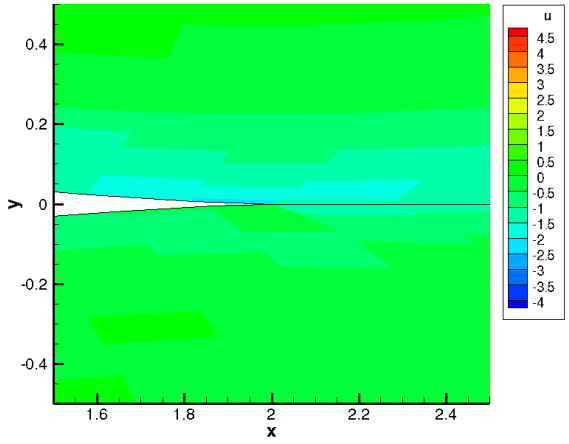
(a) Uniform: Leading Edge



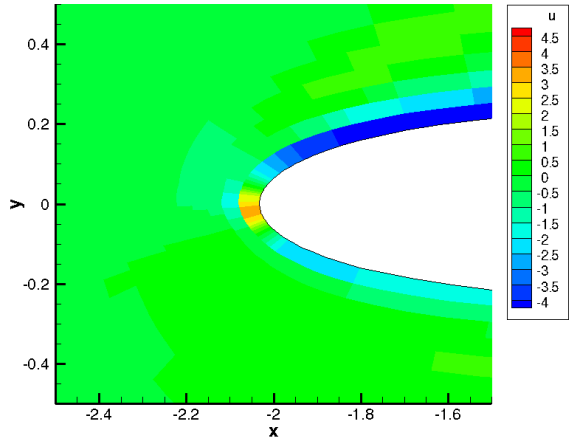
(b) Uniform: Trailing Edge



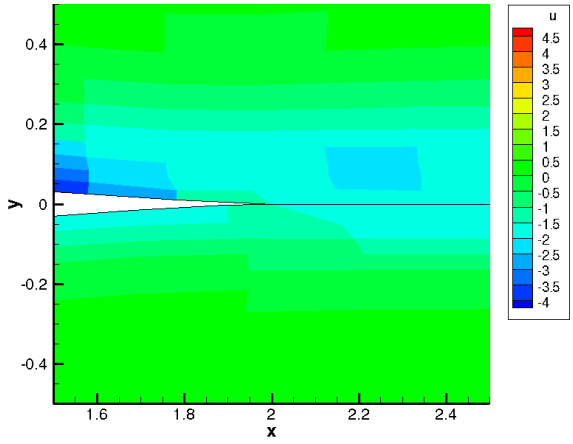
(c) Anderson: Leading Edge



(d) Anderson: Trailing Edge



(e) Center of Mass: Leading Edge



(f) Center of Mass: Trailing Edge

Figure 2.33: Karman-Trefftz Airfoil: DE U-Velocity (Grid Level: 129x65)

The convergence of the L_2 norm of u-velocity discretization error for the initial and adapted grids is given in Fig. 2.34. The discretization error on the uniform grids does not begin to decrease until after the 129x65 node grid level. This is likely because the asymptotic range is not achieved until the finest grid level for this primitive variable. On the other hand, it should be noted that the remaining primitive variables reach the asymptotic range on the second finest grid rather than the finest. On the finest grid, the discretization error appears to be converging at a first order rate. The discretization error norms for the adapted grids seem to be converging on average at approximately a second order rate. But, the discretization error on the coarsest grid has increased relative to the uniform grid. This can be attributed to the lack of resolution of the flowfield on this grid. Since the grid is so coarse, the adaptation schemes are unable to detect that points should be relocated toward the surface of the airfoil.

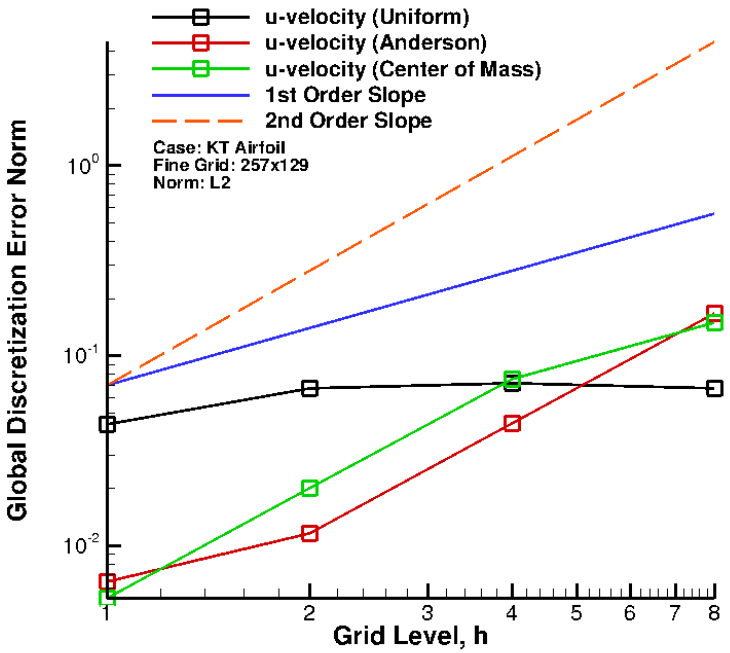


Figure 2.34: Karman-Trefftz Airfoil: Global Discretization Error - U-Velocity

2.10 Conclusions

Truncation error is shown to be the local source of discretization error in numerical solutions and is used to drive grid adaptation. Several r-adaptation schemes are reviewed and applied within the framework of truncation error based adaptation, including an adaptive Poisson grid generator, a variational grid generator, a center of mass based scheme, and a scheme based on deforming maps. Boundary conditions for each adaptation scheme are discussed and a framework for preserving geometry information is presented. In order to drive adaptation, truncation error is formulated as a weight function by averaging the truncation error from each governing equation normalized by its L_2 norm on the initial grid. The adaptation process is monitored using a truncation error based metric, the weight function equidistribution error, and a grid movement metric. The effectiveness of each adaptation scheme in reducing discretization error is evaluated by applying the adaptation schemes to supersonic flow around a 12° downward turn, supersonic flow around a 10° compression turn, supersonic flow past a diamond-shaped airfoil, and subsonic flow over a Karman-Trefftz airfoil. For the 2-D expansion fan, it is found that over a six time reduction in discretization error can be achieved on adapted grids. The adaptation achieved its greatest improvements in discretization error for supersonic flow around a compression turn by reducing norms of discretization error by eleven times. The diamond-shaped airfoil case attained a modest four time improvement in discretization error while the Karman-Trefftz airfoil case reduced discretization error by eight times on the adapted grids. Also, discretization error on a set of independently adapted grids is computed and convergence rates are compared to rates of discretization error convergence on uniform grids of the same grid level. For the 2-D expansion fan and the 2-D compression turn cases, it is found that coarse, adapted grids are able to achieve the same level of discretization error as a uniform grid approximately three grid levels finer. Of the adaptation schemes used, Anderson's adaptation scheme [15] and the center of mass scheme [19] performed best, achieving the largest amounts of discretization error reduction for the test cases examined.

In this study, exact truncation error is used to drive adaptation. Future work will need to be conducted to investigate the use of estimated truncation error for more practical applications. Since the adaptation techniques reviewed here are primarily implemented for single block grids, future work will also need to include application of these adaptation schemes to multi-block grids so that more complex geometries may be tested.

2.11 Bibliography

[1] Roy, C., "Strategies for Driving Mesh Adaptation in CFD (Invited)," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Jan 2009.

[2] Warren, G., Anderson, W., Thomas, J., and Krist, S., "Grid convergence for adaptive methods," *10th Computational Fluid Dynamics Conference*, Jun 1991.

[3] Ainsworth, M. and Oden, J., *A Posteriori Error Estimation in Finite Element Analysis*, Wiley Interscience, New York, 2000.

[4] Zhang, X., Trepanier, J.-Y., and Camarero, R., "A posteriori error estimation for finite-volume solutions of hyperbolic conservation laws," *Computer Methods in Applied Mechanics and Engineering*, Vol. 185, No. 1, Apr 2000, pp. 1 – 19.

[5] Gu, X. and Shih, T., "Differentiating between source and location of error for solution-adaptive mesh refinement," *15th AIAA Computational Fluid Dynamics Conference*, Jun 2001.

[6] Brackbill, J. and Saltzman, J., "Adaptive zoning for singular problems in two dimensions," *Journal of Computational Physics*, Vol. 46, No. 3, Jun 1982, pp. 342 – 368.

[7] McRae, D., "r-Refinement grid adaptation algorithms and issues," *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, No. 4, Sep 2000, pp. 1161 – 1182.

[8] Baker, T. J., "Mesh adaptation strategies for problems in fluid dynamics," *Finite Elements in Analysis and Design*, Vol. 25, No. 3-4, Apr 1997, pp. 243 – 273.

[9] Hawken, D., Gottlieb, J., and Hansen, J., "Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations," *Journal of Computational Physics*, Vol. 95, No. 2, Aug 1991, pp. 254 – 302.

[10] Oberkampf, W. L. and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press (CUP), 2010.

[11] Choudhary, A. and Roy, C., "Efficient Residual-Based Mesh Adaptation for 1D and 2D CFD Applications," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Jan 2011.

[12] Phillips, T., Derlaga, J. M., Roy, C. J., and Borggaard, J., "Finite Volume Solution Reconstruction Methods For Truncation Error Estimation," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

[13] Phillips, T. and Roy, C., "Residual Methods for Discretization Error Estimation," *20th AIAA Computational Fluid Dynamics Conference*, Jun 2011.

[14] Thompson, J., Warsi, Z., and Mastin, C., *Numerical Grid Generation: Foundations and Applications*, Elsevier Science Publishing Co., Inc., 1997.

[15] Anderson, D. A., "Grid cell volume control with an adaptive grid generator," *Applied Mathematics and Computation*, Vol. 35, No. 3, Feb 1990, pp. 209 – 217.

[16] Winslow, A., "Numerical Solution of the Quasi-Linear Poisson Equation," *Journal of Computational Physics*, Vol. 1, 1966, pp. pp. 149 – 172.

[17] Brackbill, J., "An Adaptive Grid with Directional Control," *Journal of Computational Physics*, Vol. 108, No. 1, Sep 1993, pp. 38 – 50.

[18] Huang, W. and Russell, R., *Adaptive Moving Mesh Methods*, Springer, 2011.

[19] Laflin, K., *Solver-Independent r-Refinement Adaptation for Dynamic Numerical Simulations*, Ph.D. thesis, North Carolina State University, 1997.

[20] Benson, R. and McRae, D., "A Solution-Adaptive Mesh Algorithm for Dynamic/Static Refinement of Two and Three Dimensional Grids," *Proceedings of the Third International Conference on Numerical Grid Generation in Computational Field Simulations*, Barcelona, Spain, June 1991, pp. 185 – 199.

[21] Liao, G. and Anderson, D., "A new approach to grid generation," *Applicable Analysis*, Vol. 44, No. 3-4, Apr 1992, pp. 285 – 298.

[22] Moser, J., "On the Volume Elements on a Manifold," *Transactions of the American Mathematical Society*, Vol. 120, No. 2, Nov 1965, pp. 286 – 294.

[23] Grisham, J. R., Vijayakumar, N., Liao, G., Dennis, B. H., and Lu, F. K., "A Comparison Between Local h-Refinement and a Novel r-Refinement Method," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[24] Zhang, Y., Jia, Y., Wang, S. S., and Chan, H., "Boundary treatment for 2D elliptic mesh generation in complex geometries," *Journal of Computational Physics*, Vol. 227, No. 16, Aug 2008, pp. 7977 – 7997.

[25] Fritsch, F. N. and Carlson, R. E., "Monotone Piecewise Cubic Interpolation," *SIAM J. Numer. Anal.*, Vol. 17, No. 2, Apr 1980, pp. 238 – 246.

[26] Choudhary, A., *Verification of Compressible and Incompressible Computational Fluid Dynamics Codes and Residual-based Mesh Adaptation*, Ph.D. thesis, Virginia Polytechnic Institute and State University, November 2014.

[27] Derlaga, J. M., Phillips, T., and Roy, C. J., "SENSEI Computational Fluid Dynamics Code: A Case Study in Modern Fortran Software Development," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

[28] Roe, P., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 135, No. 2, Aug 1997, pp. 250 – 258.

[29] van Leer, B., "Flux-vector splitting for the Euler equations," *Lecture Notes in Physics*, 1982, pp. 507 – 512.

[30] van Leer, B., "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunovs method," *Journal of Computational Physics*, Vol. 32, No. 1, Jul 1979, pp. 101 – 136.

[31] Beam, R. M. and Warming, R., "An implicit finite-difference algorithm for hyperbolic systems in conservation-law form," *Journal of Computational Physics*, Vol. 22, No. 1, Sep 1976, pp. 87 – 110.

[32] Choudhary, A. and Roy, C. J., "Structured Mesh r-Refinement using Truncation Error Equidistribution for 1D and 2D Euler Problems," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

[33] Staff, "Equations, Tables, and Charts for Compressible Flow," Tech. Rep. TR 1135, National Advisory Committee for Aeronautics, Moffett Field, CA, 1953.

[34] Anderson, J. D., *Modern Compressible Flow with Historical Perspective*, McGraw-Hill, New York, New York, 2003.

[35] van Albada, G. D., van Leer, B., and Roberts, W. W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Upwind and High-Resolution Schemes*, 1997, pp. 95 – 103.

[36] Banks, J., Aslam, T., and Rider, W., "On sub-linear convergence for linearly degenerate waves in capturing schemes," *Journal of Computational Physics*, Vol. 227, No. 14, Jul 2008, pp. 6985 – 7002.

[37] Karamcheti, K., *Principles of Ideal-Fluid Aerodynamics*, Krieger Publishing Companion, 1966.

Chapter 3

Discussion and Conclusions

While discretization error will always be present in numerical solutions, grid adaptation provides an efficient mechanism for improving solution accuracy. In this work, truncation error is shown to be the local source for discretization error and an appropriate driver for grid adaptation. Multiple r-adaptation schemes, including an adaptive Poisson grid generator, a variational grid generator, a center of mass based scheme, and a scheme based on deforming maps, are reviewed and implemented within an existing structured adaptation module (SAM) [1]. This adaptation module is directly integrated into the SENSEI code based [2] to form an efficient, compact flow solver/adaptation package. Adaptation boundary conditions are outlined, and a framework for preserving geometry information is presented. Exact truncation error is successfully formulated into a weight function and is used to drive adaptation. Convergence of the adaptation process is monitored with variety of metrics. The adaptation schemes are applied to supersonic flow around a 12° downward turn, supersonic flow around a 10° compression turn, supersonic flow over a diamond-shaped airfoil, and subsonic flow past a Karman-Trefftz airfoil. Truncation error and discretization error improvements for each case are evaluated using known exact solutions. Relative to the initial grids for each case, substantial improvements in spatial discretization error are achieved with a maximum improvement of over an order of magnitude obtained for the 10° compression turn case. Discretization error convergence is also studied on a series of uniform and adapted grids for each case. These grid studies illustrated the tremendous computational savings that can be achieved with grid adaptation especially for problems with linear or sub-linear convergence due to the presence of one or more discontinuities. In some cases, coarse adapted grids achieved a level of discretization error as low or lower than a uniform grid three grid levels finer. Also, it is found that while truncation error norm improvements can often be a good indication that discretization error improvements will be achieved, they are not always required. For the Karman-Trefftz airfoil case, mass truncation error increased by a factor of five but a density discretization error improvement of seven times is still achieved. The flow solution as well as the initial state of the grid should also be taken into account. Furthermore, it is found that grid alignment with discontinuous flow features can have a

significant impact on the amount of discretization error improvement that can be obtained. For supersonic flow over the 10° compression ramp, the center of mass adaptation scheme achieved better alignment with the shock wave and reduced discretization error nearly three times better than the Anderson adaptation scheme. The center of mass and Anderson adaptation schemes produced qualitatively very similar grids for the remaining test cases likely due to the fact that the center of mass adaptation scheme can be shown to be equivalent to a discrete Poisson equation [3]. Of all the adaptation schemes examined in this work, the center of mass and Anderson adaptation schemes performed best for reducing discretization error.

3.1 Future Work

For this work, exact truncation error was used to drive adaptation. Since exact solutions are rarely known for practical applications, estimated truncation will be required to drive adaptation. Truncation error estimation methods are available [4] but require the flow solution to be relatively converged. Each time the grid is adapted and the iterative residuals increase for lack of a conservative interpolation between grids, the solution will need to be reconverged in order to obtain an accurate truncation error estimate. In order to improve the efficiency of the adaptive process while using estimated truncation error, the question of how far iterative residuals need to be converged to accurately estimate truncation error will need to be answered.

While h-adaptation schemes are easily implemented and perform well for complex geometries, r-adaptation algorithms are attractive for adaptation because the number of degrees of freedom does not increase. Therefore, the computation time required for the entire adaptation/flow solution process is arguably less than alternative methods of adaptation such as h-adaptation. This claim needs to be rigorously tested by monitoring total computation time in order for r-adaptation schemes to be competitive with h-adaptation schemes. To this end, an efficient multi-block implementation of these adaptation schemes will also need to be put into place. This will require relaxing the requirement for point-to-point matchup at block interfaces.

During this study, under-relaxation was necessary to prevent grid crossover for every test case. Even though under-relaxation can help prevent crossover, a great deal of user intervention is still required monitor and ensure crossover does not occur. Methods will need to be implemented for adaptive under-relaxation when crossover is detected. These preventative measures coupled with new adaptation schemes designed to never allow crossover can significantly reduce the amount of user intervention required to operate an r-adaptation scheme.

Lastly, the framework for boundary geometry preservation presented in this work proved successful for retaining the geometric integrity of the boundary. To maintain the geometry,

the proposed method fit boundary points of the initial grid with a piecewise cubic Hermite interpolating polynomial. This works extremely well for moderately fine to very fine grids. In order to properly maintain boundary geometry for coarse grids, the method will need to be updated to supply adequate boundary geometry information such as using the boundary points from a fine grid to define the curve fit for the coarse grid.

3.2 Bibliography

[1] Choudhary, A. and Roy, C. J., "Structured Mesh r-Refinement using Truncation Error Equidistribution for 1D and 2D Euler Problems," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

[2] Derlaga, J. M., Phillips, T., and Roy, C. J., "SENSEI Computational Fluid Dynamics Code: A Case Study in Modern Fortran Software Development," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

[3] Laflin, K., *Solver-Independent r-Refinement Adaptation for Dynamic Numerical Simulations*, Ph.D. thesis, North Carolina State University, 1997.

[4] Phillips, T., Derlaga, J. M., Roy, C. J., and Borggaard, J., "Finite Volume Solution Reconstruction Methods For Truncation Error Estimation," *21st AIAA Computational Fluid Dynamics Conference*, Jun 2013.

Appendix A

Coefficients of Variational Governing Equations

A.1 Smoothness

$$A = x_\xi y_\xi + x_\eta y_\eta \qquad \alpha = \frac{x_\eta^2 + y_\eta^2}{J^3} \qquad a_{s1} = -A\alpha \qquad b_{s1} = B\alpha \qquad c_{s1} = C\alpha$$
$$B = y_\xi^2 + y_\eta^2 \qquad \beta = \frac{x_\xi x_\eta + y_\xi y_\eta}{J^3} \qquad a_{s2} = 2A\beta \qquad b_{s2} = -2B\beta \qquad c_{s2} = -2C\beta$$
$$C = x_\xi^2 + x_\eta^2 \qquad \gamma = \frac{x_\xi^2 + y_\xi^2}{J^3} \qquad a_{s3} = -A\gamma \qquad b_{s3} = B\gamma \qquad c_{s3} = C\gamma$$

A.2 Orthogonality

$$a_{o1} = x_\eta y_\eta \qquad b_{o1} = x_\eta^2 \qquad c_{o1} = y_\eta^2$$
$$a_{o2} = x_\xi y_\eta + x_\eta y_\xi \qquad b_{o2} = 2(2x_\xi x_\eta + y_\xi y_\eta) \qquad c_{o2} = 2(x_\xi x_\eta + 2y_\xi y_\eta)$$
$$a_{o3} = x_\xi y_\xi \qquad b_{o3} = x_\xi^2 \qquad c_{o3} = y_\xi^2$$

A.3 Weighted Cell Volume Variation

$$a_{v1} = -x_\eta y_\eta \qquad b_{v1} = y_\eta^2 \qquad c_{v1} = x_\eta^2$$
$$a_{v2} = x_\xi y_\eta + x_\eta y_\xi \qquad b_{v2} = -2y_\xi y_\eta \qquad c_{v2} = -2x_\xi x_\eta$$
$$a_{v3} = -x_\xi y_\xi \qquad b_{v3} = y_\xi^2 \qquad c_{v3} = x_\xi^2$$