ANALYSIS AND IMPLEMENTATION OF A HIGH-ORDER RECONSTRUCTION ALGORITHM FOR AN UNSTRUCTURED FINITE VOLUME FLOW SOLVER

By

Shane Edmond Sawyer

Approved:

Kidambi Sreenivas Research Professor of Computational Engineering (Chair)	Steve L. Karman, Jr. Professor of Computational Engineering (Committee Member)
Daniel G. Hyams Associate Professor of Computational Engineering (Committee Member)	
William H. Sutton Dean of the College of Engineering and Computer Science	A. Jerald Ainsworth Dean of the Graduate School

ANALYSIS AND IMPLEMENTATION OF A HIGH-ORDER RECONSTRUCTION ALGORITHM FOR AN UNSTRUCTURED FINITE VOLUME FLOW SOLVER

By

Shane Edmond Sawyer

A Thesis
Submitted to the faculty of
The University of Tennessee, Chattanooga
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Engineering and Computer Science

The University of Tennessee, Chattanooga
Chattanooga, Tennessee

August 2012

ABSTRACT

A high-order scheme is examined and implemented in an unstructured solver. The motivation for this research is driven by research goals to simulate field equations, particularly those of fluid dynamics, with high fidelity. High-order schemes overcome computational limitations by computing comparable solutions on grids that are coarser than grids required by a second-order flow solver.

The scheme was chosen based on two criteria. The first being that it is well documented in the literature for two-dimensional flow solvers. The second is that the scheme is extendable to the framework used in the Tenasi flow solver developed at the University of Tennessee at Chattanooga SimCenter: National Center for Computational Engineering.

The accuracy of the scheme is demonstrated in the following ways: reconstruction of variables from a smooth function, verification by the method of manufactured solutions, and quantitative and qualitative results from the solution of compressible, inviscid flows.

DEDICATION

This thesis is dedicated to my parents for encouraging me to follow my dreams and to my wife Paula for standing by me along the way.

ACKNOWLEDGEMENTS

I would like to give a special thanks Dr. Kidambi Sreenivas for all the support and guidance given to me while undertaking my research. I would also like to thank Dr. Daniel Hyams, Dr. Steve Karman, Dr. Roger Briley, Dr. Matt Matthews, Dr. Kyle Anderson, Dr. Li Wang, and Dr. Tim Swafford for their support. And, a final thanks to my fellow graduate students Ryan, Vince, Louie, and Justin (to name a few) who have shared the frustration and joy of research and lived to pass a little help my way.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Q, Vector of conserved variables

F, Flux vector

CV, Control volume

CS, Surface of control volume

V, Volume

A, Area

u, v, w, Cartesian velocity components

P, Pressure

E, Total energy

ρ, Fluid density

γ, Ratio of specific heats

c, Speed of sound

C_P, Coefficient of pressure

$\hat{n}$, Normal vector

$\vec{u}$, Cartesian velocity vector

x, y, z, Subscripts to denote direction or derivative

∞, Subscript to denote freestream

α, Angle of attack

CHAPTER 1
INTRODUCTION

The continuing advances made in computer hardware development, in both performance and in affordability, have spurred the growth in many areas of computational engineering and science. Computational fluid dynamics (CFD) in particular has benefited from the rapid pace set by computer hardware. CFD has matured through decades of research from its beginnings in academia to become a major component of industrial design. It allows for relatively rapid testing of design models, saving money by reducing the number of physical models that need to be analyzed by conventional experimental means. The further development of sophisticated numerical algorithms coupled with the proliferation of computing resources ensure that CFD will remain a fixture in both academia and industry.

Motivation and Background

Unstructured grid finite volume methods have been applied to flow simulations for many years due to 1) the method's ability to handle complex geometries and 2) reduced grid generation time. The accuracy of most unstructured finite volume methods has been limited to second-order, unlike structured finite volume methods which have benefited from high-order accurate schemes such as the essentially non-oscillatory (ENO) [1] and the weighted essentially non-oscillatory (WENO) [2] methods for many years. ENO methods work by building the smoothest stencil possible from the available neighbors to reconstruct the solution. The stencil search can vary in time which may impede the convergence to the steady-state solution particularly when the stencil varies every iteration. The WENO scheme of Liu, Osher, and Chan [2] overcomes this deficiency by using all possible stencils for reconstruction. The stencils are weighted based on a smoothness criteria that penalizes non-smooth stencils with a weight on the order of the truncation error.

The development of high-order accurate schemes for unstructured grids has been stunted by challenges introduced in the discretization of the field equations on the mesh. Specifically, the loss of element regularity on unstructured grids makes the construction of a stencil with the proper characteristics a more daunting challenge. In recent years, two methods in particular have been introduced that offer promising results to achieve high-order accuracy for the equations of fluid dynamics. The first method is the Discontinuous Galerkin (DG) method. This method was first introduced by Reed and Hill [3] to solve the neutron transport problem. DG is a finite-element method that uses the differential form of the equations and applies test functions to generate the weak form of the problem. DG differs from traditional finite-element methods in that it does not explicitly enforce solution continuity at element boundaries but uses the numerical flux scheme to couple the solution at the interfaces [4]. As opposed to the high-order schemes used in finite volume methods, DG increases accuracy by a local approximation of the differential operator. This allows the method to retain a compact stencil that is fixed regardless of the order of the local approximation. A drawback to the DG method is that it is typically formulated for simple elements (triangles and quadrilaterals in two dimensions, tetrahedra and hexahedra in three dimensions). Recently, Berggren [4, 5] introduced a new DG method that can be interpreted as a high-order extension to vertex-centered, finite-volume codes.

The second method is a continuation of the pioneering work done by Barth and Frederickson [6] to use k-exact reconstruction to attain high-order accuracy. This research coupled with the existing ideas for structured meshes provided a reference for the extension of ENO schemes onto unstructured meshes. Harten and Chakravarthy [7] provided some of the earliest analysis of these ideas including the structure of the reconstruction operator. In the report, they suggested two approaches to constructing the stencil of neighboring cells for use in the reconstruction procedure. The first is the use of adaptive stencils where a group of stencils is built containing a central stencil and a minimal number of directional stencils; an idea similar to one-sided stencils in structured ENO schemes. From this collection, the stencil that is smoothest with respect to some criteria is selected for use by the reconstruction operator to build the coefficients of a truncated Taylor series. The alternative approach is to find the values of the coefficients term by term by adding neighbors to the stencil that minimize the coefficient value in a normed sense. Later, Abgrall [8] provided implementation details and numerical verification of the scheme up to third order. Friedrich [9] further extended ideas from structured ENO schemes by introducing a weighted scheme for unstructured meshes. In the work, a collection of stencils is generated for each node as in [7]. For each stencil, a reconstruction polynomial, P_i, is found. Each P_i is then assigned an oscillation value which is used to compute an associated non-negative weight ω_i such that $\sum_{i=1}^{m} \omega_i = 1$, where m is the number of stencils. Finally, the reconstruction polynomial associated with the cell is computed as $P = \sum_{i=1}^{m} \omega_i P_i$.

Hu and Shu [10] presented a slightly different formulation in their research. The starting point of the scheme begins with finding the k-exact reconstruction polynomial using a central stencil, as in [6, 7, 9], denoted as p^k. A set of order $k-1$ polynomials is then considered each acting on a stencil that is a subset of the full central stencil. The low-order polynomials are found by constraining them to agree with the control volume average of each cell in its respective stencil. For each quadrature point along the boundary segments of a cell, the reconstruction function is given by $R(x, y) = \sum_{i=1}^{m} \gamma_i p_i(x, y)$, where the p_i's are of degree $k-1$. This function is constrained by $R(x^Q, y^Q) = p^k(x^Q, y^Q)$, where the superscript Q denotes a quadrature point. Note that this indicates that each quadrature point will have unique set of linear weights (the γ_i's). The linear weights may also be replaced with non-linear weights chosen based on a smoothness indicator to make the scheme suitable for flows when discontinuities are present.

Ollivier-Gooch proposed another extension in [11, 12] with the data-dependent weighted least squares (DD-L_2-ENO) reconstruction scheme. In the DD-L_2-ENO method, a fixed stencil is used to generate the reconstruction with data-dependent weights utilized to strongly emphasize smooth data similar to structured WENO methods. Later, Ollivier-Gooch and Van Altena [13] cast the method in a slightly different form for the advection-diffusion problem. This new interpretation differed from the DD-L_2-ENO method in that the weights applied to the stencil were dependent on geometrical spacing rather than solution data. The method of was further developed for application to the Euler equations by Ollivier-Gooch, Nejat, and Michalak [14, 15] and for the Navier-Stokes equations by Ollivier-Gooch and Michalak [16]. Results presented in [14, 16, 17] suggest that it is feasible to obtain high-order accurate solutions to the compressible inviscid and Navier-Stokes equations in a reasonable time frame.

Research Objectives

The main objective of this research is to identify and test an algorithm capable of producing high-order accurate solutions to the compressible, inviscid equations of fluid dynamics. A constraint was imposed to limit the search to algorithms that would be extendable to the framework of the Tenasi flow solver developed by the University of Tennessee at Chattanooga SimCenter: National Center for Computational Engineering (SimCenter). This constraint provided some metrics in selecting the algorithm. It must have a formulation for three-dimensional flow, support control volumes generated by the median dual construction, and be parallelizable without undue penalty. The algorithm of [13], examined in detail in this thesis, was able to satisfy these requirements. The focus of the algorithm is on increasing the order of the polynomial reconstruction used in flux evaluation to increase the order of accuracy of the overall scheme. Compared to the other methods mentioned, the algorithm of Ollivier-Gooch *et al* appeared to be the most straightforward to implement and the most economical (in terms of CPU and memory requirements) since it relied on a single stencil. It also has the added benefit of being a finite volume method and consequently would require less alteration of the code base in Tenasi than implementing a DG, or another finite-element, method.

Outline

In Chapter 2, a discussion of the unstructured flow solver used in evaluating the high-order method is provided. This included details on flux calculations, evaluation of the residual, implicit terms, boundary conditions, and the strategy employed in solving the linear system.

In Chapter 3, the reconstruction scheme implemented in the flow solver is presented and discussed in detail. The results from the scheme's application to smooth test functions are also given.

In Chapter 4, the Method of Manufactured Solutions (MMS) is described and utilized to provide quantitative evidence of the solver's order of accuracy. Results are presented for geometries that include linear and curved boundaries.

In Chapter 5, numerical results are presented for simulation cases which demonstrate the performance of the new scheme in comparison to a base second-order scheme for steady solutions. One case that is used is supersonic flow in an annular geometry that possesses an analytical solution. Both schemes are used to compute a numerical solution. Error between the schemes' numerical solution and the analytical solution are quantified. Subsonic flow over a circular cylinder is also presented where the computed coefficient of pressure is compared with theoretical results. Numerical results are also demonstrated for flow about the NACA 0012 airfoil.

In Chapter 6, numerical results are shown for unsteady flow cases. The first case tested is the simulation of passive vortex transport. Here, an analytical solution is provided that can be used to assess the performance of the new scheme and the base second-order scheme. The final test case provides a qualitative assessment of both schemes' ability to maintain solution structure of shedding vortices as they are transported downstream.

Finally, Chapter 7 ends the thesis with concluding remarks and briefly discusses continuing work not addressed in the previous chapters.

CHAPTER 2
FLOW SOLVER

The principle equations studied in this thesis are the compressible, inviscid equations of Leonhard Euler. As suggested by their name, these equations neglect the influences of viscosity on the flow. While the equations are not adequate to describe all flow phenomena, they do allow for a broad range of flow simulations with interesting characteristics. Since they are easier to solve than the full Navier-Stokes equations, an inviscid solver makes an excellent testbed for new solution methods. Furthermore, a robust Euler solver is an integral component of a full Navier-Stokes solver. The Euler equations are comprised of the conservation equations of mass, momentum, and energy. The integral form of the three-dimensional Euler equations for an arbitrary control volume is given in Equation 2.1. The equations specify that the time rate of change in the vector of conserved quantities within the spatial extent of the control volume must be balanced by the flux through the control volume surface.

$$\frac{\partial}{\partial t}\int_{CV} \vec{Q}\, dV + \oint_{CS} \vec{F} \cdot \hat{n}\, d\vec{A} = 0\,, \tag{2.1}$$

where

$$\vec{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \quad \vec{F} \cdot \hat{n} = \begin{bmatrix} \rho(\vec{V} \cdot \hat{n}) \\ \rho u(\vec{V} \cdot \hat{n}) + P\hat{n}_x \\ \rho v(\vec{V} \cdot \hat{n}) + P\hat{n}_y \\ \rho w(\vec{V} \cdot \hat{n}) + P\hat{n}_z \\ (E\, +\, P)\vec{V} \cdot \hat{n} \end{bmatrix} \tag{2.2}$$

In the above vectors (from Equations 2.2), ρ is the density; u, v, and w are the Cartesian components of velocity; E is the total energy; P is pressure; $\hat{n}$ is the normal vector; and V is the vector of velocity components, $V = u\,\hat{i} + v\,\hat{j} + w\,\hat{k}$.

The above variables are given in terms of nondimensional values. The nondimensionalization of the variables is accomplished by choosing suitable reference states with which they are normalized. This process has the effect of scaling the variables close to unity which reduces the stiffness in the resulting linearized system and helps to avoid numerical errors. The reference states used in this work are

$$\rho_{ref} = \rho_\infty \, , \; U_{ref} = U_\infty \, , \; M_{ref} = M_\infty \, , \; P_{ref} = \gamma P_\infty$$

where a subscript ∞ denotes freestream values and U represents the magnitude of the velocity vector. The nondimensional values, denoted by a subscript n, are thereby given as

$$\rho_n = \frac{\rho}{\rho_{ref}} \, , \; u_n = \frac{u}{U_{ref}} \, , \; v_n = \frac{v}{U_{ref}} \, , \; P_n = \frac{P}{P_{ref}}$$

The primary focus of this research is the two-dimensional form of the equations, which has the effect of removing the fourth entry in the $\vec{Q}$ and $\vec{F} \cdot \hat{n}$ vectors. The system of equations is closed using the perfect gas equation of state:

$$P \;=\; (\gamma - 1)\left[E - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right] \tag{2.3}$$

Spatial Discretization

The equations defined in 2.1 are valid for any arbitrary volume and thus can be written at all control volumes in a given mesh. For example, the equations may be written for control volume i as

$$\frac{\partial}{\partial t}\int_{CV_i} \vec{Q}_i \, dV_i + \oint_{CS_i} \vec{F} \cdot \hat{n} \, d\vec{A}_i = 0 \, . \tag{2.4}$$

Before the discretization can continue further, the definition of a control volume must be given. There are several possible choices in defining control volumes for finite volume solvers. One approach is to directly use the elements represented by the primitive mesh, the mesh that is output from the grid generation process, and is called a cell-centered scheme. In unstructured topologies, this typically corresponds to triangles and quadrilaterals in two dimensions and tetrahedra, pyramids, prisms, and hexahedra in three dimensions. Another approach, and the one utilized in this research, is to define the control volumes as surrounding each of the nodes in the primitive mesh, making it a vertex-centered scheme. To specify the boundaries of the control volumes, the median dual approach is applied. For two dimensions, the median dual constructs the CV boundaries by connecting the midpoint of every element edge in the primitive mesh to the corresponding element centroid by a linear edge. This is illustrated in Figure 2.1 for a simple triangular mesh. The median dual has an extension to three dimensions, as well. Here, the CV boundaries are constructed by connecting the midpoint of an element's edge to the corresponding centroid of the adjacent element face and then to the element's centroid. In either dimensional case, the median dual constructs a collection of nonoverlapping control volumes that span the entire extent of the mesh provided that the primitive mesh also satisfies this criteria.

The flux integral appearing in equation 2.4 over the surface of the control volume is then approximated by a numerical quadrature. The equation may be written with the numerical approximation as

$$\frac{\partial}{\partial t}\int_{CV_i} \vec{Q}_i \, dV_i + \Re_i = 0 \,, \tag{2.5}$$

where $\Re$ is the flux residual and contains the contributions from the numerical approximation of the flux integral. The flux vector appearing in the residual is also replaced by a suitable numerical flux vector, $\vec{\Psi}$, since the values of the conserved variables are in general

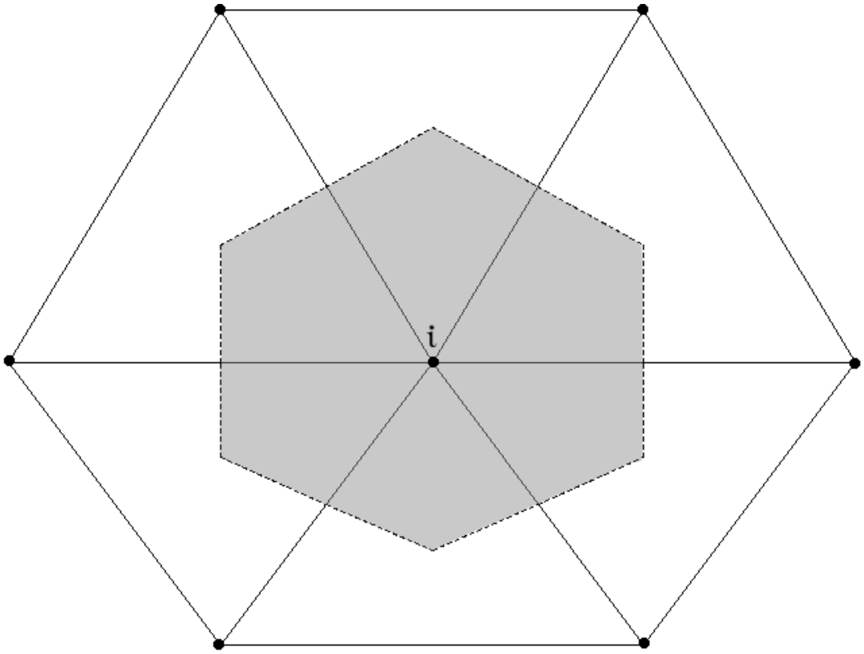


Figure 2.1 Example of a median dual construction around vertex i

discontinuous across the boundary surfaces between neighboring CVs. The numerical flux is calculated as a sum of the fluxes through the boundary surfaces of control volume i and its neighbors that share a common face (or edge in two dimensions). This can be written mathematically as

$$\Re_i = \sum_{j \in S_i} I(\vec{\Psi}_{i,j} \cdot \vec{n}_{i,j}), \tag{2.6}$$

where I represents a generic quadrature scheme and S_i is the stencil of surrounding neighbors for CV i.

The remainder of the discussion on the discretization of the equations is restricted to two dimensions on grids that are static. This restricted focus is of particular expediency since all results presented in this thesis are restricted to fixed grids in two dimensions.

Roe Scheme

The numerical flux vector implemented in this research is the Roe scheme developed in [18]. This procedure solves a one-dimensional approximate Riemann problem using the two solution vectors from both sides of the control volume interface, $\vec{Q}_L$ and $\vec{Q}_R$. The Roe scheme expresses the numerical flux as

$$\vec{\Psi} \cdot \hat{n} = \frac{1}{2}\Big(\vec{F}(\vec{Q}_L) + \vec{F}(\vec{Q}_R) - |\tilde{A}(\vec{Q}_L, \vec{Q}_R)| \cdot (\vec{Q}_R - \vec{Q}_L)\Big) \tag{2.7}$$

where the matrix $|A| = R|\Lambda|R^{-1}$. The matrices R and R^{-1} are comprised of the right and left eigenvectors, respectively, and, following Hirsch in [19], may be given as

$$R = \begin{bmatrix} 1 & 0 & \frac{\rho}{2c} & \frac{\rho}{2c} \\ u & \rho\hat{n}_y & \frac{\rho u}{2c} + \frac{\rho\hat{n}_x}{2} & \frac{\rho u}{2c} - \frac{\rho\hat{n}_x}{2} \\ v & -\rho\hat{n}_x & \frac{\rho v}{2c} + \frac{\rho\hat{n}_y}{2} & \frac{\rho v}{2c} - \frac{\rho\hat{n}_y}{2} \\ \frac{1}{2}V^2 & \rho(u\hat{n}_y - v\hat{n}_x) & \frac{1}{2}\rho(\frac{1}{2c}V^2 + \theta + \frac{c}{(\gamma-1)}) & \frac{1}{2}\rho(\frac{1}{2c}V^2 - \theta + \frac{c}{(\gamma-1)}) \end{bmatrix} \tag{2.8}$$

$$R^{-1} = \begin{bmatrix} 1 - \frac{1}{2c^2}(\gamma-1)V^2 & \frac{(\gamma-1)u}{c^2} & \frac{(\gamma-1)v}{c^2} & \frac{-(\gamma-1)}{c^2} \\ \frac{-u\hat{n}_y + v\hat{n}_x}{\rho} & \frac{\hat{n}_y}{\rho} & \frac{-\hat{n}_x}{\rho} & 0 \\ \frac{-\theta}{\rho} + \frac{1}{2\rho c}(\gamma-1)V^2 & \frac{\hat{n}_x}{\rho} - \frac{(\gamma-1)u}{\rho c} & \frac{\hat{n}_y}{\rho} - \frac{(\gamma-1)v}{\rho c} & \frac{(\gamma-1)}{\rho c} \\ \frac{\theta}{\rho} + \frac{1}{2\rho c}(\gamma-1)V^2 & \frac{-\hat{n}_x}{\rho} - \frac{(\gamma-1)u}{\rho c} & \frac{-\hat{n}_y}{\rho} - \frac{(\gamma-1)v}{\rho c} & \frac{(\gamma-1)}{\rho c} \end{bmatrix} \tag{2.9}$$

The eigenvalues are given (in absolute value) as

$$|\Lambda| = \begin{bmatrix} |\theta| & 0 & 0 & 0 \\ 0 & |\theta| & 0 & 0 \\ 0 & 0 & |\theta + c| & 0 \\ 0 & 0 & 0 & |\theta - c| \end{bmatrix} \tag{2.10}$$

In the above matrices, $\theta = u\hat{n}_x + v\hat{n}_y$ and $V^2 = \vec{V} \cdot \vec{V} = u^2 + v^2$. Using this similarity transformation for $|A|$, the numerical flux is implemented as

$$\vec{\Psi} \cdot \hat{n} = \frac{1}{2}\left(\vec{F}(\vec{Q}_L) + \vec{F}(\vec{Q}_R) - \tilde{R}|\tilde{\Lambda}|\widetilde{R^{-1}} \cdot (\vec{Q}_R - \vec{Q}_L)\right) \tag{2.11}$$

The tilde in the above equations denote that the quantities are to be evaluated with Roe averages, which are specified as

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}$$

$$\tilde{u} = \frac{\sqrt{\rho_L}u_L + \sqrt{\rho_R}u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$

$$\tilde{v} = \frac{\sqrt{\rho_L}v_L + \sqrt{\rho_R}v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$

$$\tilde{H}_t = \frac{\sqrt{\rho_L}H_{tL} + \sqrt{\rho_R}H_{tR}}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$

$$\tilde{c}^2 = (\gamma - 1)\left[\tilde{H}_t - \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2)\right]$$

Here, H_t is the total enthalpy and c is the speed of sound.

Boundary Conditions

Control volumes with boundary surfaces lying coincident with physical boundaries in the mesh must have proper boundary conditions to account for the residual contributions along these surfaces (or edges in two dimensions). The boundary conditions implemented in this research are referred to as Characteristic Variable Boundary Conditions (CVBCs). As suggested by the name, CVBCs are developed by the use of applying characteristic theory to the equations. The one-dimensional Euler equations in differential form can be used to illustrate the procedure. The equations written as

$$\frac{\partial Q}{\partial t} + \frac{\partial f}{\partial Q}\frac{\partial Q}{\partial x} = 0\,,$$

can be transformed using the similarity relationship from above as

$$\frac{\partial Q}{\partial t} + R\Lambda R^{-1}\frac{\partial Q}{\partial x} = 0$$

If the equation is premultiplied by R^{-1}, the result is

$$R^{-1}\frac{\partial Q}{\partial t} + \Lambda R^{-1}\frac{\partial Q}{\partial x} = 0$$

If R and R^{-1} are treated as constants, using a subscript 0 to denote this, the matrix can be moved into the differential operator yielding

$$\frac{\partial(R_0^{-1}Q)}{\partial t} + \Lambda\frac{\partial(R_0^{-1}Q)}{\partial x} = 0$$

Using a change of variable, this can be written as

$$\frac{\partial W}{\partial t} + \Lambda\frac{\partial W}{\partial x} = 0$$

W is now said to be the vector of characteristic variables. Since Λ is a diagonal matrix, the system of equations represented by the one-dimensional Euler equations becomes decoupled. Characteristic theory requires information to propagate along curves whose slope is given by the eigenvalues (the entries in Λ). These characteristic curves thus specify from which direction information is said to come from. Consider the case where the eigenvalues are constant, reducing the characteristics to lines, as an example. This can be seen in Figure 2.2. The point where the three characteristics intersect is a point along the boundary at a specific instance in time. The figure illustrates that the first eigenvalue contains information from inside the computational domain and that eigenvalues 2 and 3 bring in information from outside the computational domain, which is typically specified in the problem formulation.

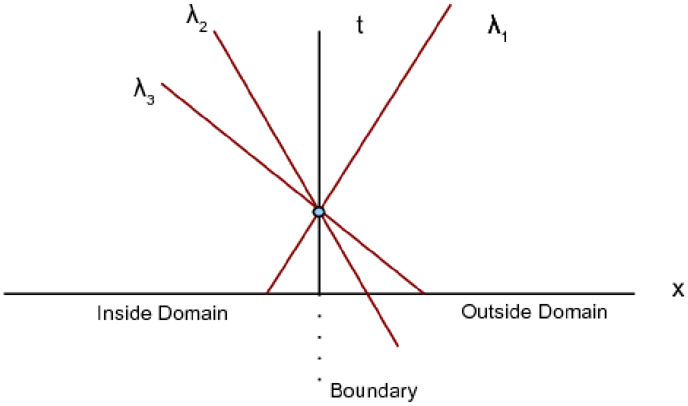


Figure 2.2 Example of characteristics with constant eigenvalues

To make the CVBCs compatible with the numerical Roe flux, the boundary values are stored at ghost CVs that are topologically connected to a corresponding physical CV. The flux at the boundary surface can then be properly computed to yield the correct result. In this way, the values of the conserved variables are not strongly enforced by setting the correct value but rather by ensuring that the correct flux will be calculated along the boundary surfaces. The implementation followed the work of Whitfield and Janus in [20]. In this work, the necessary boundary values were given for various flow conditions. Also note that in this research, the normal vector from the boundary surface is directed away from the computational domain and that the eigenvalues are denoted as $\lambda_{1,2} = \theta$, $\lambda_3 = \theta + c$, and $\lambda_4 = \theta - c$.

Subsonic Outflow

Subsonic outflow is characterized by $\lambda_{1,2,3} > 0$ and $\lambda_4 < 0$. This means that three pieces of information come from inside the computational domain and one piece is from outside the domain. The outside condition used is static pressure. The boundary values can be expressed as

$$\begin{aligned}
p_b &= p_\infty \\
\rho_b &= \rho_i + \frac{p_b - p_i}{c_0^2} \\
u_b &= u_i - \hat{n}_x \frac{p_b - p_i}{c_0 \rho_0} \\
v_b &= v_i - \hat{n}_y \frac{p_b - p_i}{c_0 \rho_0}
\end{aligned} \tag{2.12}$$

In the above equations, the subscript b denotes the boundary values, i denotes the values at the physical CV, and ∞ corresponds to freestream values. The 0 subscript represents the reference state about which the equations have been linearized. The reference chosen was to use the average of the values of the physical and ghost CV from the previous time step.

Subsonic Inflow

Subsonic inflow is characterized by $\lambda_{1,2,4} < 0$ and $\lambda_3 > 0$. In this case, three pieces of information come from outside the domain and are generally taken to be a specified density and velocity vector. The fourth piece of information thus comes from inside the computational domain and corresponds to the positive eigenvalue. The boundary values are then set to

$$\begin{aligned}
p_b &= \tfrac{1}{2}\left[p_\infty + p_i - \rho_0 c_0 \left(\hat{n}_x(u_\infty - u_i) + \hat{n}_y(v_\infty - v_i)\right)\right] \\
\rho_b &= \rho_\infty + \frac{p_b - p_\infty}{c_0^2} \\
u_b &= u_\infty + \hat{n}_x \frac{p_b - p_\infty}{c_0 \rho_0} \\
v_b &= v_\infty + \hat{n}_y \frac{p_b - p_\infty}{c_0 \rho_0}
\end{aligned} \tag{2.13}$$

Supersonic Inflow and Outflow

Supersonic flow conditions are characterized by the eigenvalues all having the same sign. In the case of inflow, the values at the boundary are taken to be the specified values. For outflow, the conditions at the boundary are taken from the interior of the domain which is taken to mean the values at the physical CV.

Inviscid Surface

The inviscid surface boundary conditions represents an impermeable boundary that obstructs flow, such as the subject of the simulation. The physical boundary condition is for the flow to be parallel to the boundary, mathematically stated as

$$\vec{u} \cdot \hat{n} = 0$$

The eigenvalues will be $\lambda_{1,2} = 0$, $\lambda_3 > 0$, and $\lambda_4 < 0$. Using this information, the boundary values are then found to be

$$\begin{aligned} p_b &= p_i + \rho_0 c_0 \left(u_i \hat{n}_x + v_i \hat{n}_y \right) \\ \rho_b &= \rho_i + \frac{p_b - p_i}{c_0^2} \\ u_b &= u_i - \hat{n}_x \frac{p_b - p_i}{c_0 \rho_0} \\ v_b &= v_i - \hat{n}_y \frac{p_b - p_i}{c_0 \rho_0} \end{aligned} \tag{2.14}$$

An alternative formulation for the inviscid surface condition that does not use characteristic values is to modify the flux directly using the fact that $\vec{u} \cdot \hat{n} = 0$. This reduces the Euler flux to

$$\vec{F} = \begin{bmatrix} 0 \\ P\hat{n}_x \\ P\hat{n}_y \\ 0 \end{bmatrix}$$

High-Order Accuracy

High-order spatial accuracy is achieved by extrapolating (reconstructing) the vector of conserved variables, $\vec{Q}$, from the nodes to the appropriate interface between neighboring control volumes, where the flux is computed. For second-order accuracy, this can be accomplished using the solution gradients in the CVs to obtain the solution state at the CV interfaces as

$$\vec{Q}_{interface} = \vec{Q}_i + \nabla\vec{Q}_i \cdot \vec{r} \tag{2.15}$$

where $\vec{r} = \vec{x}_{interface} - \vec{x}_i$. In this context, $\vec{x}_{interface}$ is the midpoint of the interface between two adjoining CVs and is the quadrature point used to approximate the integral. It is important to note that for second-order spatial accuracy a single quadrature point is sufficient. The calculation for the gradients needed in the extrapolation are found by solving an unweighted, overconstrained system in a least squares sense. The construction of the system is based on a truncated Taylor series expansion of the local terms. The detailed development of this formulation can be found in the work of Anderson and Bonhaus in [21] and later by Hyams in [22]. Though the goal of this research is to extend spatial accuracy beyond second-order, its inclusion is important since it is easier to test and gives a reference point for comparing performance.

Third-order spatial accuracy is accomplished in similar manner. The extrapolation though must now also include second derivatives and takes the formulation

$$\vec{Q}_{interface} = \vec{Q}_i + \nabla\vec{Q}_i \cdot \vec{r} + \frac{1}{2}\vec{r}^T \cdot \nabla^2\vec{Q}_i \cdot \vec{r}, \tag{2.16}$$

where $\nabla^2\vec{Q}_i$ represents the Hessian matrix (the matrix of second derivatives). For fourth-order, the extrapolation must again include more derivatives,

$$\vec{Q}_{interface} = \vec{Q}_i + \nabla\vec{Q}_i \cdot \vec{r} + \frac{1}{2}\vec{r}^T \cdot \nabla^2\vec{Q}_i \cdot \vec{r} + \frac{1}{6}(\vec{r} \cdot (\vec{r} \cdot (\vec{r} \cdot \nabla^3\vec{Q}_i))) \tag{2.17}$$

Here, $\nabla^3\vec{Q}_i$ is a tensor containing the third derivatives. It is important to note here that second-order extrapolations typically use the cell average value for $\vec{Q}_i$. For higher order schemes, this is no longer correct; the point value at the cell vertex must be used during the extrapolation in order to prevent an $O(h^2)$ error from being introduced [7].

Whereas a single quadrature point was necessary for second-order, third-order accuracy (and higher orders) requires that the numerical approximation of the integral be of a commensurate degree of accuracy. The number of quadrature points necessary to satisfy this requirement will vary depending on the numerical scheme chosen. Gaussian quadrature was chosen in this work since it gives maximal order with the fewest quadrature points. A more complete treatment of the algorithm for high-order accuracy will be given in Chapter 3.

Temporal Discretization and Evolution

The time derivative appearing in Equation 2.5 still needs to be discretized so that the equations can be solved numerically. First, the volume integral can be replaced using the substitution

$$\frac{1}{V_i}\int_{CV_i} \vec{Q}_i \, dV_i = \overline{Q}_i \tag{2.18}$$

so that $\overline{Q}_i$ is said to be the control volume average. If the residual is calculated using the values at the next time level, the equation can now be rewritten in implicit form as

$$\frac{\partial}{\partial t}(V_i\overline{Q}_i) + \Re(\overline{Q}^{n+1}) = 0 \tag{2.19}$$

Replacing the time derivative with a second-order finite difference approximation yields

$$V_i\left(\frac{3\overline{Q}^{n+1} - 4\overline{Q}^{n} + \overline{Q}^{n-1}}{2\triangle t}\right) + \Re(\overline{Q}^{n+1}) = 0 \tag{2.20}$$

The residual is now linearized about the current solution by applying Newton's Method. The goal here is to drive the Newton iterates $\overline{Q}^{m+1}$ to $\overline{Q}^{n+1}$. The equation can now be written as

$$\left[\frac{3}{2}\frac{V_i}{\triangle t}I + \frac{\partial \Re^m}{\partial Q}\right]\triangle\overline{Q} = -\left[\frac{V_i}{2\triangle t}(3\overline{Q}^{m} - 4\overline{Q}^{n} + \overline{Q}^{n-1}) + \Re(\overline{Q}^{m})\right] \tag{2.21}$$

This equation yields a sparse linear system that is solved for the solution update $\triangle\overline{Q} = \overline{Q}^{m+1} - \overline{Q}^{m}$. The linear system may be solved by a number of means, but the choice for this research is to use the Symmetric Gauss-Seidel (SGS) method. The solution of the linear system produces an update vector for the next Newton iteration. The time step used in Equation 2.21 is determined by considerations of the physical flow.

The above equations refer to unsteady, or time accurate, flow. In steady-state flow, the time derivative appearing in Equation 2.5 is zero. In practice though, the time derivative is included; it functions to increase diagonal dominance of the linear system allowing the solution update to be computed by the SGS algorithm. Using a first-order finite difference expression for time, this yields

$$\left[\frac{V_i}{\triangle t} I + \frac{\partial \Re^m}{\partial Q}\right] \triangle \overline{Q} = -\Re(\overline{Q}^m) \tag{2.22}$$

As opposed to unsteady flow where the time step is limited to physical conditions, steady-state flow allows the time step to be as large as is necessary to provide stability. Further, since time evolution in this case is fictitious, each CV is allowed to have a distinct time step. This procedure is referred to as local time stepping. In local time stepping, each CV determines the maximum time step it can process based on a global CFL number and the maximum local eigenvalue, which determine the speed at which information is propagating through the spatial extent of the CV. This can expressed mathematically as

$$\triangle t_i = CFL \frac{V_i}{\sum_{j \in Si} \left(|\theta| + c\right) A} \tag{2.23}$$

where the sum is over the CV interfaces from the median dual construction and A is the length of the interface.

Though there is no physical basis for a limit on the time step, in practice, the global CFL is generally started at a low number, such as 1, and then allowed to ramp up to a maximum value over a specified number of iterations. Starting with a low CFL number at the beginning of the iterations has the effect of adding additional stability to the linear system. This is particularly useful since the largest changes in the solution are expected at the beginning of the iteration process.

Flux Jacobians

The formulation of the Euler equations into an implicit solution procedure generates the presence of the flux Jacobian $\frac{\partial \Re}{\partial Q}$. It represents the change in the residual in all control volumes with respect to the change in the solution variables. In other words, the calculation of $\frac{\partial \Re}{\partial Q}$ depends on computing $\frac{\partial \Phi}{\partial Q}$. For the Roe scheme, the flux Jacobian is generally computed in one of two ways. The first approach is to apply a perturbation to each component of the solution vector, one at a time, and use the flux calculations of the perturbed and unperturbed solution vectors to form a forward finite difference approximation. The computation of the numerical flux involves left and right states corresponding to two CVs that share a face and so the flux Jacobian must likewise be differentiated with respect to Q_L and Q_R. The finite difference approximation can be written as

$$\frac{\partial F_E}{\partial Q_{Li}} = \frac{[F(Q_L + \epsilon e_i, Q_R) - F(Q_L, Q_R)]_E}{\epsilon} \tag{2.24}$$

$$\frac{\partial F_E}{\partial Q_{Ri}} = \frac{[F(Q_L, Q_R + \epsilon e_i) - F(Q_L, Q_R)]_E}{\epsilon} \tag{2.25}$$

where e_i is the i^{th} unit vector and the subscript E denotes the flux is defined for a particular CV interface which is an edge in two dimensions. The value of epsilon used is in general on the order of the square root of machine zero.

The other approach, and the one undertaken in this research, is to use an approximate flux linearization. Recalling that the Roe flux vector can be written as

$$F_{Roe} = \frac{1}{2}\left(\vec{F}(\vec{Q}_L) + \vec{F}(\vec{Q}_R) - |\tilde{A}(\vec{Q}_L, \vec{Q}_R)| \cdot (\vec{Q}_R - \vec{Q}_L)\right) ,$$

the flux can easily be differentiated if the matrix $\tilde{A}$ is treated as constant. Differentiating the above equation with respect to both Q_L and Q_R yields

$$\frac{\partial F_{Roe}}{\partial Q_L} = \frac{1}{2}\left(A(Q_L) + |\tilde{A}|\right) \tag{2.26}$$

$$\frac{\partial F_{Roe}}{\partial Q_R} = \frac{1}{2}\left(A(Q_R) - |\tilde{A}|\right) \tag{2.27}$$

where A is the analytical derivative of the Euler flux vector with respect to the conserved variables.

Overview of Solution Procedure

The previous sections described the various details that went into the construction of a two-dimensional, unstructured flow solver for the compressible, inviscid Euler equations. It seems prudent to summarize this information and discuss any remaining issues before proceeding into the details of the solution reconstruction.

One detail that was neglected in the discussion of the linear solver is that a fixed number of iterations are applied rather than fully converging the solution update, or at least within some specified tolerance. With point-iterative schemes like SGS, the high frequency errors can be effectively dampened by a small number of iterations. Fully converging the linear system would take many more iterations. Now since the solution update from the linear system is not the converged solution, this impacts the convergence of the Newton iterations. There is thus a trade-off between the number of iterations spent in the linear solve and the number of iterations needed to converge the solution in Newton's method. Experience is generally the guiding hand in finding the balance between the two.

Another detail that was not fully explained is the use of approximations to the flux Jacobian and its effect on the solution of the system. Using the procedures above, the residual is computed at full accuracy while the Jacobians that form the left hand side of the linear problem are approximated at a lower order since: 1) the full stencil of surrounding control volumes is not used and 2) an approximate linearization of the flux is used. The effect of this mismatch of accuracy between the left and right hand sides of the equation is again manifested in convergence of the Newton iterations. In other words, the recovery of the correct solution is expected to incur an increased count of iterations. This penalty is not so egregious considering that computing the full order left hand side would require a commensurate amount of additional calculations.

A flowchart is offered in Figure 2.3 that displays the steps utilized in the flow solver in a hierarchy. The solution process begins by assigning initial values to the solution variables. The outermost loop iterates over the number of time steps specified. In the next step, the value of the time step is calculated. For unsteady solutions, the value is specified at run time; whereas in steady-state solutions, local time stepping is applied. The solver then begins the Newton iterations where most of the work is done. For steady state solutions, a single Newton iteration is used. In unsteady solutions, the number of iterations is chosen so that the unsteady residual (the right hand side of Equation 2.21) converges to a desired level. For steady state solutions, the solver does not use high-order reconstruction at the beginning. Rather, the procedure is to begin with first-order iterations and then switch to second-order reconstruction after a specified number of iterations. This stabilizes the solution process by dampening out large oscillations in the derivatives. To achieve third-order solutions, the second-order reconstruction is used for another specified number of iterations. The process is analogous for fourth-order solutions.

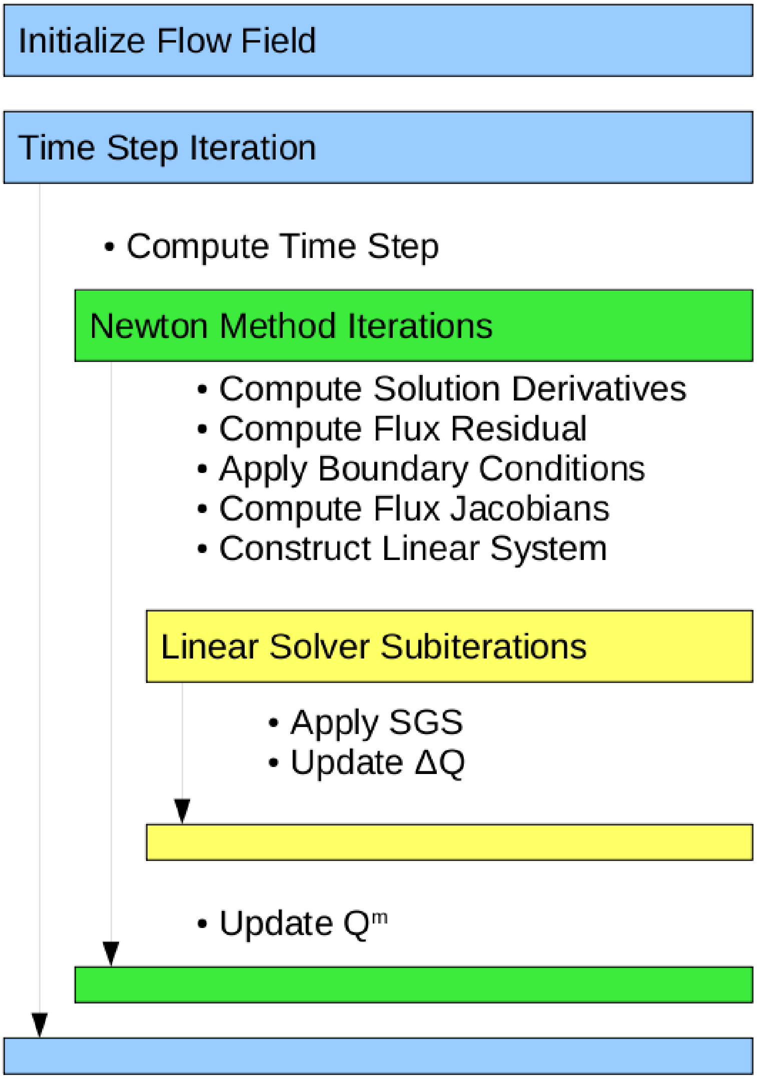


Figure 2.3 Flowchart of the solution procedure in the two-dimensional flow solver

CHAPTER 3
RECONSTRUCTION

The computation of high-order solutions necessitates a high-order representation of the conserved variables inside the control volume. As seen in Chapter 1, there are a number of ways to accomplish this for a flow solver for the Euler equations. Flow solvers for structured grid topologies have made use of ENO and WENO schemes to serve this purpose. For these flow solvers, the implementation of the methods is natural due to the mesh regularity present in structured grids. The advent of unstructured grid solvers as the dominant paradigm in numerical simulation has required the adoption of techniques from structured grid solvers to be adapted for unstructured meshes. The work of Ollivier-Gooch *et al* [13, 14, 15] seen in Chapter 1 provides a framework for extending the method of Barth and Frederickson [6] which uses a minimum energy solution (from the least squares problem) to generate a k-exact reconstruction of the conserved variables. The researchers present their method as an analogous form of WENO methods for unstructured grids. Along with the work described in Chapter 1 for finite volume techniques, other researchers have contributed similar methods for unstructured grids including Suresh and Jorgenson [23] who developed ENO type methods and Haselbacher [24] who developed a WENO type algorithm. The crux of these schemes, including this research, is how the neighbor stencil is used to provide a high-order representation of the variables. This chapter will construct the framework of the method used and outline the details of implementation.

Overview of Reconstruction Scheme

The objective of k-exact reconstruction is to find a representation of the solution based on data from adjacent control volumes where the induced truncation error of the solution inside each control volume is on the order of h^k, where h is some measure of local grid spacing. Barth and Frederickson presented three criteria for the reconstruction operator: conservation in the mean, k-exact reconstruction, and compact support. The meaning of each of these criteria in the context of the reconstruction scheme is described below.

Conservation in the Mean

In the context of a finite volume flow solver, the domain of interest is tessellated into a collection of disjoint control volumes. Each control volume V_i has a geometric reference point $\vec{x_i}$. As discussed in the previous chapter, the median dual is employed to build the control volumes from the original mesh. Since this is the case, the reference points are chosen to be the element vertices. This choice makes the reference point consistent with existing flow solver techniques, including the Tenasi flow solver. Since the new method is indifferent to the choice of the reference point, this decision will not impact performance. Suppose now there is a smooth function $u(\vec{x})$ that has support over the control volume V_i and a reconstruction operator $R_i(\vec{x}-\vec{x_i})$ that is also valid in the control volume. Conservation in the mean requires that control volume averaged values of the functions u and R_i be equivalent (at least up to truncation error):

$$\frac{1}{V_i}\int_{V_i} R_i(\vec{x}-\vec{x_i})dV = \frac{1}{V_i}\int_{V_i} u(\vec{x})dV \equiv \overline{u_i} \tag{3.1}$$

$$R_i(\vec{x}-\vec{x_i}) = \sum_{a+b+c\leq k} \alpha_{(a,b,c)} P_{(a,b,c)}(x-x_i, y-y_i, z-z_i) \tag{3.2}$$

$$P_{(a,b,c)}(x - x_i, y - y_i, z - z_i) = (x - x_i)^a (y - y_i)^b (z - z_i)^c \tag{3.3}$$

The $\alpha_{(a,b,c)}$ terms appearing in Equation 3.2 are taken to be the derivatives of a continuous function with respect to the geometric variables which can be seen when the Taylor series expansion of the function R_i is expressed as:

$$\begin{aligned} R_i(\vec{x} - \vec{x_i}) = \; & u\big|_{\vec{x_i}} + \frac{\partial u}{\partial x}\bigg|_{\vec{x_i}} (x - x_i) + \frac{\partial u}{\partial y}\bigg|_{\vec{x_i}} (y - y_i) + \frac{\partial u}{\partial z}\bigg|_{\vec{x_i}} (z - z_i) \\ & + \frac{\partial^2 u}{\partial x^2}\bigg|_{\vec{x_i}} \frac{(x - x_i)^2}{2} + \frac{\partial^2 u}{\partial y^2}\bigg|_{\vec{x_i}} \frac{(y - y_i)^2}{2} \\ & + \frac{\partial^2 u}{\partial z^2}\bigg|_{\vec{x_i}} \frac{(z - z_i)^2}{2} + \frac{\partial^2 u}{\partial x \partial y}\bigg|_{\vec{x_i}} ((x - x_i)(y - y_i)) \\ & + \frac{\partial^2 u}{\partial x \partial z}\bigg|_{\vec{x_i}} ((x - x_i)(z - z_i)) + \frac{\partial^2 u}{\partial y \partial z}\bigg|_{\vec{x_i}} ((y - y_i)(z - z_i)) + ... \end{aligned} \tag{3.4}$$

The imposition of the conservation of the mean is achieved by integrating the above expansion over the control volume i and setting this result equal to the mean value using the relation from Equation 3.1 yielding:

$$\begin{aligned} \overline{u_i} = \; & u\big|_{\vec{x_i}} + \frac{\partial u}{\partial x}\bigg|_{\vec{x_i}} \overline{x_i} + \frac{\partial u}{\partial y}\bigg|_{\vec{x_i}} \overline{y_i} + \frac{\partial u}{\partial z}\bigg|_{\vec{x_i}} \overline{z_i} \\ & + \frac{\partial^2 u}{\partial x^2}\bigg|_{\vec{x_i}} \frac{\overline{x_i^2}}{2} + \frac{\partial^2 u}{\partial y^2}\bigg|_{\vec{x_i}} \frac{\overline{y_i^2}}{2} \\ & + \frac{\partial^2 u}{\partial z^2}\bigg|_{\vec{x_i}} \frac{\overline{z_i^2}}{2} + \frac{\partial^2 u}{\partial x \partial y}\bigg|_{\vec{x_i}} \overline{xy_i} \\ & + \frac{\partial^2 u}{\partial x \partial z}\bigg|_{\vec{x_i}} \overline{xz_i} + \frac{\partial^2 u}{\partial y \partial z}\bigg|_{\vec{x_i}} \overline{yz_i} + ... \end{aligned} \tag{3.5}$$

where

$$\overline{(x^a y^b z^c)_i} \equiv \frac{1}{V_i} \int_{V_i} (x - x_i)^a (y - y_i)^b (z - z_i)^c dV \tag{3.6}$$

forms the relation for the control volume moments.

k-Exact Reconstruction

The reconstruction function is said to be k-exact if R_i reconstructs polynomials of degree k or less exactly. It further suggests that if the exact solution is given by a smooth function $U_E(\vec{x})$ then the following can be said:

$$R_i(\vec{x} - \vec{x_i}) = U_E(\vec{x}) + O((h)^{k+1}) \tag{3.7}$$

For the integral form, this implies that the error between the exact mean value and the reconstructed mean value is given by:

$$\frac{1}{V_i} \int_{V_i} R_i(\vec{x} - \vec{x_i}) dV - \frac{1}{V_i} \int_{V_i} U_E(\vec{x}) dV = O((h)^{k+1}) \tag{3.8}$$

Compact Support

The criterion of compact support enforces locality into the stencil selection. The stencil of control volumes surrounding V_i will consist of neighbors that are both spatially near V_i and express topological locality to V_i. In the presence of a smooth function with support throughout the mesh, the data from the neighbors in the stencil will also be physically close to data of V_i. This reconstruction stencil must be large enough to compute the values of the derivatives associated with the control volume. In the present research, attention is restricted to third- and fourth-order accurate reconstructions. For third-order, this entails finding five derivatives in two dimensions and nine in three dimensions. For fourth-order, this requires finding nine and nineteen derivatives in two and three dimensions, respectively. For the general case of a k^{th}-order accurate reconstruction, the size of the stencil must be at least $(k+1)(k+2)(k+3)/6$. It is desired, though, to have more neighbors in the stencil than there are derivatives so as to increase the robustness of the reconstruction scheme. Since it is generally insufficient to have only first degree neighbors (those connecting to vertex i by an edge in the original mesh) comprise the stencil, second degree neighbors are also added to the stencil (first degree neighbors of vertex i's set of first degree neighbors) for third-order. For fourth-order, third degree neighbors, which are the first degree neighbors of the second degree neighbors, are added to the stencil. During the generation of the stencil for the vertices, only unique neighbors are stored. To ensure robustness, the nodes are checked to verify that enough neighbors are in their stencil. Nodes that have only the minimum number of neighbors or less are given the next layer of neighbors. An example of a representative stencil for third-order can be seen for a simple two dimensional mesh in Figure 3.1. The control volume of vertex i is seen colored in yellow. Its first degree neighbors are colored in green and second degree neighbors are in blue. Other control volumes further separated from V_i are in grey. The extension of this method of stencil selection is trivial for three dimensions; however, it more difficult to render an appropriate representative stencil.

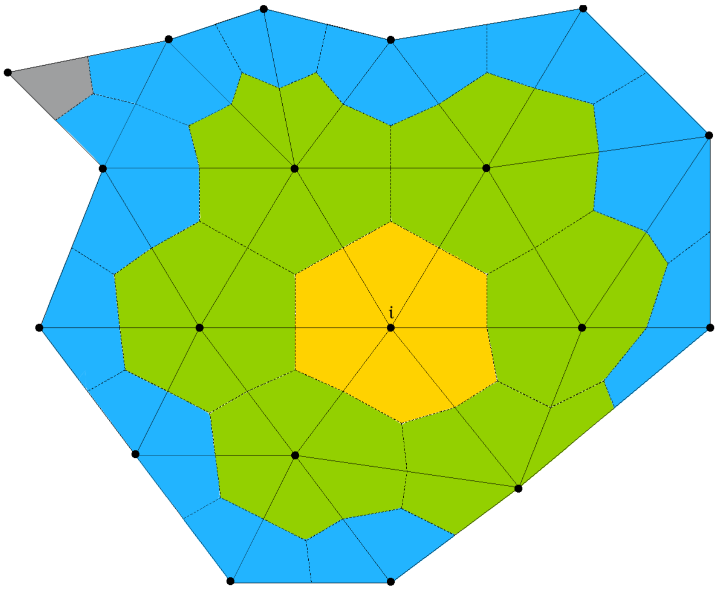


Figure 3.1 Neighborhood of control volumes around V_i color coded to represent the degree of topological separation

Implementation

The studied high-order accurate reconstruction scheme requires the computation of the desired derivatives by minimizing the error in computing the mean value of the reconstructed function in all control volumes V_j in the stencil surrounding V_i. In other words, the scheme should find the derivatives such that the error between the actual control volume averages $\overline{u_j}$ and the volume averaged value of R_i computed over V_j is minimized. Considering third-order reconstruction and using the relationship in Equation 3.4, the volume averaged value for V_j using the reconstruction function R_i is:

$$
\begin{aligned}
\frac{1}{V_j}\int_{V_j} R_i(\vec{x}-\vec{x_i})dV = \; & u\big|_{\vec{x_i}} \\
& +\frac{\partial u}{\partial x}\bigg|_{\vec{x_i}}\left(\frac{1}{V_j}\int_{V_j}(x-x_i)dV\right) \\
& +\frac{\partial u}{\partial y}\bigg|_{\vec{x_i}}\left(\frac{1}{V_j}\int_{V_j}(y-y_i)dV\right) \\
& +\frac{\partial u}{\partial z}\bigg|_{\vec{x_i}}\left(\frac{1}{V_j}\int_{V_j}(z-z_i)dV\right) \\
& +\frac{\partial^2 u}{\partial x^2}\bigg|_{\vec{x_i}}\left(\frac{1}{2V_j}\int_{V_j}(x-x_i)^2dV\right) \\
& +\frac{\partial^2 u}{\partial y^2}\bigg|_{\vec{x_i}}\left(\frac{1}{2V_j}\int_{V_j}(y-y_i)^2dV\right) \\
& +\frac{\partial^2 u}{\partial z^2}\bigg|_{\vec{x_i}}\left(\frac{1}{2V_j}\int_{V_j}(z-z_i)^2dV\right) \\
& +\frac{\partial^2 u}{\partial x\partial y}\bigg|_{\vec{x_i}}\left(\frac{1}{V_j}\int_{V_j}(x-x_i)(y-y_i)dV\right) \\
& +\frac{\partial^2 u}{\partial x\partial z}\bigg|_{\vec{x_i}}\left(\frac{1}{V_j}\int_{V_j}(x-x_i)(z-z_i)dV\right) \\
& +\frac{\partial^2 u}{\partial y\partial z}\bigg|_{\vec{x_i}}\left(\frac{1}{V_j}\int_{V_j}(y-y_i)(z-z_i)dV\right)
\end{aligned}
\tag{3.9}
$$

The moment terms that appear in the above equation can be simplified by a simple arithmetic equality: replace $\beta-\beta_i$ with $(\beta-\beta_j)+(\beta_j-\beta_i)$ where β is x,y, or z. If we consider a single moment term, this yields:

$$\begin{aligned}\frac{1}{V_j}\int_{V_j}(x-x_i)dV = & \frac{1}{V_j}\int_{V_j}\left((x-x_j)+(x_j-x_i)\right)dV \\ = & \frac{1}{V_j}\int_{V_j}(x-x_j)dV+\frac{1}{V_j}\int_{V_j}(x_j-x_i)dV \\ = & \frac{1}{V_j}\int_{V_j}(x-x_j)dV+(x_j-x_i)\frac{1}{V_j}\int_{V_j}dV \\ = & \overline{x_j}+(x_j-x_i)\end{aligned} \tag{3.10}$$

Using this relationship, it is only necessary to compute 9 moments for each control volume rather than computing 9 moments per neighboring control volume in the stencil for each vertex i. After this substitution and then integration, Equation 3.9 becomes:

$$\begin{aligned}\frac{1}{V_j}\int_{V_j}R_i(\vec{x}-\vec{x_i})dV = \quad & u\Big|_{\vec{x_i}} \\ & +\frac{\partial u}{\partial x}\Big|_{\vec{x_i}} \quad (\overline{x_j}+(x_j-x_i)) \\ & +\frac{\partial u}{\partial y}\Big|_{\vec{x_i}} \quad (\overline{y_j}+(y_j-y_i)) \\ & +\frac{\partial u}{\partial z}\Big|_{\vec{x_i}} \quad (\overline{z_j}+(z_j-z_i)) \\ & +\frac{\partial^2 u}{\partial x^2}\Big|_{\vec{x_i}} \quad \left(\frac{1}{2}(\overline{x_j^2}+2\overline{x_j}(x_j-x_i)+(x_j-x_i)^2)\right) \\ & +\frac{\partial^2 u}{\partial y^2}\Big|_{\vec{x_i}} \quad \left(\frac{1}{2}(\overline{y_j^2}+2\overline{y_j}(y_j-y_i)+(y_j-y_i)^2)\right) \\ & +\frac{\partial^2 u}{\partial z^2}\Big|_{\vec{x_i}} \quad \left(\frac{1}{2}(\overline{z_j^2}+2\overline{z_j}(z_j-z_i)+(z_j-z_i)^2)\right) \\ & +\frac{\partial^2 u}{\partial x\partial y}\Big|_{\vec{x_i}} \quad \Big(\overline{xy_j}+\overline{x_j}(y_j-y_i)+\overline{y_j}(x_j-x_i) \\ & \qquad +(x_j-x_i)(y_j-y_i)\Big) \\ & +\frac{\partial^2 u}{\partial x\partial z}\Big|_{\vec{x_i}} \quad \Big(\overline{xz_j}+\overline{x_j}(z_j-z_i)+\overline{z_j}(x_j-x_i) \\ & \qquad +(x_j-x_i)(z_j-z_i)\Big) \\ & +\frac{\partial^2 u}{\partial y\partial z}\Big|_{\vec{x_i}} \quad \Big(\overline{yz_j}+\overline{y_j}(z_j-z_i)+\overline{z_j}(y_j-y_i) \\ & \qquad +(y_j-y_i)(z_j-z_i)\Big)\end{aligned} \tag{3.11}$$

Equation 3.11 is then written for every neighbor in the stencil. The resulting equations coupled with the mean constraint form a linear system that is solved in a least squares sense. The solution of this system may be found by a variety of ways; Gram-Schmidt orthonormalization (GS), QR factorization , singular value decomposition (SVD), etc. In two dimensions, the least squares problem becomes:

$$\left[\begin{array}{cccccc} 1 & \overline{x} & \overline{y} & \overline{x^2} & \overline{y^2} & \overline{xy} \\ \hline \\ w_{i1} & w_{i1}\widehat{x_{i1}} & w_{i1}\widehat{y_{i1}} & w_{i1}\widehat{x^2_{i1}} & w_{i1}\widehat{y^2_{i1}} & w_{i1}\widehat{xy_{i1}} \\ w_{i2} & w_{i2}\widehat{x_{i2}} & w_{i2}\widehat{y_{i2}} & w_{i2}\widehat{x^2_{i2}} & w_{i2}\widehat{y^2_{i2}} & w_{i2}\widehat{xy_{i2}} \\ w_{i3} & w_{i3}\widehat{x_{i3}} & w_{i3}\widehat{y_{i3}} & w_{i3}\widehat{x^2_{i3}} & w_{i3}\widehat{y^2_{i3}} & w_{i3}\widehat{xy_{i3}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{in} & w_{in}\widehat{x_{in}} & w_{in}\widehat{y_{in}} & w_{in}\widehat{x^2_{in}} & w_{in}\widehat{y^2_{in}} & w_{in}\widehat{xy_{in}} \end{array}\right] \begin{pmatrix} u \\ \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{1}{2}\frac{\partial^2 u}{\partial x^2} \\ \frac{1}{2}\frac{\partial^2 u}{\partial y^2} \\ \frac{\partial^2 u}{\partial x \partial y} \end{pmatrix}_i = \left(\begin{array}{c} \overline{u_i} \\ \hline \\ w_{i1}\overline{u_1} \\ w_{i2}\overline{u_2} \\ w_{i3}\overline{u_3} \\ \vdots \\ w_{in}\overline{u_n} \end{array}\right) \tag{3.12}$$

In the above system, the first row corresponds to the mean constraint. The rows below it are the equations written for every CV in the stencil using abbreviations of the geometric terms from Equation 3.11 and include weights that emphasize neighbors that are geometrically close to the control volume i. The terms are expressed as:

$$\widehat{x^a y^b z^c_{ij}} = \sum_{f=0}^{c}\sum_{e=0}^{b}\sum_{d=0}^{a} \Bigg(\frac{c!}{f!(c-f)!}\frac{b!}{e!(b-e)!}\frac{a!}{d!(a-d)!} (x_j - x_i)^d (y_j - y_i)^e (z_j - z_i)^f \overline{x^{a-d}y^{b-e}z^{c-f}}_j \Bigg) \tag{3.13}$$

$$w_{ij} = \frac{1}{|\vec{x_j} - \vec{x_i}|^p} \tag{3.14}$$

The exponent p in Equation 3.14 can take any value; however, it is suggested that values of zero, one, or two are chosen.

In practice, the least squares problem is generally stated with the mean constraint analytically eliminated. This enforces conservation in the mean and constrains the least squares problem which can now be stated as:

$$\begin{bmatrix} w_{i1}(\widehat{\overline{x}_{i1}} - \overline{x}) & w_{i1}(\widehat{\overline{y}_{i1}} - \overline{y}) & w_{i1}(\widehat{x^2_{i1}} - \overline{x^2}) & w_{i1}(\widehat{y^2_{i1}} - \overline{y^2}) & w_{i1}(\widehat{\overline{xy}_{i1}} - \overline{xy}) \\ w_{i2}(\widehat{\overline{x}_{i2}} - \overline{x}) & w_{i2}(\widehat{\overline{y}_{i2}} - \overline{y}) & w_{i2}(\widehat{x^2_{i2}} - \overline{x^2}) & w_{i2}(\widehat{y^2_{i2}} - \overline{y^2}) & w_{i2}(\widehat{\overline{xy}_{i2}} - \overline{xy}) \\ w_{i3}(\widehat{\overline{x}_{i3}} - \overline{x}) & w_{i3}(\widehat{\overline{y}_{i3}} - \overline{y}) & w_{i3}(\widehat{x^2_{i3}} - \overline{x^2}) & w_{i3}(\widehat{y^2_{i3}} - \overline{y^2}) & w_{i3}(\widehat{\overline{xy}_{i3}} - \overline{xy}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{in}(\widehat{\overline{x}_{in}} - \overline{x}) & w_{in}(\widehat{\overline{y}_{in}} - \overline{y}) & w_{in}(\widehat{x^2_{in}} - \overline{x^2}) & w_{in}(\widehat{y^2_{in}} - \overline{y^2}) & w_{in}(\widehat{\overline{xy}_{in}} - \overline{xy}) \end{bmatrix} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{1}{2}\frac{\partial^2 u}{\partial x^2} \\ \frac{1}{2}\frac{\partial^2 u}{\partial y^2} \\ \frac{\partial^2 u}{\partial x \partial y} \end{pmatrix}_i = \begin{pmatrix} w_{i1}(\overline{u_1} - \overline{u_i}) \\ w_{i2}(\overline{u_2} - \overline{u_i}) \\ w_{i3}(\overline{u_3} - \overline{u_i}) \\ \vdots \\ w_{in}(\overline{u_n} - \overline{u_i}) \end{pmatrix} \tag{3.15}$$

The mean constraint from Equation 3.5 is used in conjunction with the solution of the least squares problem and the control volume average to compute the point value of the conserved variable for vertex i. The extension of this analysis to fourth-order is straightforward; more derivatives are included, adding additional columns to the least squares problem.

The derivatives for all variables can found using the above where $\overline{u}$ is replaced by the actual variable. In the previous literature, it has been suggested to use the primitive variables (ρ, u, v, and P) during the reconstruction process. In the current work, the solver is coded such that it can reconstruct either the primitive or the conserved variables. Test results showed little if any dependence on which set of variables was chosen. Therefore, it was decided to use the conserved variables in the reconstruction, and all results are based on this selection unless otherwise stated.

High-Order Flux Integration

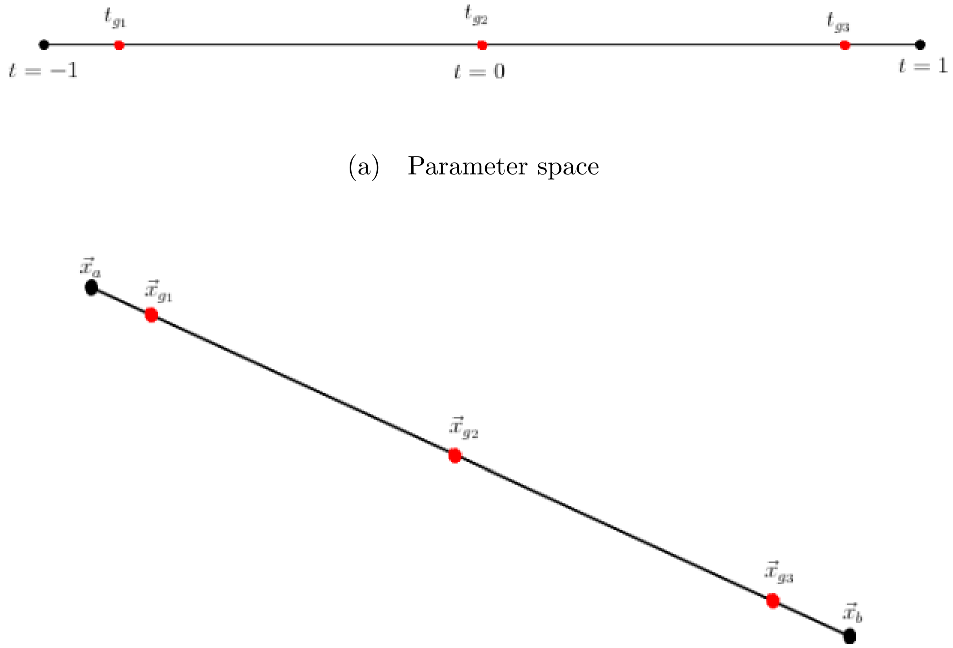
The success of a high-order flow solver also requires that the flux integral arising in the linearized system be computed to at least the intended order used in the reconstruction process. As stated in Chapter 2, this is accomplished in this research with the use of Gaussian quadrature. Three quadrature nodes are used in all calculations. This means that the solver was able to handle both third- and fourth-order reconstructions. This also adds additional resolution of curved boundaries and prevents error in the flux integral from propagating into the solution. The flux integral is computed along the edges generated by the application of the median dual to the mesh. Each edge is then parameterized by the variable t that varies linearly along the edge. The parameterization is given as

$$\vec{x}(t) = \frac{1}{2}\left(\vec{x}_a + \vec{x}_b\right) + \frac{1}{2}\left(\vec{x}_b - \vec{x}_a\right) t \tag{3.16}$$

This equation relates the locations of the quadrature nodes, which are given in terms of the parameter t, to their coordinates in x, y space. The coordinates in t-space and associated weights for the three quadrature nodes are given in Table 3.1. The action of the parameterization from Equation 3.16 can be seen for a general edge in Figure 3.2. The flux is then computed at the quadrature nodes using the extrapolated values and the result is aggregated to the appropriate control volumes. This approach is used for all edges bounding the control volumes that are linear; non-linear, or curved, boundaries are given special consideration. In the case of curved boundaries, the quadrature nodes on edges incident with a physical boundary are placed along the curve describing the boundary and not along a linear segment traversing the extent of the edge. It has been further suggested that the inclusion of curved boundaries and the proper placement of the quadrature points is a necessity for third- and higher order schemes [13]. An illustration of the placement of quadrature nodes and normal vectors can be seen for an example control volume in Figure 3.3.

Table 3.1 Gaussian quadrature nodes in one dimension

t	Weight
0	$8/9$
$\pm\sqrt{3/5}$	$5/9$



(a) Parameter space

(b) Computational space

Figure 3.2 Transformation from parameter space to an edge in the grid

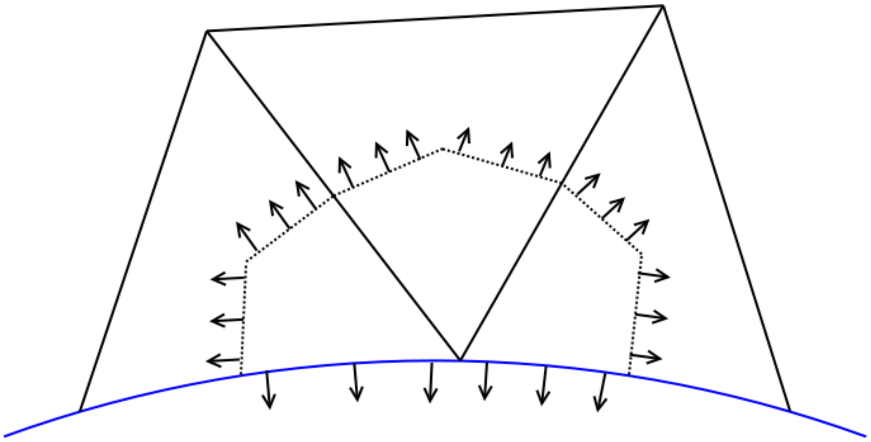


Figure 3.3 Illustration of quadrature nodes and normal vectors for a control volume

Reconstruction Test on a Smooth Function

The first step in validating the accuracy of this new procedure is to verify the accuracy in the reconstruction phase. The accuracy is proven by a grid convergence study in which three separate grids, each of which are discretized in a series of increasing resolutions, are analyzed for errors using a battery of test functions. Each of the test functions are smooth in the mathematical sense and are given as:

$$f_1(x,y) = x^2 + y^2 + xy + x + y \tag{3.17}$$

$$f_2(x,y) = 3x^3 + 5xy^2 \tag{3.18}$$

$$f_3(x,y) = \sin(\pi x)\cos(\pi y) \tag{3.19}$$

$$f_4(x,y) = e^{-r^2},\ r^2 = \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 \tag{3.20}$$

All grids used in the reconstruction test span the space $(x, y) \in [0, 1] \times [0, 1]$ and were created by the commercial software package GridGen®. The three series of grids used in this study are divided into the following categories: 1) quadrilateral elements, 2) aligned triangles, and 3) unstructured triangles. In the first category, the grids feature quadrilateral elements that are aligned with the axes. The grids in the second category were constructed from the corresponding quadrilateral grids but a diagonal is inserted into each element dividing it into two triangles. Note that this increases the number of elements in these grids but does not influence the number of control volumes as no new vertices are created. In the final category, the unstructured domain algorithm in GridGen® was used to fill in the area with triangles. The three coarsest grids of each type can be seen in Figure 3.4.

The number of control volumes used in each grid can be seen in Table 3.2. The grids were constructed by specifying the number of points along each edge of the square boundary. Each level of refinement was given twice the number of points as the previous level which reduces the nodal spacing along the boundary by approximately half as seen in Table 3.3. As

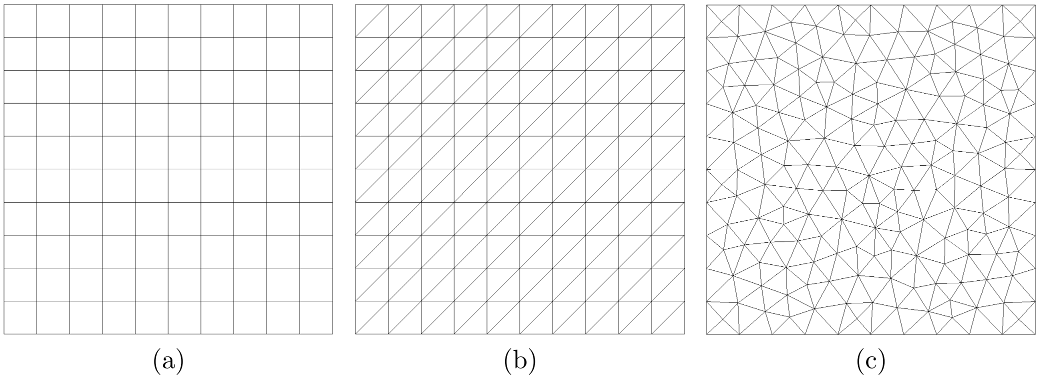


Figure 3.4 Coarsest grids used in the reconstruction test

the number of points along the boundary are doubled, the total number of control volumes increases roughly four-fold as evident in Table 3.2. In the case of the quadrilateral and aligned triangle grids, this increase translates to a reduction of the nominal interior grid spacing by a factor of two as a result of the uniform refinement utilized. The unstructured triangle grids do not enjoy this property since the refinement is applied to the boundary spacings and there is not strict guarantee that the interior spacings will be uniformly reduced due to the employed grid construction algorithm. Therefore, the unstructured grids may not posses the same convergence properties of the other two grid types.

Table 3.2 Number of control volumes in each grid

Grid	Level 0	Level 1	Level 2	Level 3
Quadrilateral	121	441	1681	6561
Aligned Triangles	121	441	1681	6561
Unstructured Triangles	192	754	2862	11175

The procedure to verify the accuracy of the reconstruction algorithm starts with the computation of control volume averages. The calculation of derivatives follows next after the construction of the least squares system. In all test cases, the third- and fourth-order

Table 3.3 Point spacing along square boundary

Refinement Level	Boundary Spacing
0	0.11111
1	0.052632
2	0.025641
3	0.012658

schemes used a weighting parameter of zero and the SVD was used to solve the least squares problem. The construction and solution of the second-order least squares system follows the standard algorithm presented in detail by Hyams [22]. The error is assessed by tracking the maximal error in each CV between the actual function value and the reconstructed function value at specified quadrature nodes. While this succinctly describes the procedure, some details need to be further elucidated.

The computation of the control volume averaged value of the function is an important step and must be computed to at least the same order as the reconstruction scheme [25]. There are two methods used in this research to accomplish this computation numerically. The first method is to apply the Divergence Theorem to transform the area integral into a contour integral as is done in the discretization of the governing equations. However, some functions are not amiable to this treatment, and it becomes more convenient to integrate over the area directly. Unfortunately, integration rules are typically given only for simple regions, whereas the median dual construction yields arbitrary shapes. To remedy this situation, the undertaken idea is to subdivide each control volume into a set of constituent triangles. These triangles are formed by connecting each endpoint of the contour edges to the associated vertex of the control volume. Since the integral enjoys the property of additivity on intervals, the integral over the area of the control volume is equivalent to the sum of the integrals over the constituent triangles. The integrals over the constituent triangles is computed numerically using a seven point, fifth-order accurate Gaussian quadrature rule. The quadrature nodes are given in Table 3.4 in terms of barycentric coordinates. The coordinates of the quadrature nodes in computational space are determined by a linear combination of the vertices of the constituent triangle with the barycentric coordinates acting as weights:

$$\vec{x}_{G_i} = \xi_i \, \vec{x}_1 + \eta_i \, \vec{x}_2 + \zeta_i \, \vec{x}_3$$

Furthermore, it is these quadrature nodes which are used in the assessment of error.

Table 3.4 Gaussian quadrature nodes for a triangle

Node	ξ	η	ζ	Weight
1	$1/3$	$1/3$	$1/3$	$9/40$
2	$(6-\sqrt{15})/21$	$(6-\sqrt{15})/21$	$(9+2\sqrt{15})/21$	$(155-\sqrt{15})/1200$
3	$(6-\sqrt{15})/21$	$(9+2\sqrt{15})/21$	$(6-\sqrt{15})/21$	$(155-\sqrt{15})/1200$
4	$(9+2\sqrt{15})/21$	$(6-\sqrt{15})/21$	$(6-\sqrt{15})/21$	$(155-\sqrt{15})/1200$
5	$(6+\sqrt{15})/21$	$(6+\sqrt{15})/21$	$(9-2\sqrt{15})/21$	$(155+\sqrt{15})/1200$
6	$(6+\sqrt{15})/21$	$(9-2\sqrt{15})/21$	$(6+\sqrt{15})/21$	$(155+\sqrt{15})/1200$
7	$(9-2\sqrt{15})/21$	$(6+\sqrt{15})/21$	$(6+\sqrt{15})/21$	$(155+\sqrt{15})/1200$

The results of the first test function, Equation 3.17, are found in Table 3.5. Because the applied function is quadratic, and thus the third and higher derivatives are zero, it is expected that the third- and fourth-order schemes will be able to completely reconstruct the function since the associated truncation error is zero. However, round-off error is still present since the values are computed with finite precision. This behavior is reflected in the errors seen in Table 3.5, where the maximal error grows as the mesh size is correspondingly decreased. The ratio of successive errors is not calculated for these two schemes since the error ratio is meaningless. The data also indicates that the second-order scheme performs as expected; the extrapolation errors are reduced by a factor of four as the nominal grid spacing is reduced by one half.

Table 3.5 Convergence results using Equation 3.17

Quadrilateral	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	6.6634e-03	—	1.4744e-13	—	1.5188e-13	—
1	1.6659e-03	4.00	5.7110e-13	—[*]	6.5725e-13	—[*]
2	4.1464e-04	4.00	2.4247e-12	—[*]	2.7267e-12	—[*]
3	1.0412e-04	4.00	9.4111e-12	—[*]	1.0896e-11	—[*]
Aligned Triangle	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	6.6924e-03	—	1.3589e-13	—	1.4744e-13	—
1	1.6731e-03	4.00	5.1781e-13	—[*]	5.5067e-13	—[*]
2	4.1828e-04	4.00	2.5207e-12	—[*]	2.7089e-12	—[*]
3	1.0457e-04	4.00	8.9688e-12	—[*]	9.7433e-12	—[*]
Unstructured	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	5.2726e-03	—	1.9451e-13	—	2.2293e-13	—
1	1.5218e-03	3.46	7.8870e-13	—[*]	8.1357e-13	—[*]
2	4.5827e-04	3.32	4.4418e-12	—[*]	4.4942e-12	—[*]
3	1.0733e-04	4.27	1.5164e-11	—[*]	1.7272e-11	—[*]

[*] Order is not calculated.

The error results for the remaining test functions from Equations 3.18, 3.19, and 3.20 are given in Tables 3.6, 3.7, and 3.8, respectively. Also note that since Equation 3.18 is cubic, the fourth-order scheme is expected to exhibit the same behavior as with the first function. The tabulated data demonstrates that both the second-order and high-order schemes perform well. In the case of the third-order scheme, the ratio of successive errors is or approaches the value of eight which is expected as the one half scaling of grid size is raised to the power of three (2^3). Similarly, for fourth-order, the ratio of successive errors approach 16 (one half raised to the fourth power). The data also shows that the quadrilateral and aligned triangle grids perform better than the unstructured triangle grids owing to greater control in the interior nominal spacing.

Table 3.6 Convergence results using Equation 3.18

Quadrilateral	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	4.0353e-02	—	5.0925e-03	—	2.3448e-13	—
1	1.0549e-02	3.83	6.3656e-04	8.00	1.0054e-12	—[*]
2	2.6948e-03	3.91	7.9570e-05	8.00	4.1771e-12	—[*]
3	6.8090e-04	3.96	9.9463e-06	8.00	1.7055e-11	—[*]
Aligned Triangle	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	5.2913e-02	—	1.0562e-02	—	2.2737e-13	—
1	1.3644e-02	3.88	1.3203e-03	8.00	8.3755e-13	—[*]
2	3.4631e-03	3.94	1.6504e-04	8.00	4.2819e-12	—[*]
3	8.7226e-04	3.97	2.0630e-05	8.00	1.5256e-11	—[*]
Unstructured	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	2.9875e-02	—	6.0237e-03	—	3.2507e-13	—
1	9.1621e-03	3.26	7.9060e-04	7.62	1.1893e-12	—[*]
2	2.6454e-03	3.46	1.0746e-04	7.36	7.0957e-12	—[*]
3	6.5611e-04	4.03	1.4226e-05	7.55	2.7027e-11	—[*]

[*] Order is not calculated.

The data for the reconstruction schemes found in Table 3.7 is also shown in Figure 3.5 which shows the extrapolation error as a function of boundary spacing. The asymptotic convergence of the error is given next to the corresponding grid type. The slopes of the first two grid series nearly recover the correct value of three for third-order and four for fourth-order while the slope of the unstructured grid series does not agree quite as well, as

Table 3.7 Convergence results using Equation 3.19

Quadrilateral	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	1.7784e-02	—	8.7404e-03	—	1.9896e-03	—
1	4.4525e-03	3.99	1.1138e-03	7.85	1.2672e-04	15.70
2	1.1132e-03	4.00	1.3990e-04	7.96	7.9800e-06	15.88
3	2.7828e-04	4.00	1.7508e-05	7.99	4.9969e-07	15.97
Aligned Triangle	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	2.2846e-02	—	2.3403e-02	—	8.1498e-03	—
1	5.7962e-03	3.94	3.0806e-03	7.60	5.6026e-04	14.55
2	1.4583e-03	3.97	3.9010e-04	7.90	3.5665e-05	15.71
3	3.6516e-04	3.99	4.8921e-05	7.97	2.2393e-06	15.93
Unstructured	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	1.6397e-02	—	9.6389e-03	—	4.3549e-03	—
1	5.2777e-03	3.11	1.3352e-03	7.22	2.4776e-04	17.58
2	1.4361e-03	3.67	2.4533e-04	5.44	1.9559e-05	12.67
3	3.8402e-04	3.74	2.9955e-05	8.19	1.2656e-06	15.46

was seen in the error ratios. However, the slope of the error approaches the correct value as the grid is refined.

Selection of Algorithm for the Least Squares Problem

As mentioned above, the SVD was used to find the derivatives by solving the least squares problem. It seems prudent to discuss why this selection was made over the other two mentioned algorithms, namely QR factorization and Gram-Schmidt orthonormalization. To determine which algorithm had the best performance, a test was devised to measure each method's sensitivity to the condition number of the least squares system. The idea being

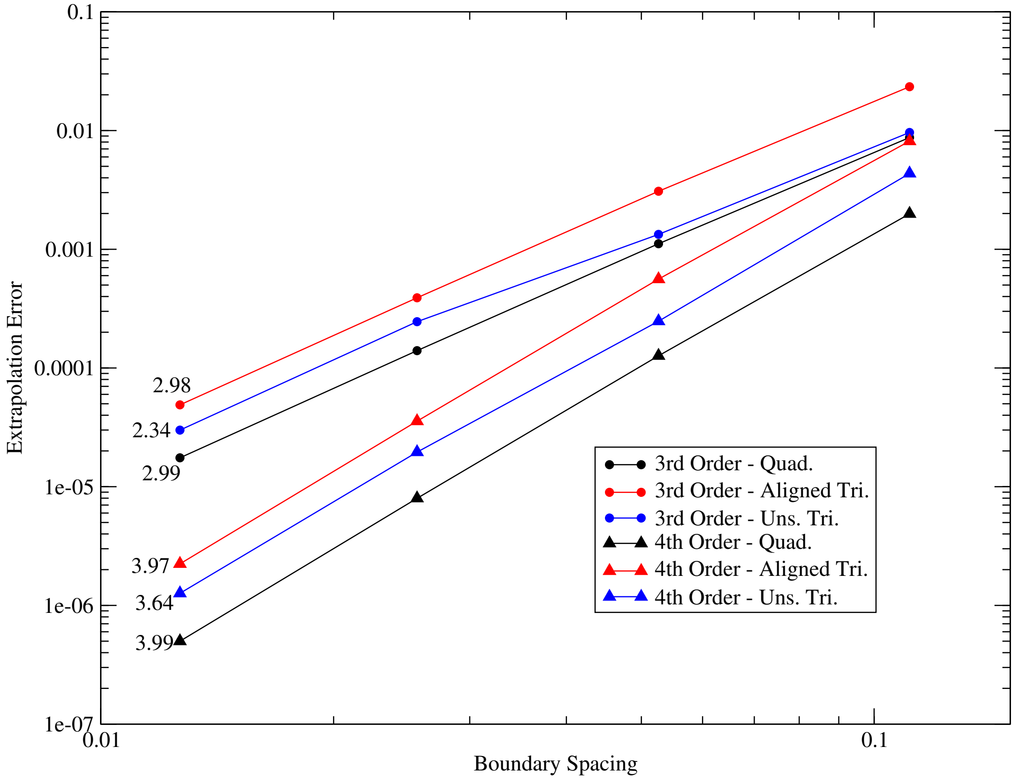


Figure 3.5 Grid convergence plot for Equation 3.19 using all grid types

that the most stable algorithm would find the best approximation to the derivatives. Two sequences of meshes were used to measure the errors from the methods. The first sequence again uses quadrilateral elements to discretize the domain described in the smooth function reconstruction test. Rather than employ uniform refinement, the number of points along the constant y boundaries was frozen at 11. The number of points along the constant x boundaries was then increased in each successive refinement level. The goal was to create meshes that had a specific aspect ratio (measured as the ratio of an element's width to height) of 10, 20, 100, and 1000. In the second sequence of meshes, the structured grids are transformed to aligned triangular grids as in the reconstruction test. The reason this was done was so that the conditioning of the least squares problem, which is only dependent on geometrical properties, would be adversely affected. The condition number

Table 3.8 Convergence results using Equation 3.20

Quadrilateral	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	3.6024e-03	—	1.0034e-03	—	1.8325e-04	—
1	9.0183e-04	3.99	1.5603e-04	6.43	8.3499e-06	21.95
2	2.2553e-04	4.00	2.0189e-05	7.73	5.3484e-07	15.61
3	5.6388e-05	4.00	2.5247e-06	8.00	3.3783e-08	15.83
Aligned Triangle	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	4.5605e-03	—	1.8853e-03	—	4.8907e-04	—
1	1.1370e-03	3.99	2.9638e-04	6.36	2.7505e-05	17.78
2	2.8404e-04	4.00	3.7712e-05	7.86	1.5899e-06	17.30
3	5.6388e-05	4.00	4.7200e-06	7.99	9.9428e-08	15.99
Unstructured	Second-Order		Third-Order		Fourth-Order	
Refinement Level	Error	Ratio	Error	Ratio	Error	Ratio
0	3.2537e-03	—	7.3870e-04	—	2.4422e-04	—
1	9.6067e-04	3.39	1.3950e-04	5.30	1.6232e-05	15.05
2	3.3468e-04	2.87	1.9402e-05	7.19	1.0095e-06	16.08
3	7.2119e-05	4.64	2.5914e-06	7.49	7.1807e-08	14.06

for the least squares problem is shown for in Table 3.9 for all grids and is measured as the ratio of the largest singular value to the smallest singular value [26]. The SVD algorithm was implemented in a function call to the LAPACK library. The QR factorization was accomplished by using Householder reflections and GS uses the standard procedure; both algorithms are discussed in detail by Demmel [26].

The test function used in the results is given by:

$$f(x,y) = 3x^2 + 4y^2 + 10xy + x + 5y \tag{3.21}$$

Table 3.9 Condition number of the matrix arising from the least squares problem

Quadrilateral Aspect Ratio	 Condition Number
10	6.50173e+03
20	2.51543e+04
100	6.15035e+05
1000	6.04002e+07
Aligned Triangle **Aspect Ratio**	 **Condition Number**
10	7.09614e+03
20	2.74811e+04
100	6.72451e+05
1000	6.60871e+07

The test was restricted to look at errors in the third-order scheme with the hope that the results would be indicative of the performance for fourth-order. The derivatives are

$$f_x = 6x + 10y + 1\,,\; f_y = 8y + 10x + 5\,,\; f_{xx} = 6\,,\; f_{yy} = 8\,,\; f_{xy} = 10$$

The error in the derivatives is also examined for a fixed control volume associated with the node at $x = \frac{1}{2}$ and $y = 0$. This control volume lies in the region that has elements with the specified aspect ratio. The analytical derivatives for this node are

$$f_x = 4\,,\; f_y = 10\,,\; f_{xx} = 6\,,\; f_{yy} = 8\,,\; f_{xy} = 10 \tag{3.22}$$

The errors, in absolute value, between the analytical and computed derivatives for the node using all grids and algorithms are given in Table 3.10. While this table looks at only a single node, the results shown in Table 3.11 provide a measure of the error in the derivatives for all nodes.

The errors listed in the two tables reveal that all the methods appear competitive for aspect ratios 10, 20, and 100. When the aspect ratio is at 1000, only the SVD and Gram-Schmidt algorithms perform well for the entire grid. This is a somewhat surprising result since Householder reflectors are backwards stable and the implemented classical GS is unstable [26]. This error might be addressed by ensuring that proper scaling is applied during the procedure. The decision was ultimately made to use the SVD algorithm for all results in this research since it is the most robust and stable method and the choice used in other work [27]. While this test is in no way a complete analysis of the methods, it does help bestow confidence that the SVD will be an acceptable choice for the remaining cases.

Table 3.10 Error in derivatives using Equation 3.21

Quad., SVD					
Aspect Ratio	x	y	xx	yy	xy
10	1.20e-14	6.04e-14	2.40e-14	5.20e-12	1.71e-13
20	7.90e-14	9.06e-14	4.44e-14	5.68e-12	1.07e-13
100	0.00e+00	1.49e-13	1.87e-14	3.05e-10	9.59e-13
1000	3.11e-15	7.44e-12	4.26e-14	3.57e-08	9.81e-12
Quad., QR					
10	2.22e-15	9.06e-14	4.44e-15	7.30e-12	2.10e-13
20	0.00e+00	1.10e-13	3.11e-14	1.07e-11	2.61e-13
100	8.88e-16	2.59e-13	7.99e-15	6.98e-11	1.93e-12
1000	3.65e-09	5.54e-05	1.64e-08	5.33e-01	9.63e-05
Quad., GS					
10	2.22e-15	4.97e-15	1.60e-14	3.64e-12	2.70e-13
20	0.00e+00	1.21e-13	4.71e-14	1.37e-11	1.95e-14
100	8.88e-16	1.81e-13	1.78e-14	3.49e-10	3.88e-12
1000	8.88e-16	6.38e-12	4.62e-14	2.45e-08	2.58e-11
Tri., SVD					
Aspect Ratio	x	y	xx	yy	xy
10	2.66e-15	2.20e-13	4.09e-14	1.14e-11	4.90e-13
20	3.11e-15	5.70e-13	4.44e-14	7.72e-11	5.81e-13
100	3.51e-14	6.80e-13	1.13e-13	5.96e-10	4.71e-13
1000	3.11e-15	1.77e-12	5.33e-15	3.40e-08	1.54e-11
Tri., QR					
10	0.00e+00	2.10e-13	4.80e-14	1.04e-11	4.80e-13
20	3.11e-15	5.81e-13	2.93e-14	7.07e-11	2.81e-13
100	4.88e-15	5.60e-13	7.02e-14	5.50e-10	5.81e-13
1000	3.72e-08	1.25e-05	1.56e-07	4.50e-01	6.04e-04
Tri., GS					
10	8.88e-16	1.90e-13	4.80e-14	7.86e-12	7.00e-13
20	4.88e-15	9.09e-13	2.84e-14	1.36e-10	7.50e-13
100	4.00e-14	3.20e-13	8.62e-14	5.44e-10	3.92e-12
1000	0.00e+00	1.05e-11	4.44e-15	1.24e-07	1.33e-11

Table 3.11 Norms of total error in derivatives using Equation 3.21

Quad., SVD Aspect Ratio	L_1	L_2	L_∞
10	7.8786e-09	1.1719e-09	4.9001e-10
20	1.6813e-08	2.5931e-09	1.1542e-09
100	2.6217e-07	6.5015e-08	3.6521e-08
1000	1.4269e-05	3.8959e-06	2.0522e-06
Quad., QR			
10	7.8789e-09	1.1707e-09	4.9001e-10
20	1.6799e-08	2.5927e-09	1.1589e-09
100	2.6293e-07	6.5371e-08	3.6723e-08
1000	3.3600e+02	2.0866e+02	1.7873e+02
Quad., GS			
10	7.9267e-09	1.1800e-09	4.9018e-10
20	1.6820e-08	2.5782e-09	1.1267e-09
100	2.6431e-07	6.5441e-08	3.7319e-08
1000	1.4288e-05	3.8823e-06	2.0508e-06
Tri., SVD Aspect Ratio	L_1	L_2	L_∞
10	7.1313e-09	9.5147e-10	4.5356e-10
20	2.1673e-08	4.5517e-09	3.5099e-09
100	2.4985e-07	5.1380e-08	2.8622e-08
1000	2.1663e-05	4.9313e-06	2.8599e-06
Tri., QR			
10	7.1250e-09	9.5110e-10	4.5355e-10
20	2.1639e-08	4.5241e-09	3.4735e-09
100	2.4910e-07	5.1316e-08	2.8582e-08
1000	2.0594e+03	2.5973e+02	1.0519e+02
Tri., GS			
10	7.0971e-09	9.5122e-10	4.5379e-10
20	2.1890e-08	4.5688e-09	3.5170e-09
100	2.4611e-07	5.0659e-08	2.7682e-08
1000	2.1804e-05	4.9591e-06	2.9074e-06

Solution Monotonicity

The discussion of the reconstruction has yet to address a critical problem that has plagued high-order methods; the introduction of oscillations in the solution when discontinuities exist in the flow field. The ENO and WENO methods discussed in Chapter 1 were able to overcome this problem since they utilize more than one stencil and can thus either select the smoothest stencil, as in the case of ENO methods, or assign very small weights to non-smooth stencils, in the case of WENO methods. The method studied in this research is restricted to a fixed stencil of cells that are added by only considering geometrical and graph relationships. To maintain solution monotonicity, a gradient, or slope, limiter is employed as suggested in [17, 28]. A full discussion on limiters is omitted; details of the implemented limiters are provided for brevity.

The use of slope limiters necessitates a change in the extrapolation of the solution variables. The generic extrapolation of a variable to a quadrature node can be written as

$$u_G = u(\vec{x}_i) + S(\vec{x}_G - \vec{x}_i) + H(\vec{x}_G - \vec{x}_i), \tag{3.23}$$

where S is the contribution of the gradient and H represents the collected contribution of higher derivatives. The inclusion of a limiter changes this to

$$u_G = u(\vec{x}_i) + \phi_i(S(\vec{x}_G - \vec{x}_i) + H(\vec{x}_G - \vec{x}_i)), \phi \in [0, 1] \tag{3.24}$$

This idea was first introduced for unstructured grids by Barth and Jespersen [29]. The intent is to find the largest admissible value of ϕ such that no new local extrema are created at the quadrature nodes during extrapolation. The algorithm developed by Barth and Jespersen can be enumerated for a cell i as:

1. Find the largest CV average value of u in the stencil of first degree neighbors of cell i, including itself, denoted u_i^{max}. Similarly, find u_i^{min}.

2. Compute the unlimited reconstruction value of u at every quadrature node j of cell i, u_{ij}.

3. For each quadrature node, find the value of the limiter ϕ:

$$\hat{\phi}_{ij} = \begin{cases} \min(1, \frac{u_i^{max}-u_i}{u_{ij}-u_i}), & \text{if } u_{ij}-u_i > 0 \\ \min(1, \frac{u_i^{min}-u_i}{u_{ij}-u_i}), & \text{if } u_{ij}-u_i < 0 \\ 1, & \text{if } u_{ij}-u_i = 0 \end{cases}$$

4. $\phi_i = \min(\hat{\phi}_{ij})$.

The use of the min function in the above limiter procedure makes the reconstruction non-differentiable. This has the negative effect of stalling convergence to steady state. This drawback was studied by Venkatakrishnan [30] who introduced a replacement to the function $\min(1, y)$ with

$$\theta(y) = \frac{y^2 + 2y}{y^2 + y + 2} \tag{3.25}$$

Venkatakrishnan further suggested a modification that was aimed at restricting the use of the limiter in regions where the flow is approximately uniform. Following the notation in [28], this can be written as:

$$\hat{\phi}_{ij} = \begin{cases} \frac{1}{\Delta_2}\left[\frac{(\Delta_{1,\max}^2 + \varepsilon^2)\Delta_2 + 2\Delta_2^2\Delta_{1,\max}}{\Delta_{1,\max}^2 + 2\Delta_2^2 + \Delta_{1,\max}\Delta_2 + \varepsilon^2}\right] & \text{if } \Delta_2 > 0 \\ \frac{1}{\Delta_2}\left[\frac{(\Delta_{1,\min}^2 + \varepsilon^2)\Delta_2 + 2\Delta_2^2\Delta_{1,\min}}{\Delta_{1,\min}^2 + 2\Delta_2^2 + \Delta_{1,\min}\Delta_2 + \varepsilon^2}\right] & \text{if } \Delta_2 < 0 \end{cases} \tag{3.26}$$

where,

$$\Delta_2 = u_{ij} - u_i \, , \, \Delta_{1,\max} = u_i^{max} - u_i \, , \, \Delta_{1,\min} = u^{min} - u_i \tag{3.27}$$

$$\varepsilon^2 = (K\triangle x)^2 \tag{3.28}$$

In the above, $\triangle x$ is a measure of local length scale for the CV. There are several choices that can be made for the local length scale. The method adopted in this research is to associate with each control volume an equilateral triangle that has the same area and whose sides are $\triangle x$. K is a parameter that determines how strictly monotonicity is enforced in the solution. Large values of K effectively turn off limiting. A value of zero recovers the limiter of Barth and Jespersen with Equation 3.25 replacing the min function in the procedure.

Nejat reported that the application of the limiter to the full set of derivatives in the extrapolation phase lead to a more diffusive solution [28]. The approach adopted in the work is to further modify the reconstruction to

$$u_G = u(\vec{x}_i) + ((1-\sigma_i)\phi_i + \sigma_i)S(\vec{x}_G - \vec{x}_i) + \sigma_i(H(\vec{x}_G - \vec{x}_i))) \tag{3.29}$$

The goal of this approach is to have σ_i be zero near discontinuities and approach one in smooth regions of the flow. The prescribed form of σ is given as a differentiable semi-step function,

$$\sigma = \frac{1 - \tanh(S(\phi_0 - \phi))}{2} \tag{3.30}$$

where S determines the slope of the step transition and ϕ_0 signifies the limiter value that activates the switch. Values of $\phi_0 = 0.8$ and $S = 20$ were given as reasonable values to use.

Later, Michalak and Ollivier-Gooch [17] presented a new replacement to the min function used in Barth and Jespersen's limiter since they showed that for third- and fourth-order schemes Venkatakrishnan's limiter did not provide sufficient accuracy. The new supplied function is of the form

$$\widetilde{\min} = \begin{cases} P(y) & y < y_t \\ 1 & y \geq y_t \end{cases} \tag{3.31}$$

where $y_t \in (1, 2)$ is a threshold and $P(y)$ is a cubic polynomial subject to the following conditions:

$$P|_0 = 0\,,\; P|_{y_t} = 1\,,\; \left.\frac{dP}{dy}\right|_0 = 1\,,\; \left.\frac{dP}{dy}\right|_{y_t} = 0$$

As in [28], Equation 3.29 is used for the extrapolation; however, a new function is provided to compute the high-order limiter as

$$\sigma_i = \begin{cases} 1 & \delta u^2 \leq (K\Delta x)^3 \\ s\left(\frac{\delta u^2 - (K\Delta x)^3}{(K\Delta x)^3}\right) & (K\Delta x)^3 < \delta u^2 < 2(K\Delta x)^3 \\ 0 & \delta u^2 \geq 2(K\Delta x)^3 \end{cases} \tag{3.32}$$

where s is a transition function to smoothly vary from zero to one defined by

$$s(y) = 2y^3 - 3y^2 + 1 \tag{3.33}$$

All of the aforementioned limiters have been implemented in the flow solver for this research.

CHAPTER 4
METHOD OF MANUFACTURED SOLUTIONS

A brief description of the Method of Manufactured Solutions (MMS) is offered in this chapter. A thorough discussion of the method and its applications can be found in the report by Salari and Knupp [31]. The driving idea behind MMS is to find the order of accuracy of a numerical code in solving a system of PDEs by supplying a known solution that is continuous and differentiable. The supplied (manufactured) solution should posses enough derivatives so that all terms in a scheme are fully exercised. In this research that considers up to fourth-order accuracy, the manufactured solution would need at least the third derivatives to be non-zero so that all terms in the extrapolation are utilized. An advantage that MMS has over traditional verification methods is that the manufactured solution given to the code does not have to be a physically realistic one. This means that one is free to select a solution that may be as simple as polynomials, exponentials, or trigonometric functions. However, one caveat exists in the case of a solver for the Euler equations; the supplied solution should respect the positivity of density and pressure so that the numerical flux can properly evolve the solution.

The implementation of the MMS to the flow solver in this research follows the work of Murali [32] and Burg and Murali [33]. This entails specifying a set of functions that describe the conserved, or primitive, variables. These functions are then integrated over the control volume and are then set to be the exact solution, $\vec{Q}^E$, after dividing through by the area of the CV. The manufactured solutions are then used to generate a source term so that the

Euler equations may be rewritten as

$$\frac{\partial}{\partial t}\int_{CV}\vec{Q}\,dV + \oint_{CS}\vec{F}\cdot\hat{n}\,d\vec{A} = S(x,y)\,, \tag{4.1}$$

The source term in the above equation is computed in much the same way as the residual is calculated in Chapter 2. Thus, the source term for CV i is computed as

$$S(x,y) = \oint_{CS_i}\vec{F}(\vec{Q}^E)\cdot\hat{n}\,d\vec{A}_i \tag{4.2}$$

where the flux is computed using the exact solution. Whereas the work in [32, 33] used exponentials for the manufactured solution allowing the contour integral in Equation 4.2 to be readily calculated, this research uses a modified set of supplied functions following closely to those described by Erwin et al [34]. The manufactured solutions used are

$$\begin{aligned} \rho &= 1 + \tfrac{1}{4}\mathrm{sin}(\pi x)\mathrm{sin}(\pi y) \\ u &= \tfrac{1}{4} + \tfrac{1}{4}\mathrm{sin}(\pi x)\mathrm{cos}(2\pi y) \\ v &= \tfrac{1}{4} + \tfrac{1}{4}\mathrm{cos}(2\pi x)\mathrm{sin}(\pi y) \\ P &= \tfrac{1}{\gamma} + \tfrac{1}{20}\mathrm{cos}(2\pi x)\mathrm{cos}(2\pi y) \end{aligned} \tag{4.3}$$

These equations are infinitely differentiable and bounded so they are valid on any computational domain in two dimensions. However, they are more difficult to resolve using a contour integral so the Divergence theorem is used to formulate them for integration over the area. The integration employs a fifth-order Gaussian quadrature scheme over the constituent triangles. This is expressed as

$$\iint_{CV_i}\nabla\cdot\vec{F}dV_i \approx \sum_{j=1}^{n} W_j\left(\nabla\cdot\vec{F}(\vec{x_j})\right)A_i \tag{4.4}$$

The source term may now be viewed as a residual computation associated with the manufactured solution and, since they do not have any temporal dependency, are independent of the iteration. Therefore, it is sufficient to calculate the source residual once, at the beginning of the solve. The solution algorithm can be restated as

$$\left[\frac{V_i}{\triangle t}I + \frac{\partial \Re^m}{\partial Q}\right] \triangle \overline{Q} = -\Re(\overline{Q}^m) + \Re(\vec{Q}^E) \tag{4.5}$$

The solution process starts by first computing the exact solution which is also used as the initial condition to the solver (the process will still drive the initial solution to the proper numerical solution). Next, the source term is computed and stored for every control volume. The solver then follows the procedure given in Chapter 2; the only modification is that the source term is now added to the right hand side. All boundaries are treated as farfield and all ghost nodes associated with the boundary nodes are given the exact solution as their state using the boundary nodes' coordinates. The error between the numerical and exact solutions can be assessed in a normed sense using

$$L_p = \left(\frac{\sum_{i=1}^{N_{CV}} A_i \left|\overline{E}_i^p\right|}{\sum_{i=1}^{N_{CV}} A_i}\right)^{\frac{1}{p}} \tag{4.6}$$

$$\overline{E}_i = \frac{1}{A_i}\iint_{CV_i} \left(Q^E - R(x,y)\right) dA_i \tag{4.7}$$

In Equation 4.7, the integration is approximated by the same Gaussian quadrature scheme on the constituent triangles and $R(x,y)$ denotes the reconstruction function. The results also make use of the infinity norm which is a measure of local accuracy and is represented by

$$L_\infty = \max\left|\overline{E}_i\right| \tag{4.8}$$

The observed order of accuracy of the solver can be calculated between two successive grids with a known refinement using

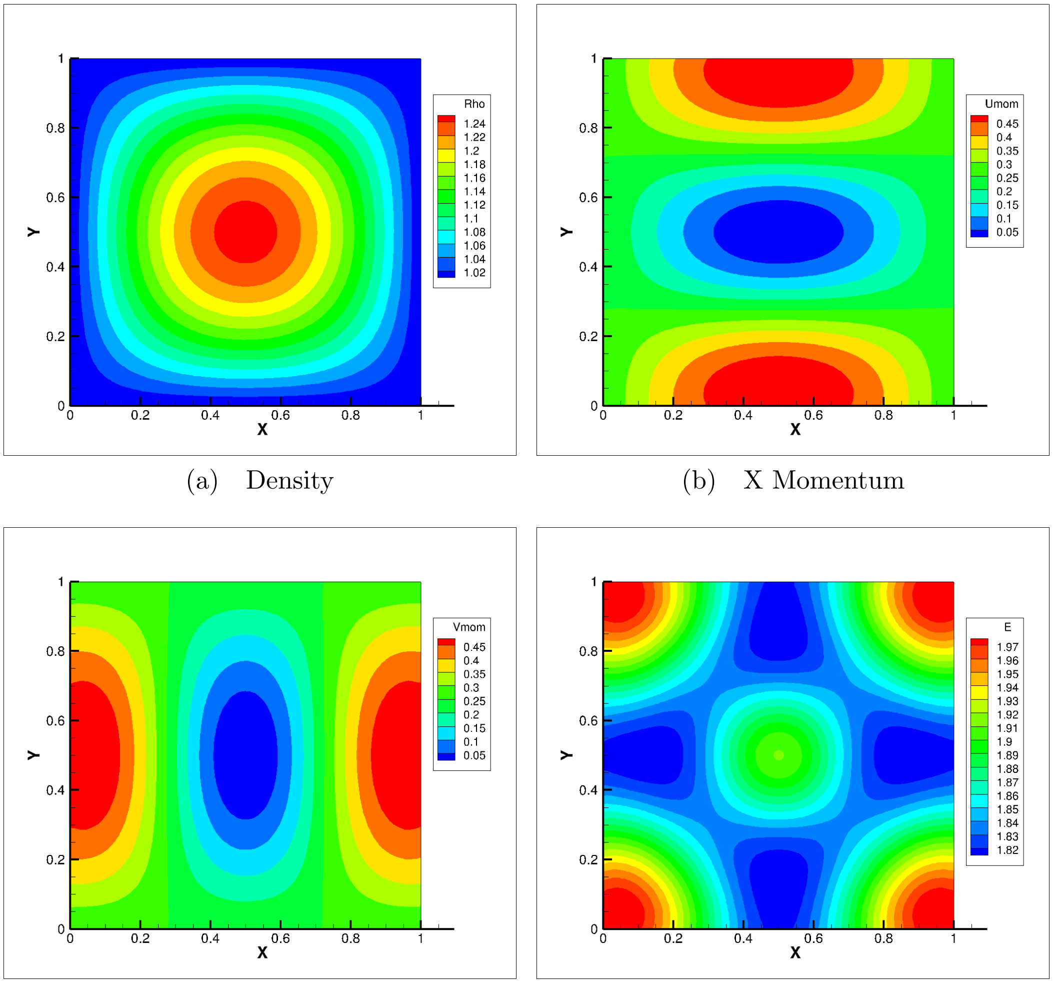
$$k = \log_{10}\left(\frac{L_{p,c}}{L_{p,f}}\right)/\log_{10}(r) \tag{4.9}$$

where c is the coarse mesh, f the fine mesh, and r is the refinement factor.

Linear Boundary Example

The first set of grids that was examined is the same set of grids used in the verification of the reconstruction algorithm on smooth functions from Chapter 3. The MMS was used to determine the observed order of accuracy for second-, third-, and fourth-order accurate reconstructions. Another method that was tested was to use the second-order scheme and assume that the solution varied quadratically along the edge connecting a node to its first degree neighbor. This method will be denoted as pseudo third-order since it attempts to account for $O(h^2)$ perturbations in an a posteriori fashion and does not perform high-order quadrature. The exact solution for all variables on this domain can be seen in Figure 4.1. The results for the three series of grids and all orders is tabulated below. The errors are calculated for density, x momentum, and total energy. Y momentum is omitted since the results for it follow closely to the results for x momentum, due to symmetry in the equations discretized on the grids. In all cases of third- and fourth-order solutions, the parameter controlling the influence of geometrical distance in the least squares problem was set to zero (no distance weighting). The refinement factor is 2 for these sequences of grids. As noted in Chapter 3, this is not strictly correct for the unstructured grid sequence since the refinement is not uniform due to the way in which the software generates these grids.

The results for the second-order scheme in Table 4.1 show that it recovers the expected order of accuracy in the L_1 norm. The order seen from the L_∞ errors, given in Table 4.2, show that the recovered order approaches 1 asymptotically. This issue is related to how the



(a) Density

(b) X Momentum

(c) Y Momentum

(d) Total Energy

Figure 4.1 Contours of the exact solution variables for the square domain

boundary conditions are implemented for this scheme. Here, the flux along boundaries is computed at the nodes' themselves and not to an extrapolated quadrature node, which limits the performance to first-order. The norms of the error for the pseudo third-order scheme, Tables 4.3 and 4.4, show that the order of accuracy follows closely to that of the second-order scheme, though it seems to suffer more degradation in performance at the boundaries. This can be explained partially by the fact that it does not do high-order flux integration and is thus limited to second-order convergence. Results from the third-order scheme, Tables

4.5 and 4.6, and from the fourth-order scheme, Tables 4.7 and 4.8, demonstrate that they converge nicely to the expected order of accuracy. Note that they do not suffer as poorly as the second-order and pseudo third-order schemes in the L_∞ norm since the flux along the boundaries is computed at quadrature nodes using high-order reconstructed values. However, the L_∞ norm results from the unstructured grid sequence do not show as clear a trend as the other grid types do for the third- and fourth-order schemes. From Tables 4.6 and 4.8, the observed order drops as the grid is refined from index one to two, particularly in the density and momentum errors. Going from two to three, both schemes are able to recover a better observed order, though the fourth-order scheme fails to recover as gracefully as the third-order scheme. An additional level of refinement was subsequently tested but failed to settle the observed orders into a clear trend. It was therefore concluded that the origination of this vacillation was in the nature of the unstructured grids and how they are generated. Specifically, the size of interior triangles, while influenced by boundary spacing, is not guaranteed to be uniform across the domain. Further, the L_∞ norm is especially sensitive to disparate element sizes as it is a measure of local error. The L_1 norms for all orders on the aligned triangular grid sequence is shown in Figure 4.2 for density. The convergence of the third-order scheme appears to be slightly anemic since it follows closely the performance of the second-order and pseudo third-order schemes. From Tables 4.1 and 4.3, it is seen that for density these two schemes exhibit better than expected orders of accuracy.

Table 4.1 L_1 norm of error using the second-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	4.37e-03	—	4.78e-03	—	3.63e-03	—
1	9.64e-04	2.18	1.34e-03	1.83	9.80e-04	1.89
2	1.67e-04	2.53	3.19e-04	2.08	2.41e-04	2.02
3	3.38e-05	2.30	7.46e-05	2.10	5.83e-05	2.05
Tri.,	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	7.81e-03	—	8.67e-03	—	1.33e-02	—
1	1.40e-03	2.48	1.84e-03	2.24	2.28e-03	2.55
2	2.02e-04	2.79	4.21e-04	2.13	3.66e-04	2.64
3	3.57e-05	2.50	1.01e-04	2.05	8.32e-05	2.14
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.88e-03	—	6.59e-03	—	5.16e-03	—
1	5.56e-04	2.38	1.25e-03	2.40	9.52e-04	2.44
2	1.17e-04	2.25	2.85e-04	2.13	2.18e-04	2.13
3	3.21e-05	1.87	7.43e-05	1.94	5.11e-05	2.10

Table 4.2 L_∞ norm of error using the second-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.91e-02	—	1.68e-02	—	1.44e-02	—
1	4.97e-03	1.94	4.78e-03	1.81	3.00e-03	2.26
2	1.48e-03	1.75	1.18e-03	2.02	9.67e-04	1.63
3	6.70e-04	1.14	5.30e-04	1.15	3.98e-04	1.28
Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.12e-02	—	3.60e-02	—	3.79e-02	—
1	6.21e-03	1.77	8.17e-03	2.14	7.43e-03	2.35
2	1.81e-03	1.78	1.80e-03	2.18	1.57e-03	2.24
3	7.44e-04	1.28	5.38e-04	1.74	3.66e-04	2.11
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	9.28e-03	—	2.07e-02	—	1.76e-02	—
1	2.60e-03	1.83	7.25e-03	1.51	4.21e-03	2.07
2	1.23e-03	1.07	2.49e-03	1.54	1.03e-03	2.03
3	9.36e-04	0.40	1.09e-03	1.19	4.38e-04	1.23

Table 4.3 L_1 norm of error using the pseudo third-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	4.35e-03	—	3.81e-03	—	3.93e-03	—
1	9.93e-04	2.13	9.24e-04	2.04	9.00e-04	2.13
2	2.38e-04	2.06	2.08e-04	2.15	2.02e-04	2.15
3	5.86e-05	2.02	4.91e-05	2.08	4.73e-05	2.10

Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	6.33e-03	—	6.92e-03	—	9.37e-03	—
1	1.03e-03	2.62	1.28e-03	2.44	1.36e-03	2.78
2	1.85e-04	2.47	2.60e-04	2.30	2.00e-04	2.77
3	3.92e-05	2.24	6.31e-05	2.04	3.50e-05	2.52

Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.54e-03	—	3.11e-03	—	3.47e-03	—
1	4.87e-04	2.38	6.36e-04	2.29	6.47e-04	2.42
2	1.27e-04	1.93	1.52e-04	2.06	1.37e-04	2.24
3	3.24e-05	1.97	5.27e-05	1.53	3.46e-05	1.99

Table 4.4 L_∞ norm of error using the pseudo third-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.84e-02	—	1.59e-02	—	1.20e-02	—
1	5.15e-03	1.84	3.65e-03	2.13	2.56e-03	2.23
2	1.50e-03	1.78	8.43e-04	2.12	6.80e-04	1.91
3	4.57e-04	1.71	2.06e-04	2.03	1.71e-04	1.99
Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.88e-02	—	2.75e-02	—	2.15e-02	—
1	5.17e-03	1.86	5.49e-03	2.32	4.72e-03	2.19
2	1.85e-03	1.48	1.20e-03	2.19	1.04e-03	2.18
3	8.88e-04	1.06	3.30e-04	1.87	2.57e-04	2.02
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	9.67e-03	—	1.31e-02	—	1.23e-02	—
1	2.80e-03	1.79	3.95e-03	1.73	3.80e-03	1.70
2	4.04e-03	-0.53	2.10e-03	0.91	9.32e-04	2.03
3	4.07e-03	-1.13e-02	1.13e-03	0.89	3.86e-04	1.27

Table 4.5 L_1 norm of error using the third-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	4.31e-03	—	6.82e-03	—	7.97e-03	—
1	7.43e-04	2.54	1.17e-03	2.54	1.27e-03	2.65
2	1.02e-04	2.87	1.61e-04	2.86	1.65e-04	2.94
3	1.28e-05	2.98	2.10e-05	2.94	2.09e-05	2.98
Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.15e-02	—	1.24e-02	—	2.18e-02	—
1	2.13e-03	2.44	2.08e-03	2.57	3.87e-03	2.50
2	3.08e-04	2.79	3.02e-04	2.79	5.50e-04	2.81
3	4.18e-05	2.88	4.10e-05	2.88	7.28e-05	2.92
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	4.43e-03	—	7.88e-03	—	1.00e-02	—
1	6.33e-04	2.81	1.04e-03	2.92	1.30e-03	2.95
2	9.12e-05	2.79	1.98e-04	2.39	1.94e-04	2.75
3	1.16e-05	2.97	2.25e-05	3.14	2.37e-05	3.03

Table 4.6 L_∞ norm of error using the third-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.96e-02	—	2.03e-02	—	2.16e-02	—
1	3.97e-03	2.30	3.49e-03	2.54	3.48e-03	2.63
2	7.75e-04	2.36	4.97e-04	2.81	5.15e-04	2.76
3	1.10e-04	2.82	6.82e-05	2.87	6.95e-05	2.89
Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.90e-02	—	2.95e-02	—	5.09e-02	—
1	4.78e-03	2.60	6.39e-03	2.21	8.12e-03	2.65
2	7.49e-04	2.67	1.09e-03	2.55	1.10e-03	2.88
3	1.10e-04	2.77	1.57e-04	2.81	1.48e-04	2.90
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.16e-02	—	2.35e-02	—	3.02e-02	—
1	2.47e-03	2.23	3.78e-03	2.63	3.94e-03	2.93
2	7.48e-04	1.72	1.11e-03	1.77	6.21e-04	2.67
3	1.03e-04	2.86	1.05e-04	3.40	8.43e-05	2.88

Table 4.7 L_1 norm of error using the fourth-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.64e-03	—	2.54e-03	—	2.37e-03	—
1	1.22e-04	3.75	1.02e-04	4.64	9.98e-05	4.57
2	6.89e-06	4.15	5.94e-06	4.10	6.05e-06	4.05
3	2.81e-07	4.62	3.57e-07	4.06	4.09e-07	3.88
Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	7.66e-03	—	5.37e-03	—	1.16e-02	—
1	5.61e-04	3.77	5.60e-04	3.26	8.64e-04	3.74
2	2.23e-05	4.65	2.63e-05	4.41	3.42e-05	4.66
3	7.86e-07	4.82	1.22e-06	4.43	1.32e-06	4.69
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.92e-03	—	3.09e-03	—	2.39e-03	—
1	1.24e-04	3.96	1.57e-04	4.29	1.17e-04	4.36
2	7.10e-06	4.12	1.25e-05	3.65	7.14e-06	4.03
3	3.99e-07	4.15	1.49e-06	3.07	5.50e-07	3.70

Table 4.8 L_∞ norm of error using the fourth-order scheme on the square domain

Quad.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	5.02e-03	—	8.04e-03	—	6.78e-03	—
1	7.57e-04	2.73	4.40e-04	4.19	3.70e-04	4.20
2	9.23e-05	3.04	2.64e-05	4.06	2.79e-05	3.73
3	8.00e-06	3.53	1.51e-06	4.13	1.72e-06	4.02
Tri.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.16e-02	—	1.65e-02	—	2.58e-02	—
1	1.57e-03	3.78	2.29e-03	2.85	2.33e-03	3.47
2	7.99e-05	4.30	1.36e-04	4.07	1.44e-04	4.01
3	3.96e-06	4.33	6.79e-06	4.32	7.61e-06	4.24
Uns.	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	7.19e-03	—	1.05e-02	—	8.22e-03	—
1	6.82e-04	3.40	7.61e-04	3.78	4.79e-04	4.10
2	1.78e-04	1.94	2.36e-04	1.69	3.87e-05	3.63
3	1.75e-05	3.35	4.05e-05	2.54	4.34e-06	3.16

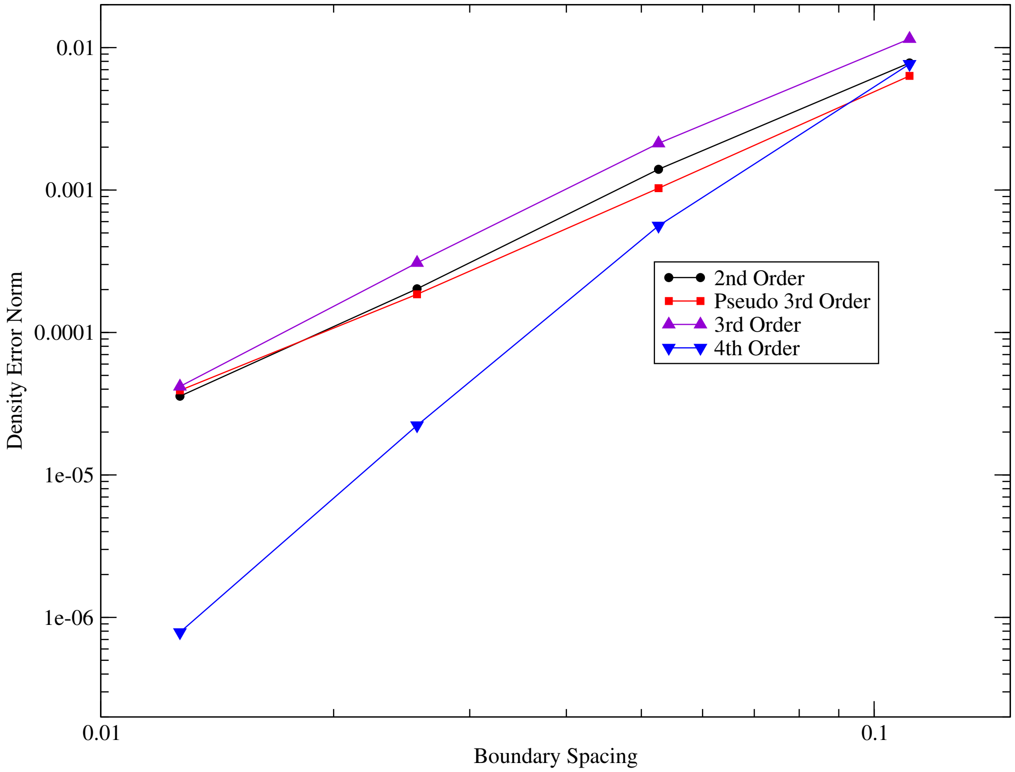


Figure 4.2 L_1 norm of density error for all orders on the aligned triangles grids

Curved Boundary Example

The next application of the MMS was to a sequence of grids using an annular region as the domain. The annulus used was from an interior radius of two to an outer radius of three in the first quadrant of the Cartesian plane in two dimensions. To ensure strict adherence to uniform refinement, only grids containing aligned triangular elements were considered. The initial grid in the study has ten points distributed in the radial direction and forty-two points distributed in the angular direction. This coarsest mesh can be seen in Figure 4.3. As in the linear grids, there are three subsequent grids, each of which has twice the resolution as the previous grid. The same four spatial accuracy schemes were again employed in the test with the same test functions described in Equation 4.3. For the third- and fourth-order schemes, the geometric weighting parameter is set at zero. As discussed in Chapter 3, the quadrature nodes needed for the high-order flux integration along the boundaries are placed correctly on the curved boundary. This is a relatively easy process since the circular segments of the boundary can be readily parameterized using the angle, which varies linearly along the arc segments, in place of the coordinates in Equation 3.16. The coordinates of the quadrature nodes in computational space are then found using polar relationships.

A more complex consequence of utilizing curved boundaries is that the constituent triangles of control volumes that are adjacent to the boundary will now contain one curved side and thus the Gaussian quadrature nodes for integration over the area will be incorrectly placed if this is not accounted for. Recall that this integral arises in the computation of the CV averaged values of the variables, the calculation of the source terms, and the integral of the error over the CV. An exaggerated illustration of this scenario can be seen in Figure 4.4. Here, the dotted lines represent the boundary of the control volume, the blue line is the curved boundary, and the dashed line the other side of the constituent triangles. The goal now is to replace the red, dotted lines that are linear segments with a better approximation of the actual curved boundary so that the area integrals can be correctly computed. This problem has also arisen in the application of the Finite-Element Method (FEM) when modeling curved surfaces. The approach used in this research follows the procedure outlined by Johnson [35] to apply the idea of isoparametric elements. Here, the linear segment on the boundary is replaced with a quadratic curve with the constraint that it pass through the both endpoints of the linear segment and the halfway point of the curved segment (in terms of arc length); all three mentioned points lie exactly on the curved boundary. The quadrature nodes (given in terms of barycentric coordinates) are first mapped to a reference element and then the quadratic transformation is used to map them onto the element in computational space. While a quadratic curve does not exactly model the arc, the hope is that the errors from this approximation are minimal and will decrease as the grid is refined.

The exact solution for this domain in all variables is shown in Figure 4.5. The results from the sequence of grids and all schemes is again tabulated in proceeding tables. The results are very similar to those given on the test using the square domain. The L_1 errors shown in Table 4.9 indicate that the second-order and pseudo third-order schemes initially out perform their theoretical orders of accuracy but asymptotically return to order two. The third-order scheme starts off below an order of three but recovers the expected order as the grid is refined. The fourth-order scheme starts off strong and continues to perform better than its theoretical order of accuracy. The L_∞ norms given in Table 4.10 demonstrate similar conclusions, though all schemes have somewhat more difficult time especially with density. The results for the third- and fourth-order schemes verify that the scheme has been properly implemented. The L_1 error norms depicted in Figure 4.6 graphically visualize the convergence from all four schemes for density.

In Chapter 3, it was stated that the higher order method studied in this work needed a similarly high-order flux integration scheme. This necessity can be demonstrated with the aid of the MMS. The code was modified so that the flux is integrated using a single quadrature node, just as in the case of the second-order scheme. The results of the L_1 error

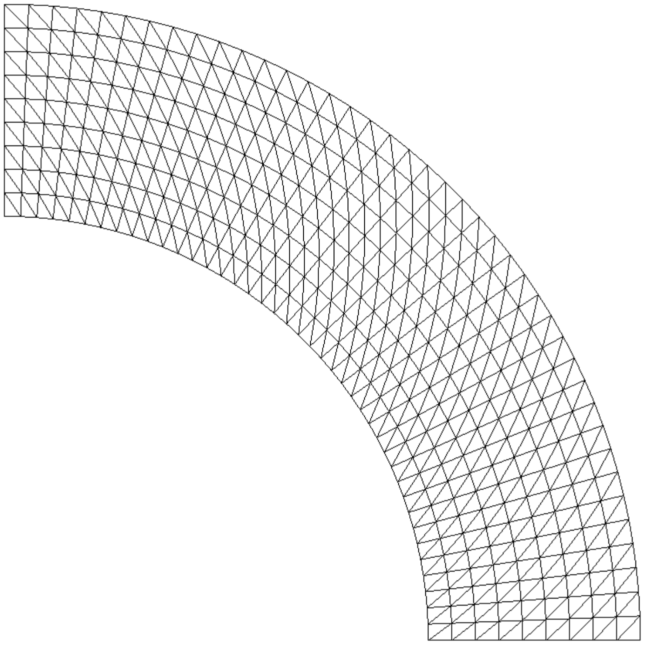


Figure 4.3 Coarsest mesh of the annulus used in the MMS test

norm for all variables are presented in Table 4.11. In theory, both schemes should degrade to second-order convergence, since they become limited by the order of the integration scheme; however, the third order scheme seems to recover a slightly better order of accuracy.

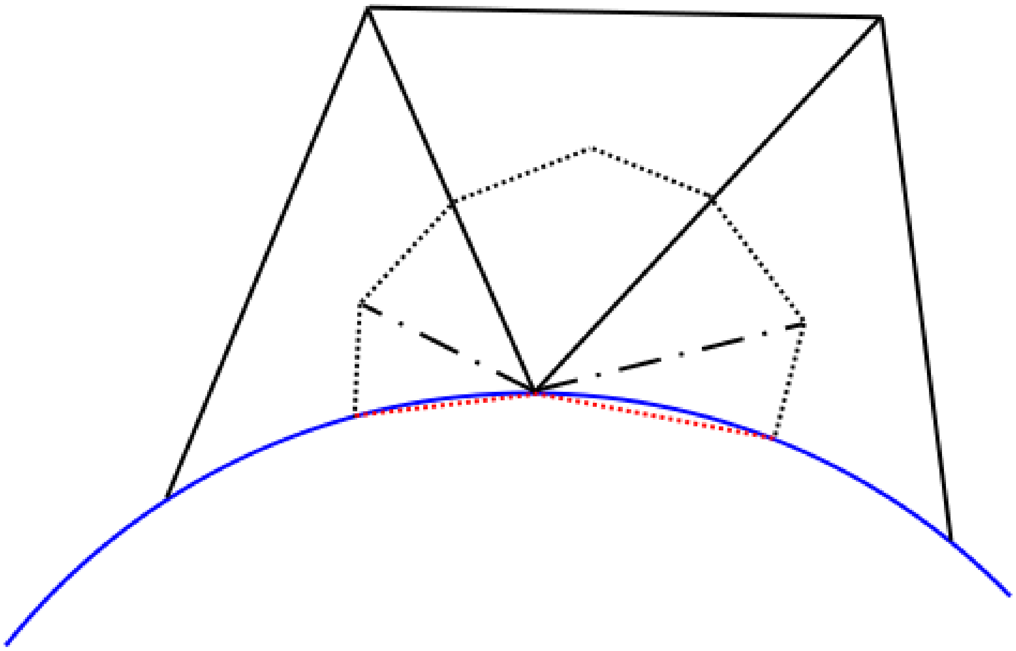
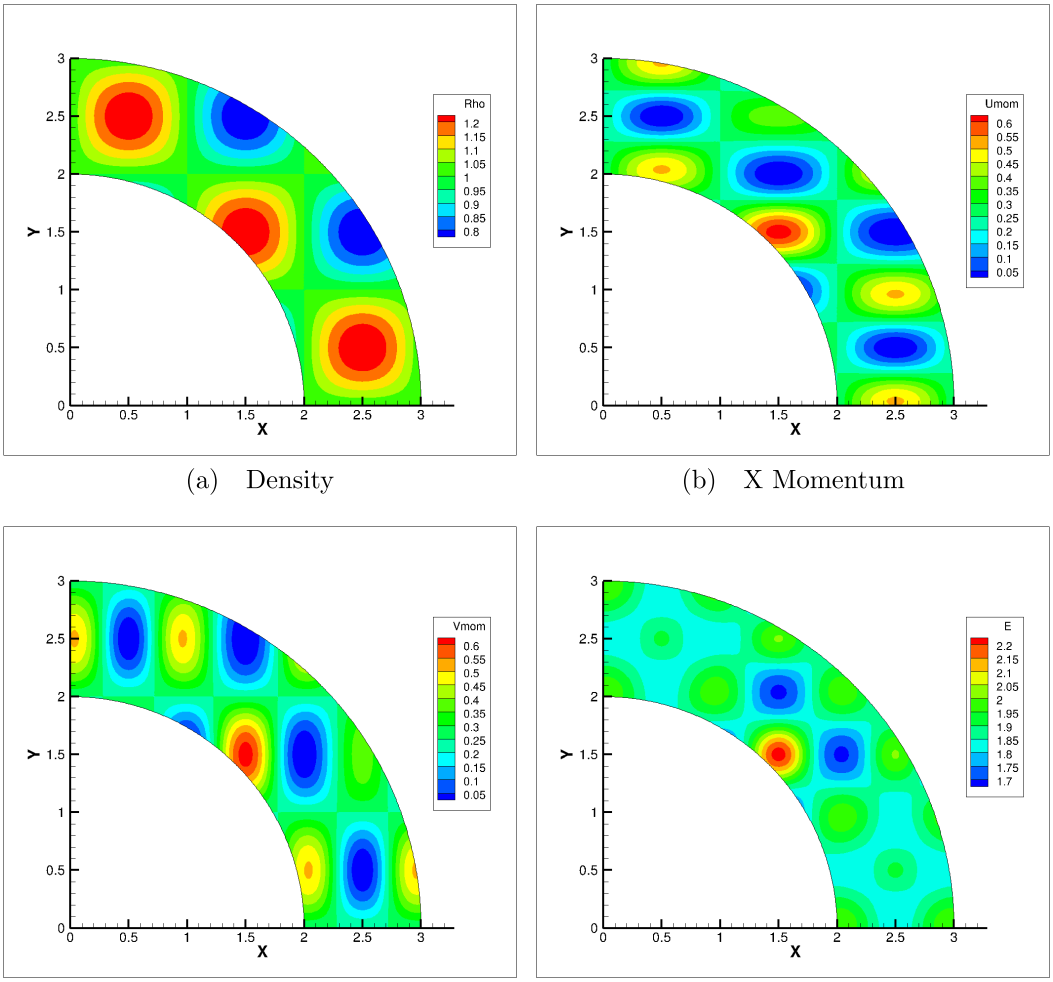


Figure 4.4 Depiction of the curved sides of two constituent triangles



(a) Density

(b) X Momentum

(c) Y Momentum

(d) Total Energy

Figure 4.5 Contours of the exact solution variables for the annular domain

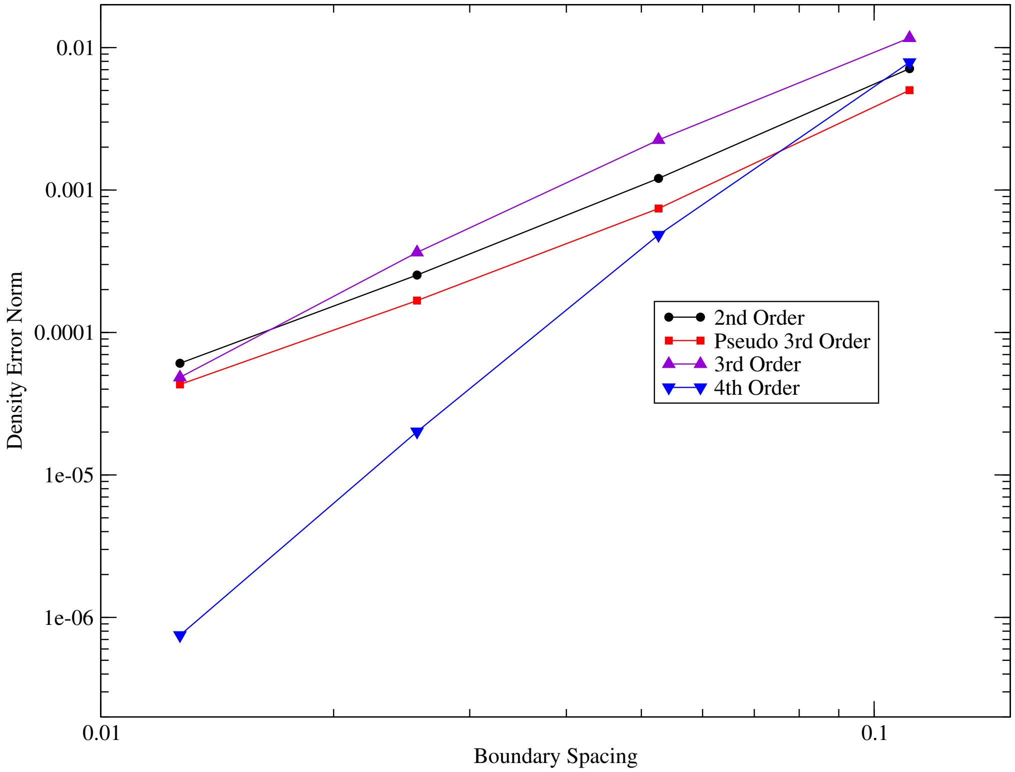


Figure 4.6 L_1 norm of density error for all orders on the annulus grids

Table 4.9 L_1 norm of error for all schemes on the annulus

2^{nd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	7.11e-03	—	1.56e-02	—	1.22e-02	—
1	1.21e-03	2.56	2.83e-03	2.46	2.23e-03	2.45
2	2.53e-04	2.25	5.29e-04	2.42	4.23e-04	2.40
3	6.08e-05	2.06	1.12e-04	2.23	9.01e-05	2.23
pseudo 3^{rd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	5.02e-03	—	1.00e-02	—	8.56e-03	—
1	7.42e-04	2.76	1.50e-03	2.74	1.30e-03	2.72
2	1.68e-04	2.15	2.91e-04	2.36	2.40e-04	2.44
3	4.32e-05	1.96	7.08e-05	2.04	4.88e-05	2.30
3^{rd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.17e-02	—	2.48e-02	—	1.97e-02	—
1	2.25e-03	2.38	4.93e-03	2.33	3.61e-03	2.44
2	3.65e-04	2.62	7.89e-04	2.64	5.72e-04	2.66
3	4.84e-05	2.91	1.02e-04	2.95	7.43e-05	2.94
4^{th}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	7.88e-03	—	1.41e-02	—	1.78e-02	—
1	4.84e-04	4.02	9.11e-04	3.95	6.96e-04	4.01
2	2.01e-05	4.59	3.91e-05	4.54	3.09e-05	4.49
3	7.49e-07	4.75	1.48e-06	4.73	1.25e-06	4.63

Table 4.10 L_∞ norm of error for all schemes on the annulus

2^{nd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.88e-02	—	6.24e-02	—	5.53e-02	—
1	6.23e-03	2.21	1.64e-02	1.93	1.26e-02	2.13
2	2.02e-03	1.62	3.35e-03	2.30	3.12e-03	2.02
3	9.93e-04	1.03	6.67e-04	2.33	7.68e-04	2.02
pseudo 3^{rd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	2.37e-02	—	4.33e-02	—	3.84e-02	—
1	7.22e-03	1.71	8.74e-03	2.31	8.79e-03	2.13
2	3.29e-03	1.13	1.76e-03	2.31	2.24e-03	1.98
3	2.03e-03	0.70	4.63e-04	1.93	5.52e-04	2.02
3^{rd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	5.83e-02	—	9.91e-02	—	7.83e-02	—
1	1.92e-02	1.60	2.37e-02	2.07	1.98e-02	1.98
2	4.86e-03	1.98	4.79e-03	2.30	3.43e-03	2.53
3	8.36e-04	2.54	7.62e-04	2.65	4.51e-04	2.93
4^{th}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	3.87e-02	—	5.52e-02	—	4.20e-02	—
1	3.01e-03	3.69	6.60e-03	3.06	4.62e-03	3.18
2	2.47e-04	3.61	3.33e-04	4.31	2.42e-04	4.26
3	3.01e-05	3.03	1.61e-05	4.37	1.16e-05	4.38

Table 4.11 L_1 norm of error for the third- and fourth-order schemes with a stunted flux integral

3^{rd}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	1.21e-02	—	2.47e-02	—	1.94e-02	—
1	2.48e-03	2.29	4.67e-03	2.40	3.91e-03	2.31
2	4.60e-04	2.43	8.10e-04	2.53	7.53e-04	2.38
3	8.45e-05	2.44	1.24e-04	2.70	1.39e-04	2.44
4^{th}	ρ		ρu		E	
Grid Index	Error	Order	Error	Order	Error	Order
0	7.89e-03	—	1.32e-02	—	1.14e-02	—
1	1.17e-03	2.76	1.44e-03	3.19	1.64e-03	2.79
2	2.92e-04	2.00	3.04e-04	2.25	2.57e-04	1.99
3	7.58e-05	1.94	7.54e-05	2.01	1.08e-04	1.93

CHAPTER 5

STEADY STATE FLOW SOLVER SIMULATION RESULTS

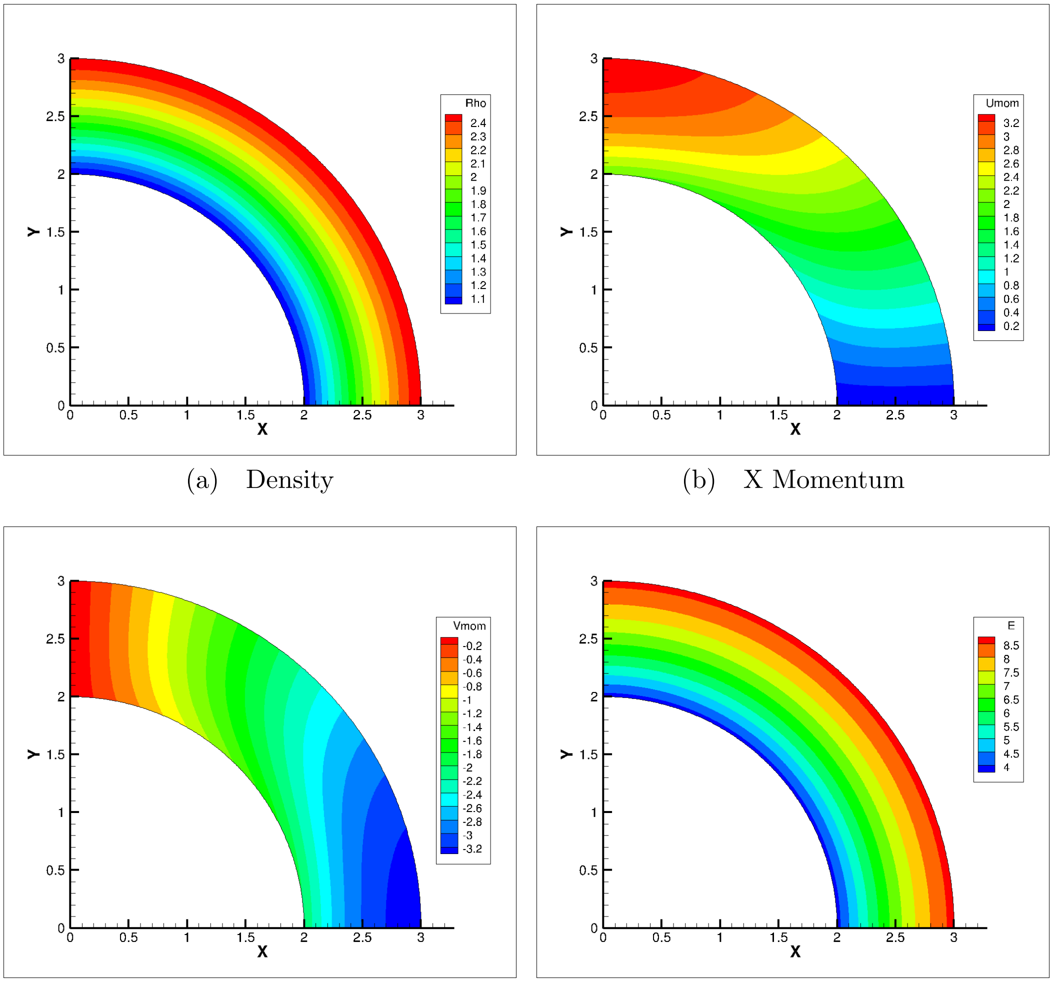
This chapter is devoted to the presentation of simulation results and the assessment of the high-order scheme in relation to the second-order scheme. The errors are quantified when a suitable analytical solution is available. Solution visualization of the variables is also used to illustrate the effectiveness of the high-order schemes.

Supersonic Flow in an Annular Geometry

The flow solver is used to study the solution of supersonic flow inside the annulus described in Chapter 4. Aftosmis [36] noted that this is a particularly useful test case for upwind methods since, for the solver to be correct, it must maintain a radial distribution of velocity and remain isentropic (no shocks are introduced). Following this, Ollivier-Gooch [25] gave the full analytical solution as

$$\begin{aligned}
\rho &= \rho_i \left(1 + \tfrac{\gamma-1}{2} M_i^2 \left(1 - \tfrac{R_i^2}{r^2}\right)\right)^{\frac{1}{\gamma-1}} \\
U_i &= M_i \rho_i^{\frac{\gamma-1}{2}} \\
U &= \tfrac{U_i R_i}{r} \\
u &= \tfrac{yU}{r} \\
v &= \tfrac{-xU}{r} \\
P &= \tfrac{\rho^\gamma}{\gamma}
\end{aligned} \tag{5.1}$$

where $R_i = 2$, $\rho_i = 1$, and $M_i = 2$. A subscript i denotes a value at the inner radius. The exact solution for this domain can be seen in Figure 5.1. Numerical solutions are computed on two sequences of grids. The first sequence uses aligned triangles as described in the MMS results from Chapter 4 on this domain. A second sequence of triangular elements from the unstructured grid algorithm was also used. This is to demonstrate the effectiveness of the high-order schemes on a general mesh and to allay doubt that the aligned triangles grids perform well simply because CV boundaries align nicely with the direction of the flow. The unstructured grid sequence is shown in Figure 5.2.



(a) Density

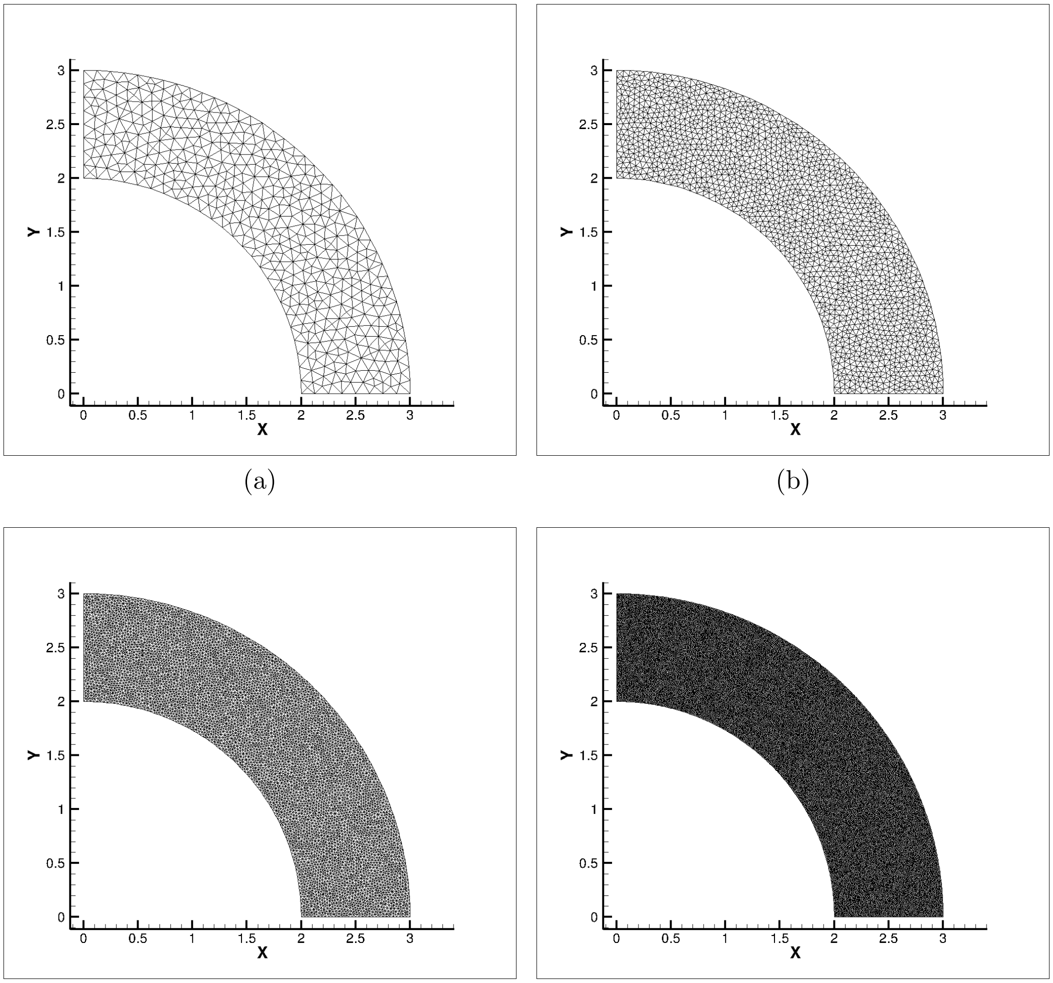
(b) X Momentum

(c) Y Momentum

(d) Total Energy

Figure 5.1 Contours of the analytical solution variables for the annulus in supersonic flow

The flow was initialized to be at rest and to have a constant density considerably less than a reference density of one following the suggestion in [36]. The density used in all



(a) (b)

(c) (d)

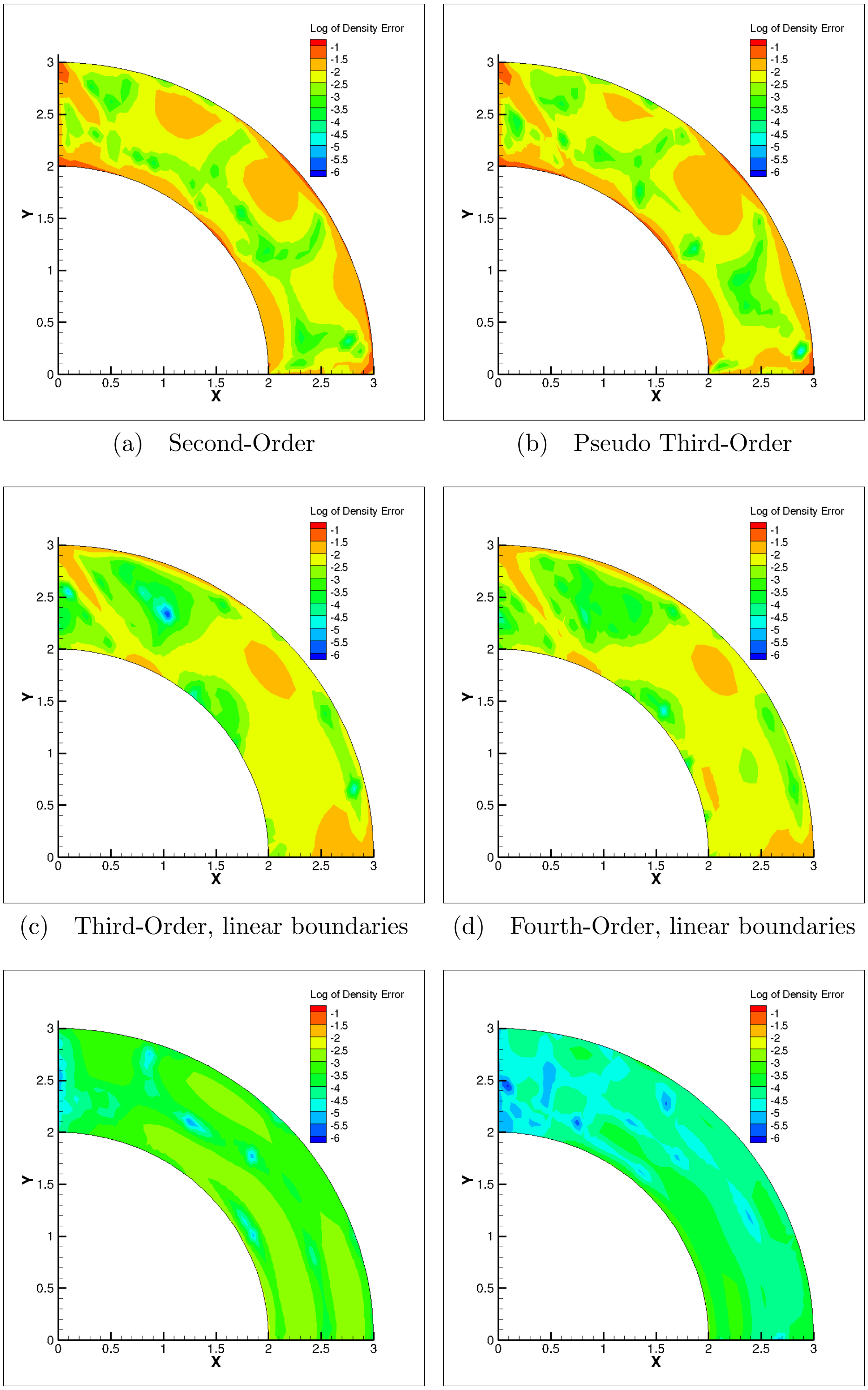
Figure 5.2 Unstructured grid sequence for the annulus geometry

proceeding tests was one-fifth. Numerical tests showed that this value worked for all grids and reconstruction schemes. This set of initial conditions created strong gradients in the transient solution, but the first-order iterations were successfully able to arrive at a converged steady solution. The radial boundaries were treated as solid walls, allowing no flow penetration. The inlet boundary, constant x, was imposed with the analytical solution distribution. Since the flow is supersonic, the outlet boundary was implemented as a simple flow extrapolation (ghost CVs are given their associated real CVs' variables). As described by the solution procedure overview in Chapter 2, the solver progressively increases the reconstruction order as the solution converges with respect to the current applied order. For the higher order reconstruction schemes, the geometric weighting parameter in equation 3.14 was set to 0, 1, and 2. Only results from the case when the parameter was 2 are given since it produced the best results. In the case of the second-order and pseudo third-order solution, the boundaries are treated as linear segments. The higher order methods were tested with both curved and linear boundaries to demonstrate the effect of imprecise boundary treatment.

The analytical values for the control volumes are found by integrating the equations found in 5.1 over the CV and dividing the solution by the area of the CV. These integrals take the curved boundaries into account, as described in Chapter 4. The base-10 logarithm of the difference in the density CV averaged values between the analytical solution and the numerical solution using all spatial accuracies on the coarsest mesh from the the aligned triangular element sequence can be seen in Figure 5.3 and from the coarsest mesh from the unstructured element sequence in Figure 5.4. The difference is shown using a logarithmic scale so that a consistent contour scale can be used across all plots, making visual inspection of the performance accessible. The contour plots indicate that the second- and pseudo third-order schemes produce similar error results on both grid types. It is also evident that the third- and fourth-order schemes using linear boundary treatment produce less error on the two coarsest grids than the second-order schemes. With the inclusion of proper treatment of the curved boundaries, the error is further diminished by the third- and fourth-order schemes, with the fourth-order scheme demonstrating the best performance. The contour plots further evince that proper boundary treatment is crucial not only for solution accuracy in the vicinity of the boundaries but for the entire computational domain.

The error norms for all variables on both grid sequences using all spatial accuracies are tabulated below. Here, the error norms are calculated with Equations 4.6 and 4.7. The observed order of accuracy between successive grids is again computed with Equation 4.9.

The observed order of accuracy is seen in Tables 5.1 and 5.3 to approach the theoretical order of two for both the second- and pseudo third-order schemes with regards to the L_1 norm on both grid sequences. Tables 5.2 and 5.4 show that the maximum errors converge nearly first-order. This is an expected result since the flux integrals along the boundary are in effect limited to first-order spatial accuracy, since there is no extrapolation or linear interpolation. This may also be in part explained by a finding in the report by Diskin and Thomas [37]. Linear treatment of the boundaries implies that nodes are the boundary are connected by a line segment from one point to the next, which the report designates as an $O(h^2)$ approximation to the curved boundary. However, in the evaluation of the flux integral along the boundary with CVs defined by the median dual, it is common to partition the boundary segment into two segments of equal length, each of which is associated with one of the endpoint nodes, to compute the integral. This, the researchers state, provides only an $O(h^1)$ approximation to the curved boundary.



(a) Second-Order

(b) Pseudo Third-Order

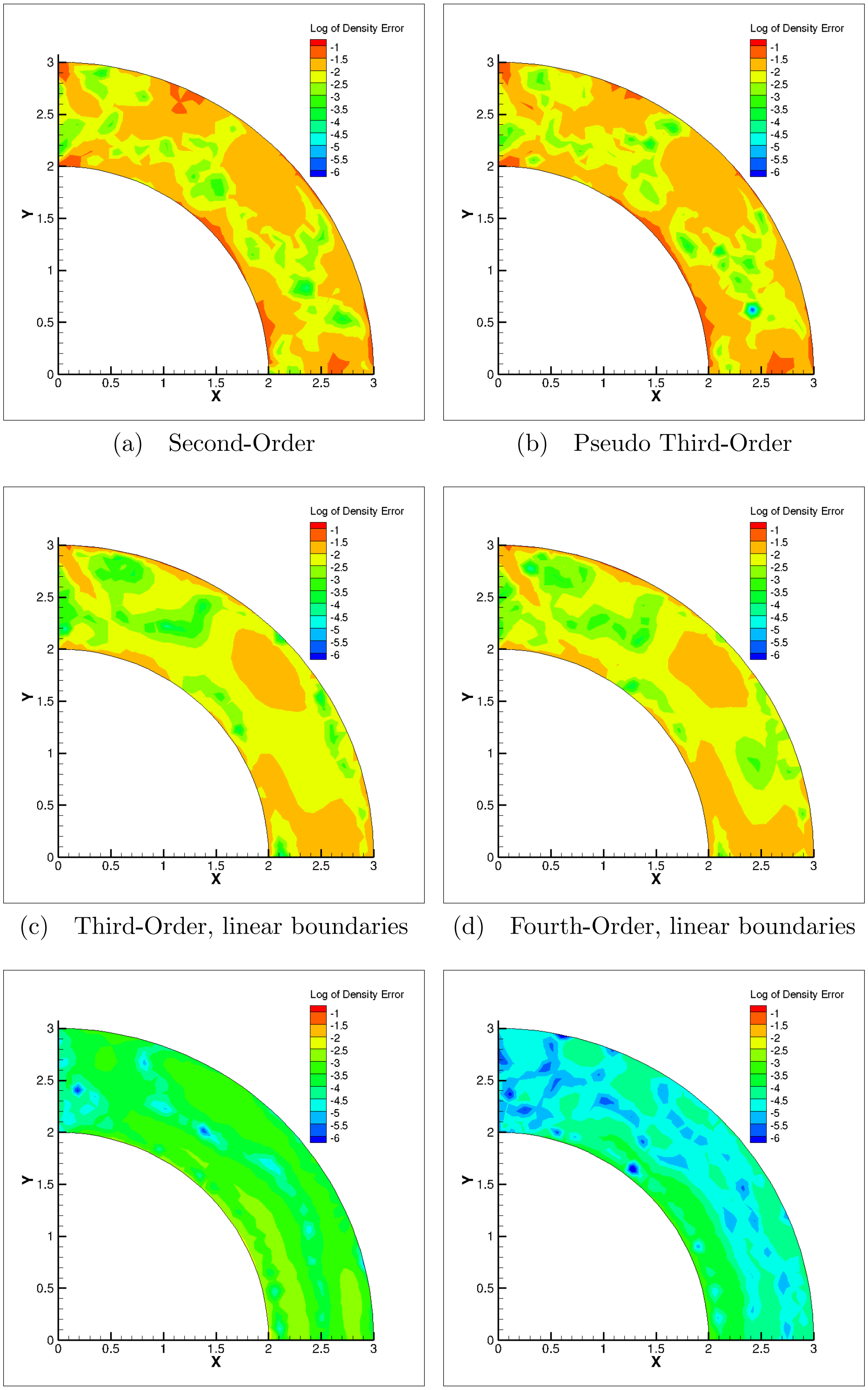
(c) Third-Order, linear boundaries

(d) Fourth-Order, linear boundaries

(e) Third-Order, curved boundaries

(f) Fourth-Order, curved boundaries

Figure 5.3 Log of the error in density on the coarsest aligned triangle grid



(a) Second-Order

(b) Pseudo Third-Order

(c) Third-Order, linear boundaries

(d) Fourth-Order, linear boundaries

(e) Third-Order, curved boundaries

(f) Fourth-Order, curved boundaries

Figure 5.4 Log of the error in density on the coarsest unstructured grid

Table 5.1 L_1 norm of error using the second-order scheme

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	8.00e-03	—	7.99e-03	—	1.02e-02	—	2.67e-02	—
1	2.30e-03	1.80	2.26e-03	1.82	3.01e-03	1.76	7.76e-03	1.78
2	7.45e-04	1.63	6.71e-04	1.75	8.96e-04	1.75	2.52e-03	1.62
3	2.21e-04	1.76	1.84e-04	1.87	2.48e-04	1.85	7.50e-04	1.75
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	1.26e-02	—	1.08e-02	—	1.22e-02	—	4.06e-02	—
1	3.70e-03	1.77	3.10e-03	1.81	3.70e-03	1.72	1.21e-02	1.74
2	1.16e-03	1.68	9.32e-04	1.73	1.15e-03	1.68	3.86e-03	1.65
3	3.46e-04	1.75	2.67e-04	1.81	3.39e-04	1.76	1.16e-03	1.73

Table 5.2 L_∞ norm of error using the second-order scheme

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.42e-02	—	9.51e-02	—	1.01e-01	—	2.24e-01	—
1	3.09e-02	1.06	4.60e-02	1.05	4.66e-02	1.11	1.08e-01	1.06
2	1.80e-02	0.77	2.27e-02	1.02	2.03e-02	1.20	5.88e-02	0.87
3	9.02e-03	1.00	1.13e-02	1.01	8.63e-03	1.24	2.92e-02	1.01
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	7.12e-02	—	1.06e-01	—	7.39e-02	—	2.48e-01	—
1	3.63e-02	0.97	5.45e-02	0.96	3.93e-02	0.91	1.27e-01	0.96
2	1.82e-02	1.00	2.74e-02	0.99	1.77e-02	1.15	6.40e-02	0.99
3	9.10e-03	1.00	1.37e-02	1.00	8.20e-03	1.11	3.20e-02	1.00

The L_∞ error norms from the third- and fourth-order schemes using straight boundaries shown in Tables 5.6 and 5.10 also demonstrate this degradation in accuracy.

Table 5.3 L_1 norm of error using the pseudo third-order scheme

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	8.15e-03	—	7.53e-03	—	1.02e-02	—	2.73e-02	—
1	2.34e-03	1.80	2.15e-03	1.81	3.00e-03	1.76	8.01e-03	1.77
2	7.47e-04	1.65	6.58e-04	1.71	9.21e-04	1.71	2.56e-03	1.64
3	2.15e-04	1.80	1.82e-04	1.85	2.53e-04	1.86	7.39e-04	1.80
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	1.25e-02	—	1.05e-02	—	1.33e-02	—	4.09e-02	—
1	3.70e-03	1.76	3.14e-03	1.75	4.06e-03	1.70	1.24e-02	1.73
2	1.14e-03	1.70	9.32e-04	1.75	1.21e-03	1.75	3.85e-03	1.68
3	3.34e-04	1.77	2.65e-04	1.82	3.46e-04	1.80	1.13e-03	1.76

Table 5.4 L_∞ norm of error using the pseudo third-order scheme

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.68e-02	—	6.55e-02	—	7.51e-02	—	2.73e-01	—
1	3.69e-02	0.86	3.15e-02	1.06	3.79e-02	0.97	1.24e-01	0.88
2	2.08e-02	0.82	1.58e-02	1.00	2.07e-02	0.88	6.96e-02	0.84
3	1.04e-02	1.01	7.83e-03	1.01	9.50e-03	1.12	3.43e-02	1.02
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.57e-02	—	7.59e-02	—	9.39e-02	—	2.02e-01	—
1	4.24e-02	0.63	3.87e-02	0.97	4.88e-02	0.95	1.44e-01	0.49
2	1.89e-02	1.16	1.94e-02	1.00	2.13e-02	1.20	6.41e-02	1.17
3	9.65e-03	0.97	9.72e-03	1.00	9.86e-03	1.11	3.23e-02	0.99

The reduction in observed order occurs here regardless of the fact that the flux integral is computed with Gaussian quadrature, even on the boundary. The L_1 error norms are seen to perform at slightly below an accuracy of two in Tables 5.5 and 5.9. This result is not altogether unexpected since the improper treatment of the boundaries corrupts the accuracy of the solver. Ollivier-Gooch [25] reported similar findings on this test case, though the observed order in the L_∞ norm approached two there. When curved boundaries are correctly accounted for, the third- and fourth-order schemes nearly recover their respective theoretical orders of accuracy as evidenced by Tables 5.7 and 5.11. With respect to the L_∞ norm, both schemes appear to lose an order of accuracy, as seen in Tables 5.8 and 5.12. It can be difficult to see full convergence in the L_∞ norm since it only records the single maximal error from all CVs. As was done in Chapter 4 with the MMS, the higher order schemes were also tested with the flux integral computed using a single quadrature node, but with correct boundary treatment, to demonstrate the necessity of a high-order flux integration scheme. Only the aligned triangle grid sequence is tested with this modification. The results are shown in Table 5.13 for the L_1 norm and in Table 5.14 for the L_∞ norm. The error norm results show that both high-order schemes become limited to near second-order performance, a result predicted by the MMS test. These results indicate that for a true high-order finite volume scheme a number of pieces must be all calculated accurately in tandem. This resonates acutely with the assertions made by Ollivier-Gooch and his fellow researchers (see [14, 28, 25] for instance) that for high-order schemes to recover their predicted orders of accuracy CV moments and the flux integral must be computed at commensurate orders of accuracy, as well as properly accounting for curvature in the boundaries, if present.

Table 5.5 L_1 norm of error using the third-order scheme with linear boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	5.38e-03	—	4.54e-03	—	7.27e-03	—	1.85e-02	—
1	1.45e-03	1.89	1.24e-03	1.88	2.10e-03	1.79	5.18e-03	1.84
2	4.56e-04	1.67	4.08e-04	1.60	7.01e-04	1.58	1.66e-03	1.64
3	1.27e-04	1.84	1.20e-04	1.76	2.03e-04	1.79	4.68e-04	1.82
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	7.19e-03	—	5.71e-03	—	9.51e-03	—	2.44e-02	—
1	2.17e-03	1.73	1.94e-03	1.56	3.22e-03	1.56	7.82e-03	1.64
2	6.31e-04	1.78	5.60e-04	1.79	9.27e-04	1.80	2.26e-03	1.79
3	1.83e-04	1.78	1.71e-04	1.71	2.75e-04	1.75	6.69e-04	1.76

Table 5.6 L_∞ norm of error using the third-order scheme with linear boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.06e-02	—	2.78e-02	—	7.94e-02	—	2.08e-01	—
1	3.37e-02	0.85	1.40e-02	0.99	3.80e-02	1.00	1.14e-01	0.86
2	1.98e-02	0.77	7.63e-03	0.88	2.26e-02	0.81	6.69e-02	0.77
3	1.00e-02	0.98	3.84e-03	0.99	1.13e-02	1.00	3.39e-02	0.98
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	5.56e-02	—	3.48e-02	—	8.99e-02	—	1.90e-01	—
1	3.81e-02	0.54	1.85e-02	0.91	5.26e-02	0.77	1.33e-01	0.53
2	1.80e-02	1.08	8.72e-03	1.09	2.39e-02	1.14	6.16e-02	1.11
3	9.14e-03	0.98	4.23e-03	1.04	1.20e-02	0.99	3.15e-02	0.97

Table 5.7 L_1 norm of error using the third-order scheme with curved boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	9.32e-04	—	1.53e-03	—	2.70e-03	—	4.19e-03	—
1	1.22e-04	2.93	1.92e-04	2.99	3.37e-04	3.01	5.32e-04	2.98
2	1.54e-05	2.99	2.30e-05	3.07	3.93e-05	3.10	6.67e-05	2.99
3	2.02e-06	2.93	2.91e-06	2.98	4.86e-06	3.02	8.59e-06	2.96
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	5.51e-04	—	8.07e-04	—	1.45e-03	—	2.38e-03	—
1	8.34e-05	2.72	1.16e-04	2.79	2.03e-04	2.84	3.57e-04	2.74
2	1.14e-05	2.87	1.56e-05	2.91	2.64e-05	2.94	4.85e-05	2.88
3	1.58e-06	2.86	2.08e-06	2.90	3.49e-06	2.92	6.64e-06	2.87

Table 5.8 L_∞ norm of error using the third-order scheme with curved boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	4.30e-03	—	4.95e-03	—	1.21e-02	—	1.73e-02	—
1	1.01e-03	2.09	1.16e-03	2.10	2.92e-03	2.05	4.13e-03	2.08
2	2.28e-04	2.14	2.60e-04	2.15	6.74e-04	2.12	9.42e-04	2.13
3	5.28e-05	2.11	5.95e-05	2.13	1.58e-04	2.09	2.19e-04	2.11
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	4.34e-03	—	5.58e-03	—	1.22e-02	—	1.80e-02	—
1	1.06e-03	2.03	1.38e-03	2.01	3.10e-03	1.98	4.55e-03	1.98
2	2.42e-04	2.14	3.03e-04	2.19	7.42e-04	2.06	1.05e-03	2.12
3	5.81e-05	2.06	7.06e-05	2.10	1.80e-04	2.04	2.53e-04	2.05

Table 5.9 L_1 norm of error using the fourth-order scheme with linear boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	5.42e-03	—	4.17e-03	—	6.83e-03	—	1.86e-02	—
1	1.45e-03	1.91	1.22e-03	1.78	2.01e-03	1.77	5.09e-03	1.87
2	4.54e-04	1.67	4.06e-04	1.58	6.62e-04	1.60	1.62e-03	1.65
3	1.27e-04	1.84	1.18e-04	1.79	1.89e-04	1.81	4.56e-04	1.82
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	7.52e-03	—	6.14e-03	—	9.68e-03	—	2.59e-02	—
1	2.27e-03	1.73	2.03e-03	1.60	3.18e-03	1.61	8.07e-03	1.68
2	6.68e-04	1.76	5.84e-04	1.80	9.10e-04	1.81	2.35e-03	1.78
3	1.91e-04	1.81	1.73e-04	1.75	2.65e-04	1.78	6.83e-04	1.78

Table 5.10 L_∞ norm of error using the fourth-order scheme with linear boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.31e-02	—	2.85e-02	—	8.70e-02	—	2.19e-01	—
1	3.58e-02	0.82	1.43e-02	1.00	4.57e-02	0.93	1.23e-01	0.83
2	2.03e-02	0.82	7.75e-03	0.88	2.62e-02	0.80	6.98e-02	0.82
3	1.03e-02	0.96	3.90e-03	0.99	1.33e-02	0.99	3.56e-02	0.97
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	5.66e-02	—	4.02e-02	—	1.01e-01	—	2.03e-01	—
1	4.17e-02	0.44	2.30e-02	0.81	6.27e-02	0.69	1.48e-01	0.45
2	1.88e-02	1.15	1.08e-02	1.09	2.78e-02	1.17	6.56e-02	1.17
3	9.61e-03	0.97	5.41e-03	1.00	1.45e-02	0.94	3.39e-02	0.95

Table 5.11 L_1 norm of error using the fourth-order scheme with curved boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	1.09e-04	—	1.63e-04	—	3.03e-04	—	4.59e-04	—
1	7.36e-06	3.89	1.04e-05	3.97	1.92e-05	3.98	2.95e-05	3.96
2	4.59e-07	4.00	6.65e-07	3.96	1.15e-06	4.06	1.85e-06	3.99
3	3.00e-08	3.93	4.26e-08	3.97	7.15e-08	4.01	1.18e-07	3.97
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.71e-05	—	9.47e-05	—	1.77e-04	—	2.87e-04	—
1	4.78e-06	3.81	6.63e-06	3.84	1.21e-05	3.87	1.99e-05	3.85
2	3.32e-07	3.85	4.70e-07	3.82	7.83e-07	3.95	1.37e-06	3.87
3	2.20e-08	3.92	3.09e-08	3.92	5.30e-08	3.89	9.13e-08	3.90

Table 5.12 L_∞ norm of error using the fourth-order scheme with curved boundaries

Tri.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	1.30e-03	—	1.47e-03	—	4.08e-03	—	5.38e-03	—
1	1.48e-04	3.14	1.63e-04	3.17	4.61e-04	3.15	6.15e-04	3.13
2	1.66e-05	3.15	1.82e-05	3.16	5.17e-05	3.16	6.97e-05	3.14
3	1.91e-06	3.12	2.11e-06	3.11	5.95e-06	3.12	8.04e-06	3.12
Uns.	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	9.43e-04	—	1.20e-03	—	2.94e-03	—	4.08e-03	—
1	1.14e-04	3.05	1.50e-04	3.01	3.60e-04	3.03	5.10e-04	3.00
2	1.46e-05	2.96	1.73e-05	3.11	4.65e-05	2.95	6.55e-05	2.96
3	1.79e-06	3.03	2.18e-06	2.99	5.83e-06	3.00	8.13e-06	3.01

Table 5.13 L_1 norm of error for the third- and fourth-order schemes with a stunted flux integral on the aligned triangle sequence

3^{rd}	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	8.82e-03	—	9.39e-03	—	1.33e-02	—	3.06e-02	—
1	2.55e-03	1.79	2.61e-03	1.85	3.35e-03	2.00	8.60e-03	1.83
2	7.64e-04	1.74	7.91e-04	1.72	9.90e-04	1.76	2.63e-03	1.71
3	2.23e-04	1.78	2.33e-04	1.76	2.90e-04	1.77	7.80e-04	1.76
4^{th}	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	9.71e-03	—	8.93e-03	—	1.18e-02	—	3.17e-02	—
1	2.96e-03	1.71	2.80e-03	1.67	3.96e-03	1.58	1.02e-02	1.63
2	8.69e-04	1.77	8.41e-04	1.73	1.16e-03	1.77	3.06e-03	1.74
3	2.45e-04	1.83	2.40e-04	1.81	3.24e-04	1.84	8.72e-04	1.81

Table 5.14 L_∞ norm of error for the third- and fourth-order schemes with a stunted flux integral on the aligned triangle sequence

3^{rd}	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.32e-02	—	9.66e-02	—	9.05e-02	—	2.22e-01	—
1	3.01e-02	1.07	4.65e-02	1.05	4.92e-02	0.88	1.06e-01	1.07
2	1.48e-02	1.02	2.32e-02	1.01	2.53e-02	0.96	5.24e-02	1.02
3	7.32e-03	1.02	1.15e-02	1.01	1.32e-02	0.94	2.59e-02	1.02
4^{th}	ρ		ρu		ρv		E	
Grid Index	Error	Order	Error	Order	Error	Order	Error	Order
0	6.39e-02	—	9.80e-02	—	1.07e-01	—	2.25e-01	—
1	3.04e-02	1.08	4.71e-02	1.06	5.48e-02	0.97	1.07e-01	1.07
2	1.49e-02	1.02	2.34e-02	1.01	2.92e-02	0.91	5.28e-02	1.02
3	7.76e-03	0.94	1.16e-02	1.01	1.42e-02	1.04	2.61e-02	1.02

The tabulated data can be more succinctly summarized by visual representation in Figure 5.5. Here, only the L_1 error norms for density are shown; the behavior of the other variables is similar. The plot shows that the second-order schemes (formal second-order, pseudo third-order, and imprecise high-order schemes) all cluster around the same data points. The third- and fourth-order schemes show a great improvement in performance, both in convergence and in initial errors. In fact, the fourth-order scheme's density error on the coarsest grid is lower than the error than any second-order scheme on the finest grid. This can also be seen by comparing the L_1 error norms on the index three grid for the second-order scheme in Table 5.1 with the error norms of the fourth-order scheme in Table 5.11 on the index zero grid. Notice that only Y momentum for the second-order scheme has a smaller error. This evidence reinforces the idea that higher order methods are capable of generating a solution with comparable accuracy to a lower order method using a coarser grid.

The issue of efficiency can be addressed by examining timing results from the second-, third-, and fourth-order schemes. The idea is to set the target error as the error generated on the finest grid from the aligned triangle sequence using the second-order scheme and then identify for the other schemes which grid produces approximately the same error. As mentioned in the previous paragraph, the coarsest mesh can be used for the fourth-order scheme. For the third-order scheme, the grid with index 1 will be used since the errors in the L_1 norm are close to those of the second-order scheme on the index 3 grid. In all three schemes, the solver starts with first-order iterations using the given initial conditions from above. The CFL number for the first iteration is one and is allowed to linearly increase to 400 spread over 200 iterations. The second-order scheme ran 125 first-order iterations before initiating second-order iterations. A total of 340 iterations were required to fully converge the solution. The third-order scheme ran 75 first-order, 50 second-order, and 150 third-order iterations. For the fourth-order scheme, the solver ran 100 first-order, 50 second-order, 50

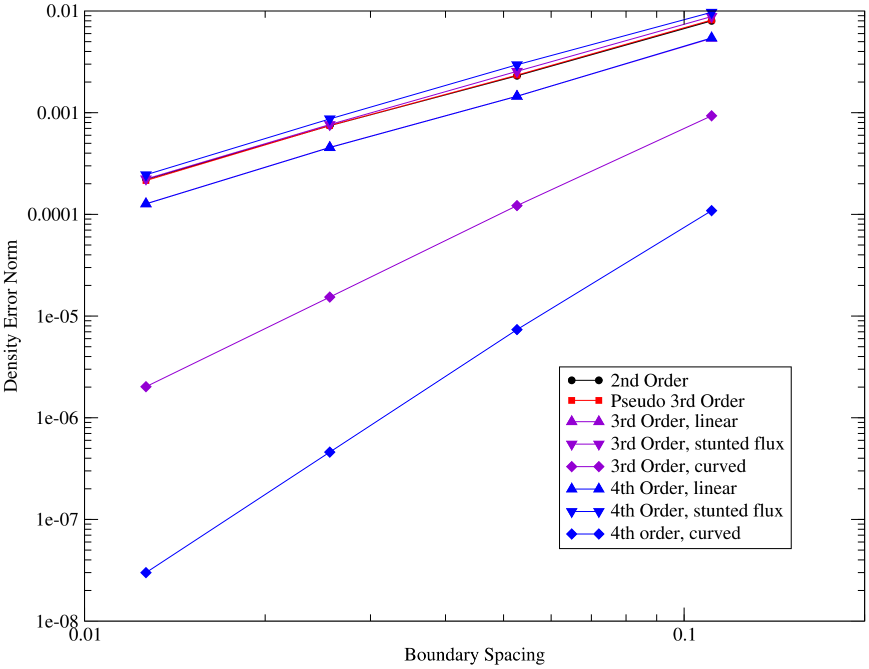


Figure 5.5 L_1 norm of density error for all schemes on the aligned triangle grid sequence for the annulus

third-order, and 100 fourth-order iterations. The number of iterations assigned to each order as the solver consecutively moves up to the desired reconstruction order was determined by finding a minimal number of iterations that preserved stability. The total number of applied iterations was found by examining at which point the convergence of the update vector stalled, and further iterations added no benefit.

The timing results for these cases are given in Table 5.15 averaged over 10 runs, as an effort to ensure that the reported time were not biased by any variability. An example of the residual vector history (in the context of the flow solver, this is the L_2 norm of the update vector, $\triangle\overline{Q}$, rather than the norm of the residual vector from Equation 2.6) for all three methods can be seen if Figure 5.6. The spikes in the residuals indicate where the solver has switched the order of reconstruction. The inclusion of additional derivatives in the reconstruction to quadrature nodes induces large changes in the magnitude of the solution update, $\triangle\overline{Q}$ from Equation 2.21. Note from the residual plot that it is not necessary to fully converge the solution on a given reconstruction order; rather, only enough iterations to dampen out the large changes in the transient state caused by the reconstruction order switching is required. As the residual plot indicates, the total number of iterations applied to each scheme is enough to converge the solution. Though not shown in Figure 5.6, these three cases were also run to 5000 total iterations to ensure the stability of the solution; all schemes remained stable in this circumstance. By comparing the total times, it is seen that the third-order scheme requires an order of magnitude less time and the fourth-order scheme requires two less orders of magnitude of time to compute an equivalent solution to the second-order scheme running on the finest grid.

The data in Table 5.15 also has a subtle detail that should be resolved. All schemes exhibit the property that time required per iteration averaged over the nodes for the first-order iterations is slightly greater than the second-order iterations which are doing more work. This result can be explained by the implementation of the linear system solver and the transient behavior of the solution during the start up. The linear solver, discussed in Chapter 2, applies a specified number of iterations each time step, stopping if the error drops below a given tolerance. During the first-order solution, the solution updates are at their most drastic level resulting in the linear system not reaching the specified convergence. However, the solution updates get smaller during the second-order iterations since the flow field is more fully resolved, thereby requiring less than the specified number of iterations to converge and therefore terminating the linear solve sooner. The end result is that while more work is done during the flux evaluation for second-order, on average, less linear solver iterations are needed.

Table 5.15 Timing results for the second-, third-, and fourth-order schemes in seconds

2^{nd}-order scheme		Total time = **127.220 s**[*], N_{CV} = 25600	
Iteration Level	Time	Time/Iteration	Node averaged Time/Iteration
1^{st}-order	49.4213	0.395370	1.54441e-05
2^{nd}-order	77.7984	0.361853	1.41349e-05

3^{rd}-order scheme		Total time = **11.3264 s**[*], N_{CV} = 1700	
Iteration Level	Time	Time/Iteration	Node averaged Time/Iteration
1^{st}-order	2.58000	0.0344000	2.02353e-05
2^{nd}-order	1.70389	0.0340778	2.00457e-05
3^{rd}-order	7.04253	0.0469502	2.76178e-05

4^{th}-order scheme		Total time = **4.32762 s**[*], N_{CV} = 420	
Iteration Level	Time	Time/Iteration	Node averaged Time/Iteration
1^{st}-order	1.13666	0.0113666	2.70633e-05
2^{nd}-order	0.564944	0.011299	2.69021e-05
3^{rd}-order	0.785478	0.015710	3.74037e-05
4^{th}-order	1.84054	0.0184054	4.38223e-05

[*] Executed on an Intel® Core™ i5 750.

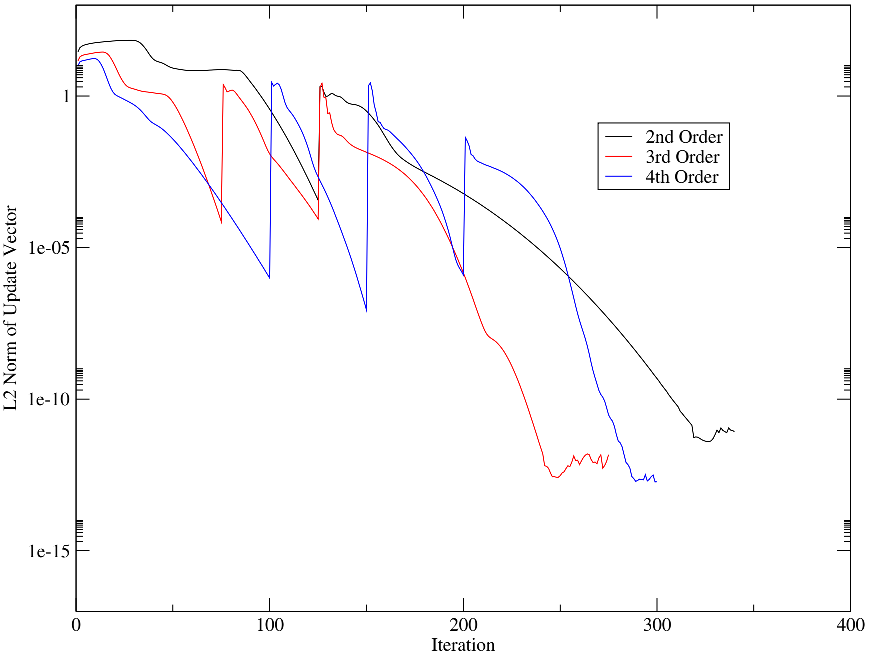


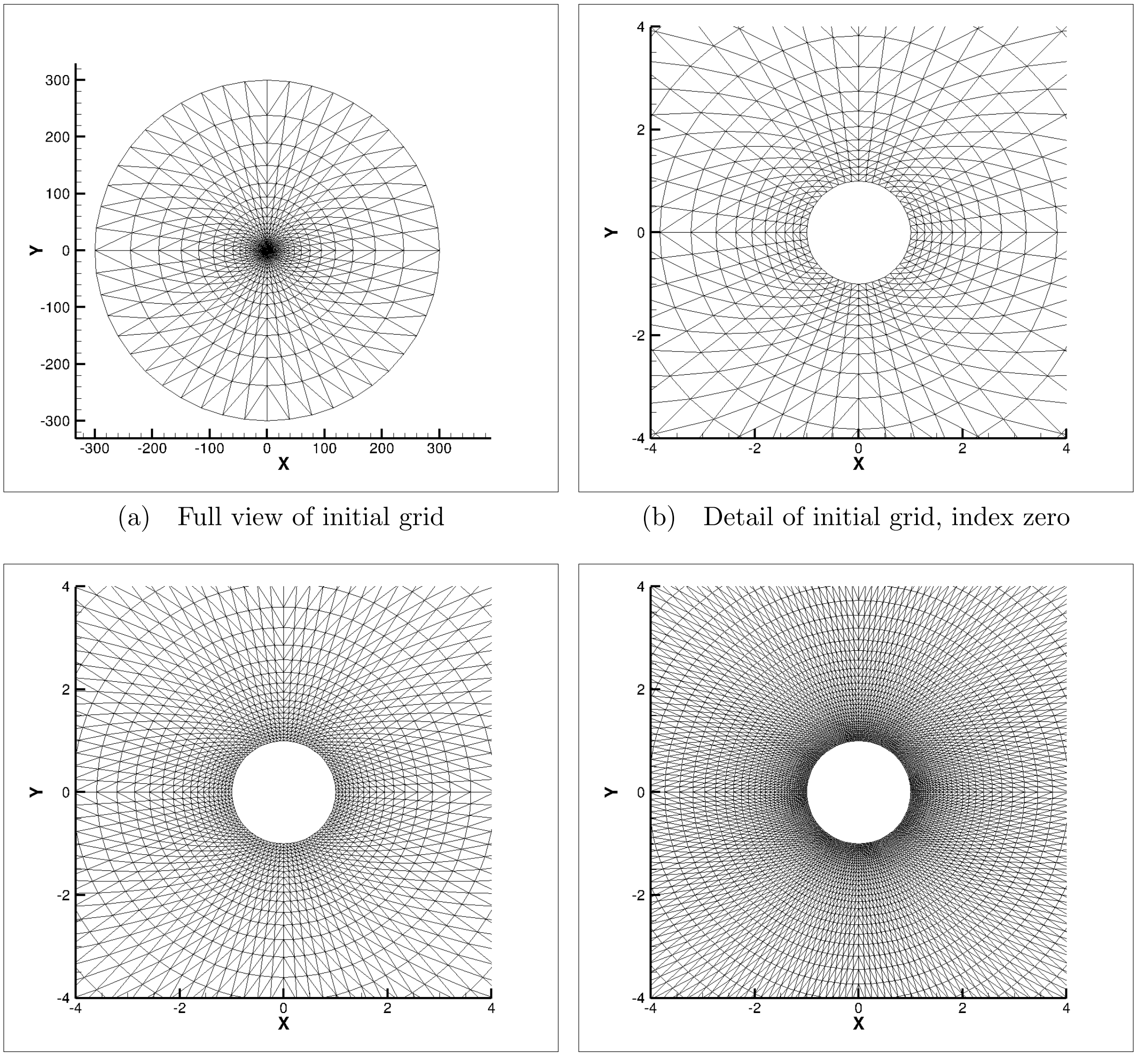
Figure 5.6 L_2 norm of the update vector for the second-, third-, and fourth-order schemes

Subsonic Flow Over a Circular Cylinder

The inviscid flow over a circular cylinder is computed in two dimensions with the unit circle centered at the origin to be the cross-section. The freestream velocity is set to $U_\infty = M_\infty = 0.3$ and is directed along the positive x axis. The farfield boundary is described by a circle centered at the origin with a radius of 300 and, relative to the inviscid boundary, is coarsely discretized to dissipate perturbations from a uniform flow field. Three separate grids were generated using GridGen$^{®}$ with successive refinement along the boundaries, which decreased spacing in the angular direction. The radial direction was also refined so that there was a smooth transition in element aspect ratio from the inner circle to the outer circle. This technique for constructing the grids did not result in uniform refinement making it difficult to assess error using norms. The domain was discretized using aligned triangular elements. Table 5.16 summarizes the information of the three grids. The grids around the inviscid boundary are shown in detail in Figure 5.7 along with a single view of the entire grid on the coarsest grid. Care was taken during the grid generation process to ensure that the elements were symmetric with respect to the x and y axes to reduce grid effects on the solution. The third-order scheme utilized a geometric weighting parameter of two, while the fourth-order scheme used a parameter of one since two was found to be unstable for this problem with the same solver input conditions as the other reconstruction schemes.

Table 5.16 Details of the grids used for the circular cylinder geometry

Grid Index	Boundary Points	Total Points	Number of Triangles
0	48	1488	2880
1	100	4600	9000
2	200	14200	28000



(a) Full view of initial grid

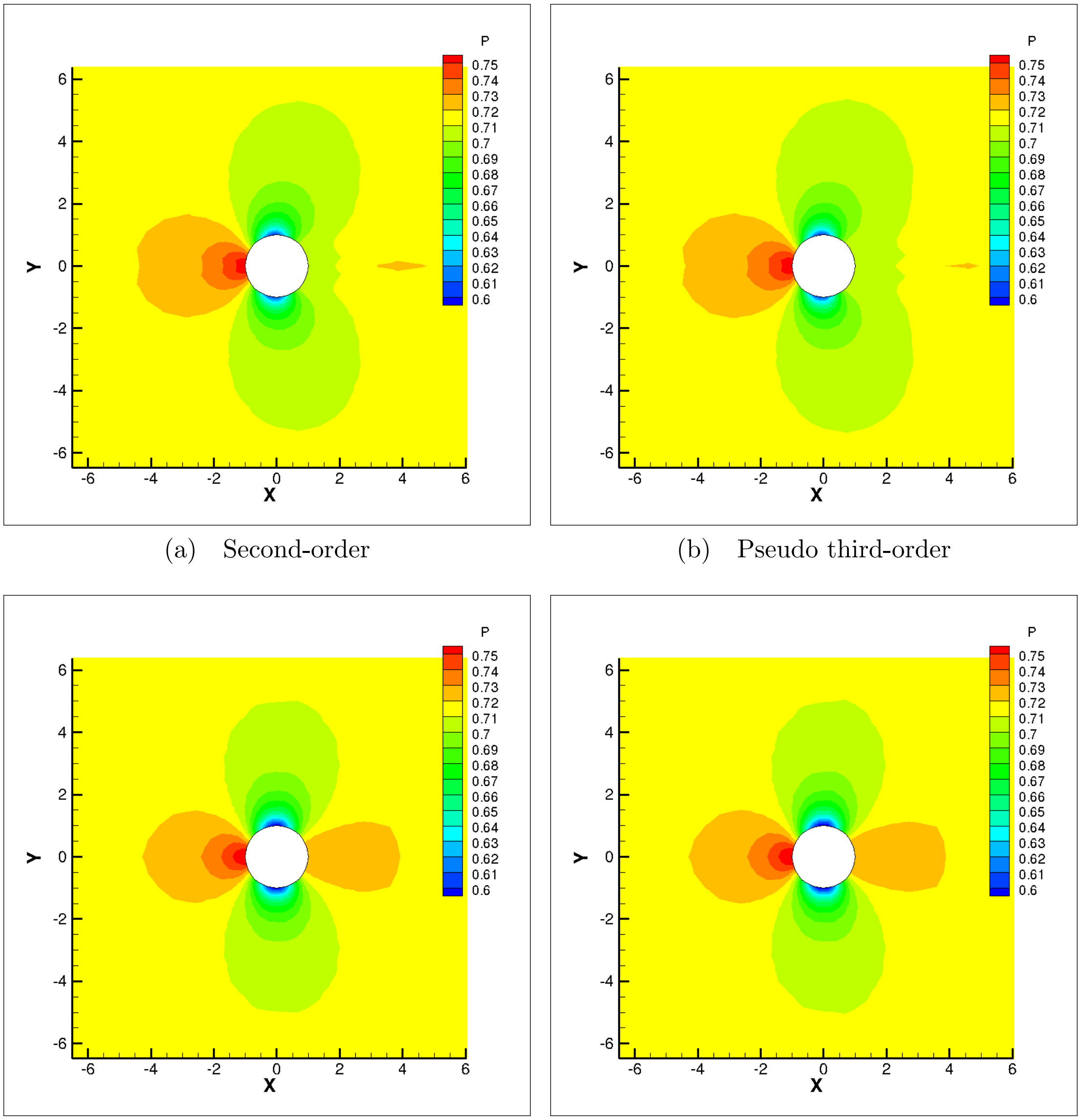
(b) Detail of initial grid, index zero

(c) Detail of first refined grid, index one

(d) Detail of second refined grid, index two

Figure 5.7 Aligned triangle grid sequence for the cylinder geometry

The numerical solutions can be assessed qualitatively by examining the pressure contours. The exact solution for this problem from potential theory, which neglects the effects of compressibility, asserts that there should regions of high pressure at the front and back of the cylinder and regions of low pressure along the top and bottom of the cylinder. Both regions are symmetric with the x and y axes, in turn. The expectation is that the higher order reconstruction schemes will provide better resolution of the rear high pressure region than the second-order schemes. The results from the coarsest grid are shown in Figure 5.8. The pressure contours illustrate the performance advantage the third- and fourth-order schemes have over the second-order schemes in that they show improved resolution of the front pressure region and can also be seen to begin to capture the rear high pressure regions. On the medium resolution grid (index one), the second-order schemes



(a) Second-order

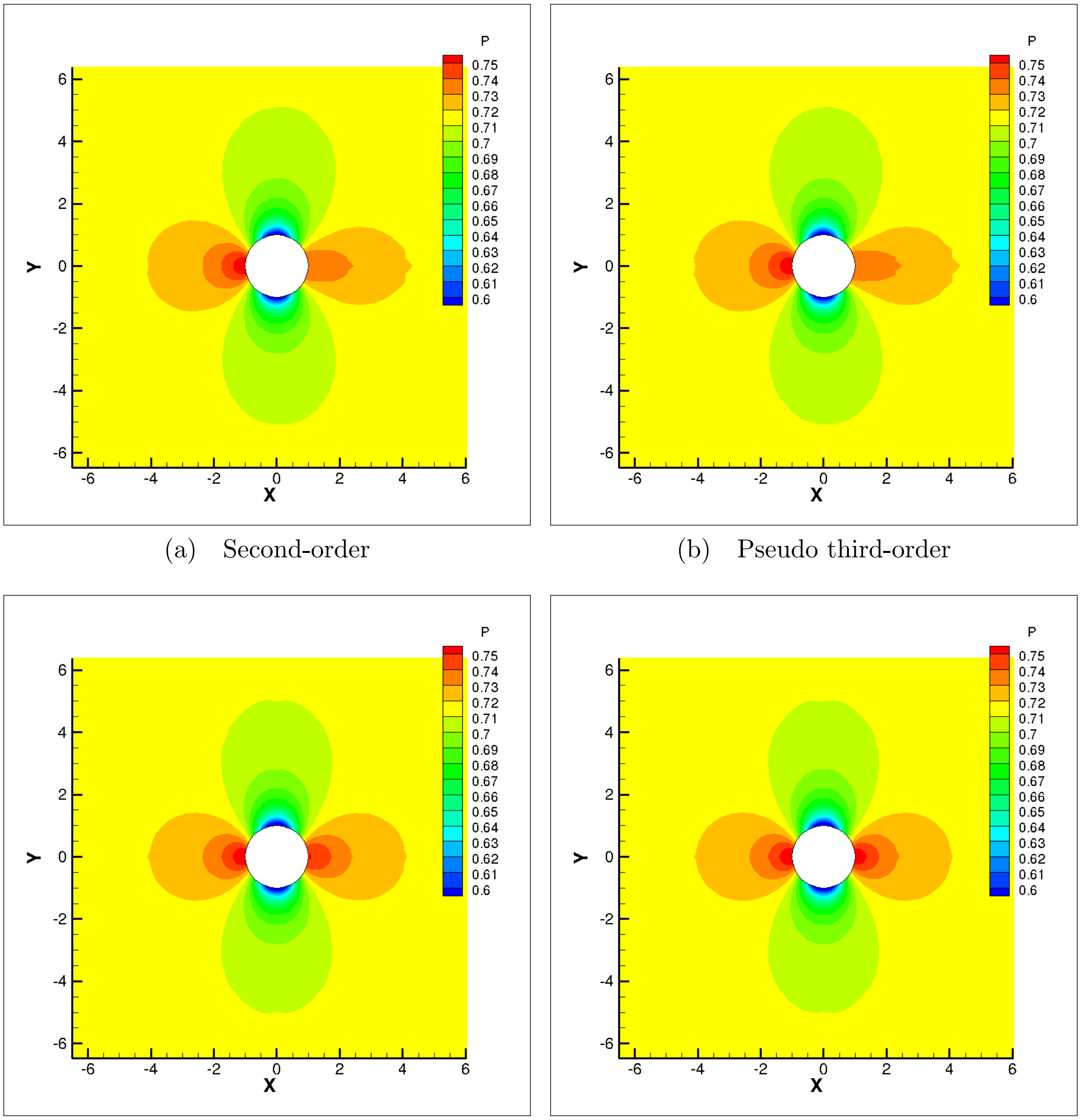
(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

Figure 5.8 Pressure contours on the index 0 grid

are beginning to resolve the rear high pressure region, as shown in Figure 5.9. The figure further demonstrates that the higher order schemes begin to fully resolve the pressure. The



(a) Second-order

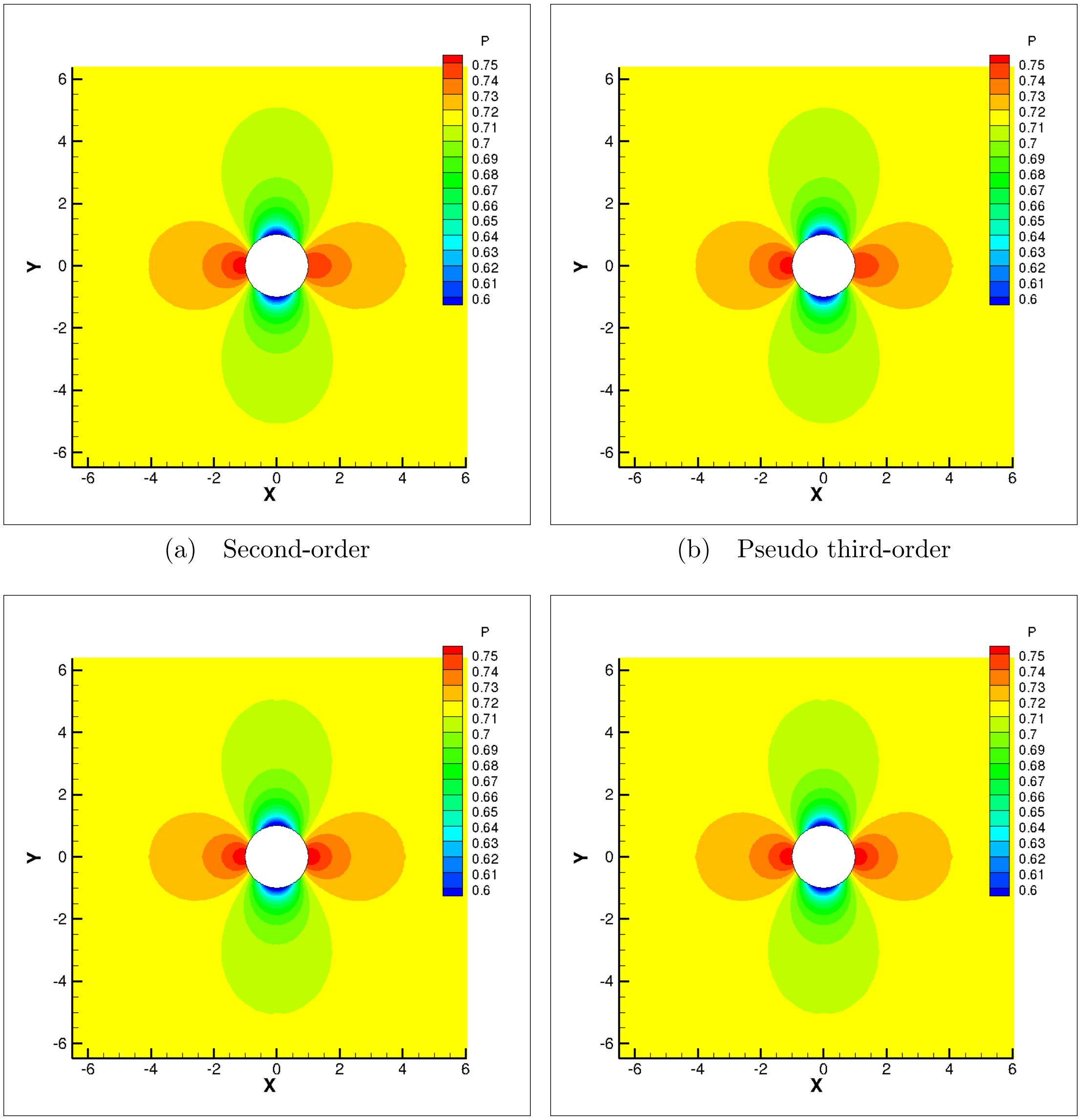
(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

Figure 5.9 Pressure contours on the index 1 grid

pressure contours from the finest grid, shown in Figure 5.10, suggest the resolution is nearly fine enough for all reconstruction orders to effectively resolve the solution. The higher order schemes still posses a slightly more resolved solution of the rear high pressure region. This concomitant resolution can be visually elucidated by closer examination of the back region of the cylinder in Figure 5.11. The figure shows that the third- and fourth-order schemes are able to recover a higher pressure at the rear stagnation zone, as indicated by the smallest red contour.



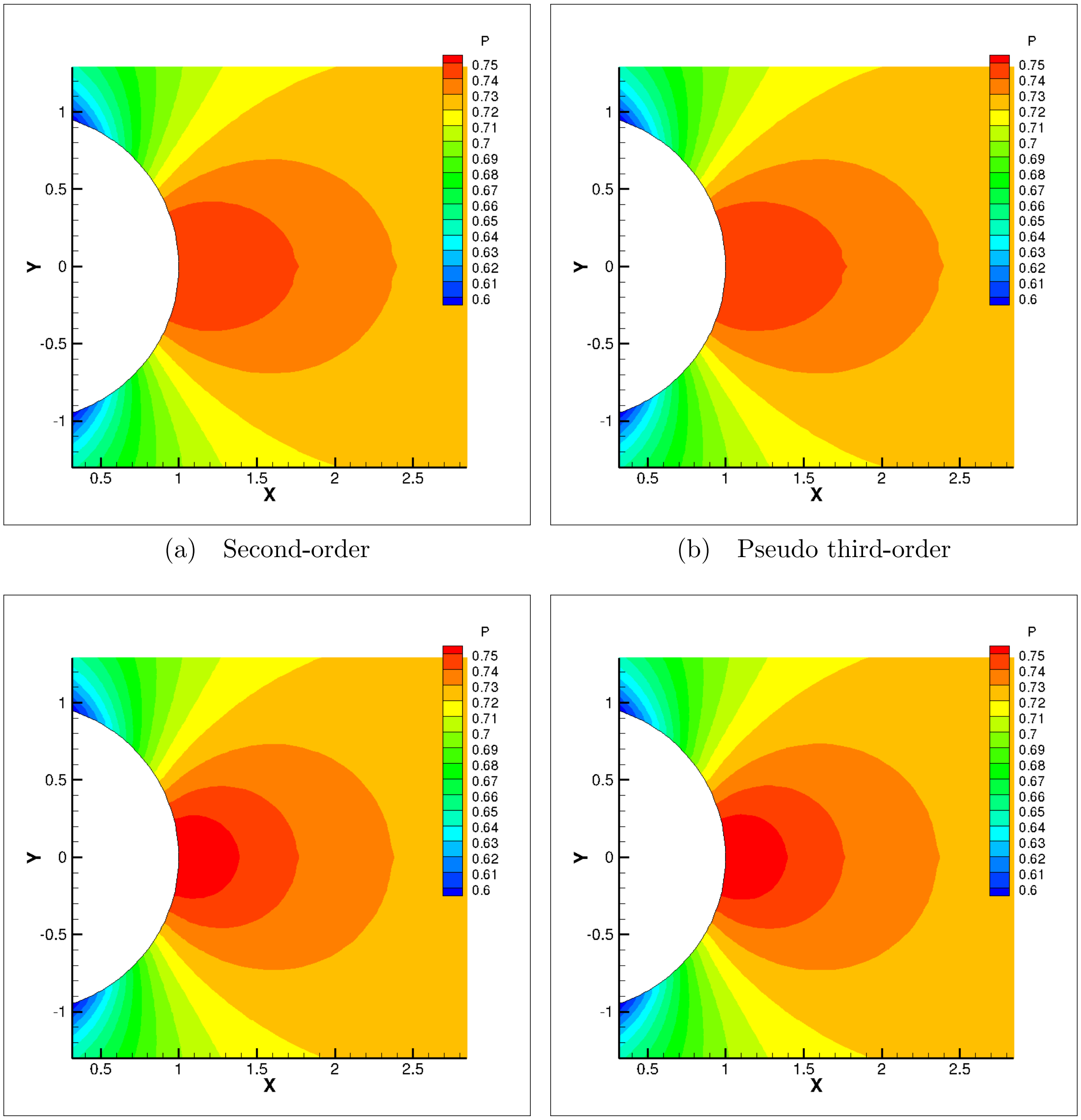
(a) Second-order

(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

Figure 5.10 Pressure contours on the index 2 grid



(a) Second-order

(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

Figure 5.11 Detailed look at pressure contours on the index 2 grid

The performance of the different reconstruction schemes can be inspected by the schemes' distribution of C_P along the boundary of the cylinder. As noted above, the exact solution from potential theory can be used to aid the assessment. The analytical C_P distribution is given as

$$C_{P,i} = 1 - 4\sin^2\theta \tag{5.2}$$

where the i in the subscript denotes that the result is for incompressible flow. The approximation used to account for compressibility effects in this study is the Prandtl-Glauert transformation,

$$C_P = \frac{C_{P,i}}{\sqrt{1 - M_\infty^2}} \tag{5.3}$$

The distribution of C_P along one half of the cylinder can be seen in Figure 5.12 for all spatial reconstructions and the Prandtl-Glauert curve. The plot shows that the third- and fourth-order schemes perform better than the second-order schemes on the coarsest grid. The high-order schemes are also shown to perform better in Figures 5.13 and 5.14 for the remaining two grids. In particular, the high-order methods follow the C_P distribution of the rear stagnation region further before diverging than the second-order schemes.

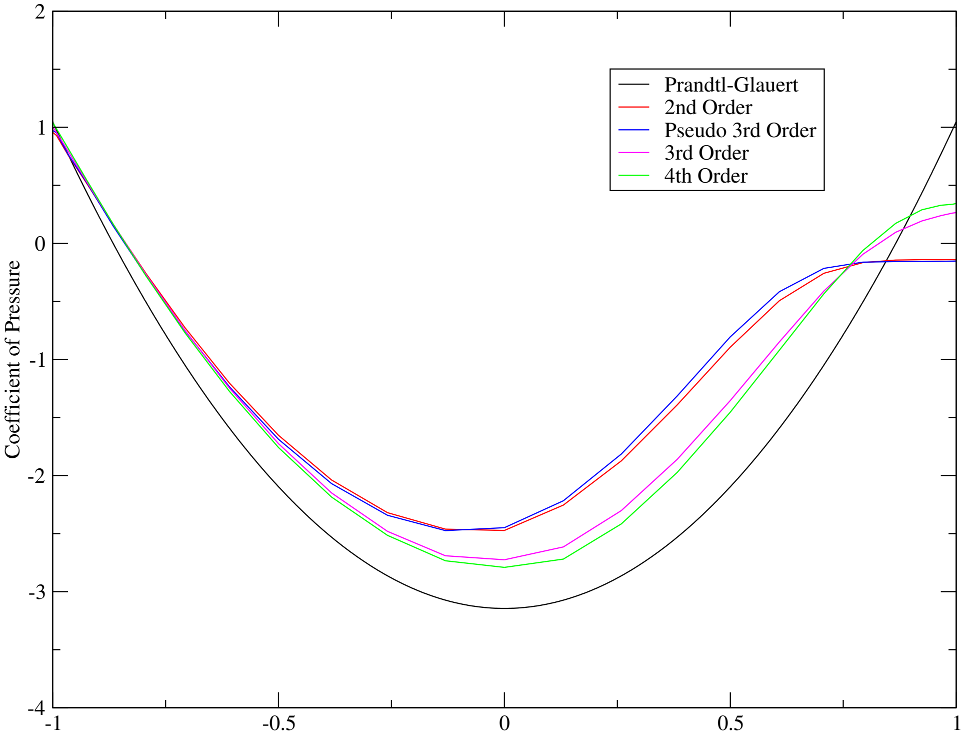


Figure 5.12 Coefficient of pressure for the index 0 grid

The numerical results demonstrate the increased performance of the high-order reconstruction schemes. Both the third- and fourth-order schemes are able to resolve the

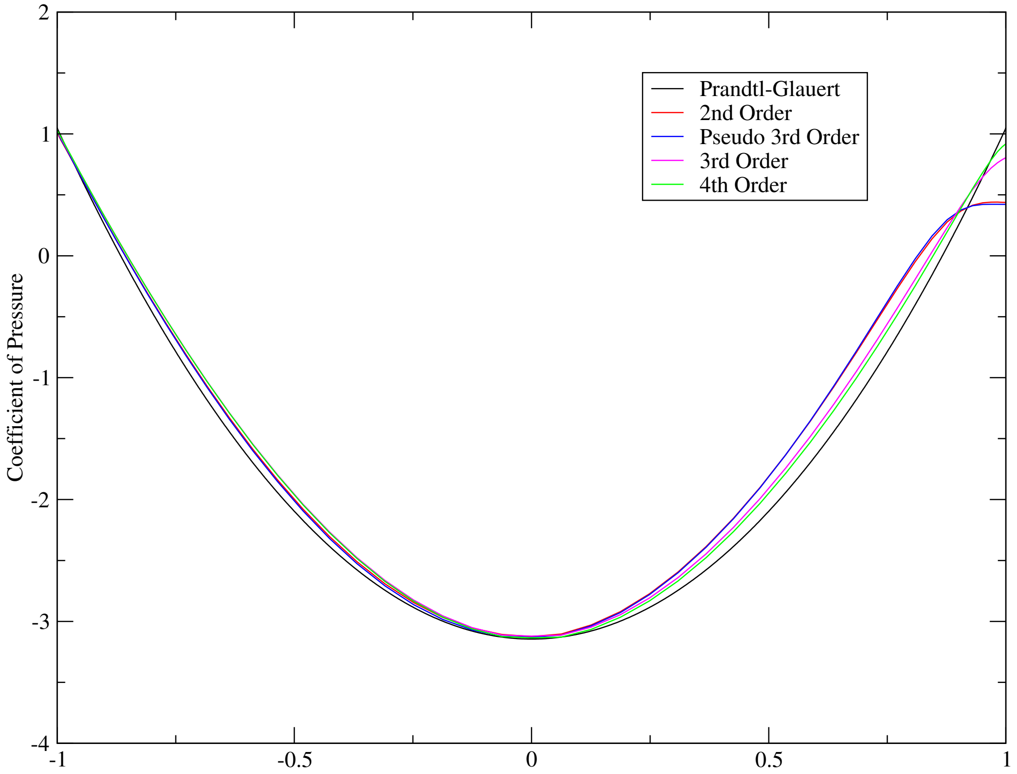


Figure 5.13 Coefficient of pressure for the index 1 grid

rear stagnation region with greater clarity than the lower order methods on the same grid. Since all schemes gave nearly identical results in the front-facing stagnation zone, shown by the pressure contours and C_P distribution plots, the question arose as to the possibility of a hybrid reconstruction scheme. Here, the meaning of a hybrid scheme is to perform low-order reconstruction in some regions of the flow and high-order reconstruction in others. This idea was implemented for the cylinder geometry by specifying second-order reconstruction for all CVs with coordinates such that $x < 0$. All other CVs are allowed to reconstruct with the fourth-order scheme. The results, shown as pressure contours, from the hybrid scheme on all the grids are given in Figure 5.15 along with the results from the full fourth-order scheme. The C_P distribution is also plotted for the Prandtl-Glauert curve, second-order, fourth-order, and hybrid schemes in Figure 5.16. A detailed plot of the low pressure region

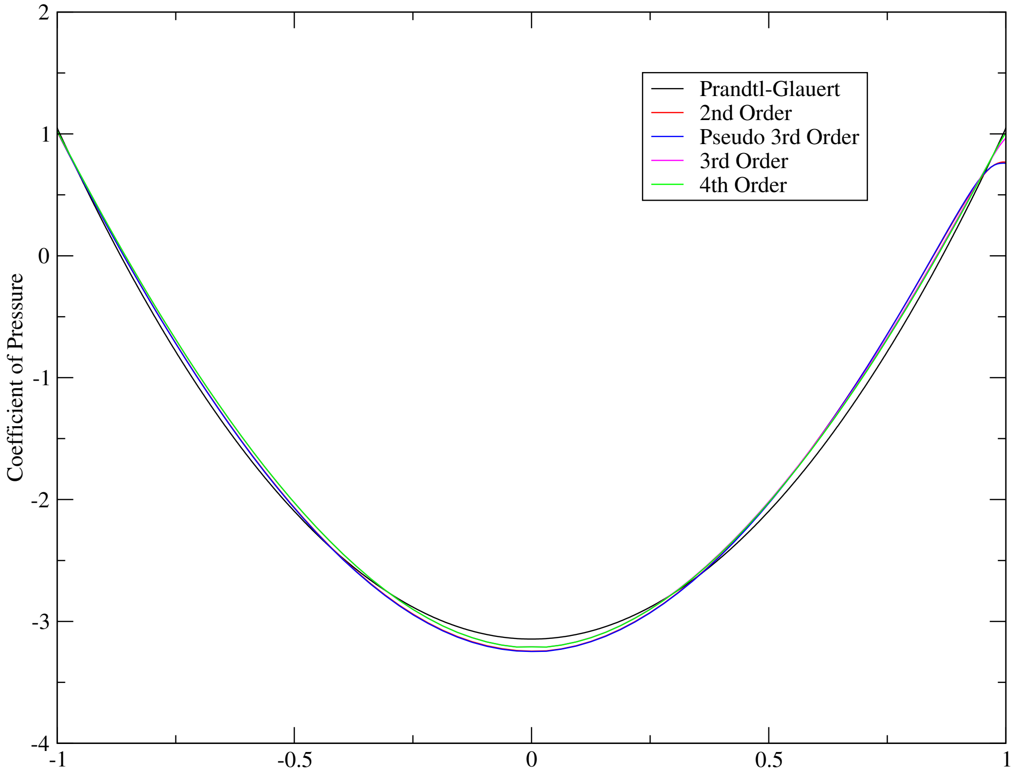


Figure 5.14 Coefficient of pressure for the index 2 grid

is shown in Figure 5.17. The detail plot reveals a glitch in the C_P as the reconstruction order on the boundary transitions from second- to fourth-order. The remaining portion of the C_P distribution in the high order region of the hybrid scheme solution follows the distribution of the fourth-order scheme. The deviation of the C_P values in the low order region from the distribution of the second-order scheme is caused by treating the entire boundary as curved rather than linear, as was done for the second-order scheme's solution. The results from the figures demonstrate that the hybrid scheme is able to produce results that are on par with the results from the full fourth-order scheme.

A final metric to assess the performance of the schemes is to look at the error they generate in total pressure as Nejat [28] did for flow about a half cylinder. The flow about the full cylinder is inviscid and isentropic so that the total pressure should be preserved everywhere in the field as gas dynamic theory predicts. Then any deviation from this condition in the numerical solutions can be seen as error produced by the respective scheme. The total pressure is given as

$$P_t = P\left(1 + \frac{1}{2}\left(\gamma - 1\right)M^2\right)^{\dfrac{\gamma}{\gamma - 1}}$$

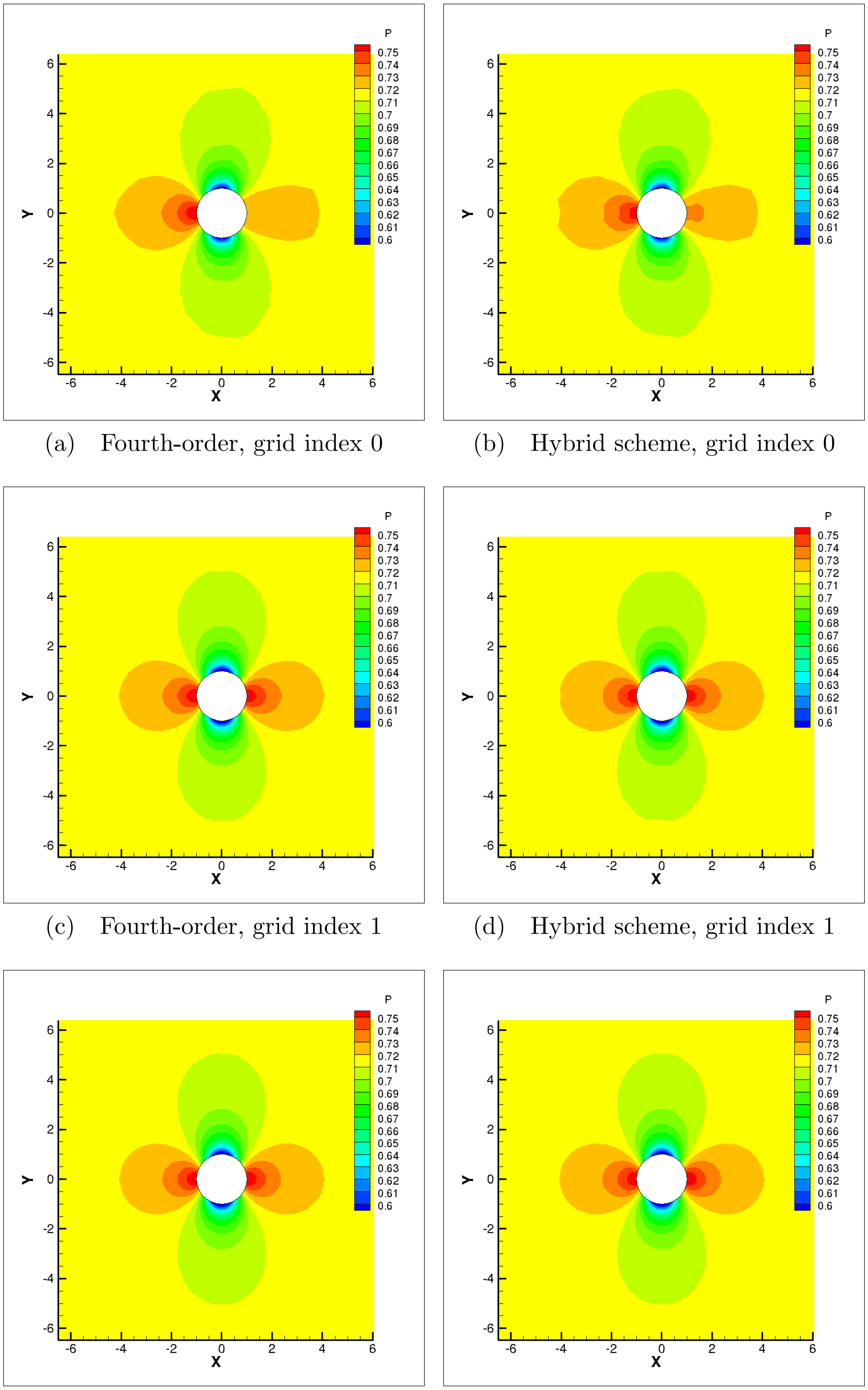
Equation 4.7 can be used to integrate the error over the control volume with

$$E_i = \frac{P_{t_i}}{P_{t_\infty}} - 1$$

replacing the integrand and E_i is used to denote the error in CV i. The calculation of the norm of the error can then proceed as usual using Equation 4.6. The results from all the numerical schemes applied to the three grids is tabulated in Table 5.17. Since the grids are not uniformly refined, the observed order of accuracy is not computed, instead the ratio of the successive errors is calculated. The ratios of error in the L_1 norm for the third- and fourth-order schemes show that the error virtually stalls between the two finest grids, although the L_∞ display better convergence. This might be a result of the non-uniform grid refinement. The hybrid reconstruction scheme error norm results further validates the performance of the scheme, even if formal accuracy can not be readily shown.

Table 5.17 Error norms for all schemes on the cylinder grids

2^{nd}				
Grid Index	Error, L_1	Ratio	Error, L_∞	Ratio
0	6.8556e-05	—	7.3320e-02	—
1	1.5694e-05	4.37	3.5645e-02	2.06
2	5.2857e-06	2.97	1.5229e-02	2.34
pseudo 3^{rd}				
Grid Index	Error, L_1	Ratio	Error, L_∞	Ratio
0	6.6912e-05	—	7.3997e-02	—
1	1.4983e-05	4.47	3.6620e-02	2.02
2	5.0613e-06	2.96	1.5811e-02	2.32
3^{rd}				
Grid Index	Error, L_1	Ratio	Error, L_∞	Ratio
0	2.7241e-05	—	4.9981e-02	—
1	2.7759e-06	9.81	1.3851e-02	3.61
2	2.6490e-06	1.05	3.5903e-03	3.86
4^{th}				
Grid Index	Error, L_1	Ratio	Error, L_∞	Ratio
0	2.1985e-05	—	4.3290e-02	—
1	2.1989e-06	10.00	6.5177e-03	6.62
2	2.1455e-06	1.02	8.4491e-04	7.74
Hybrid				
Grid Index	Error, L_1	Ratio	Error, L_∞	Ratio
0	2.5854e-05	—	4.1212e-02	—
1	3.5878e-06	7.21	6.4479e-03	6.39
2	2.2558e-06	1.59	6.6254e-04	9.73



(a) Fourth-order, grid index 0

(b) Hybrid scheme, grid index 0

(c) Fourth-order, grid index 1

(d) Hybrid scheme, grid index 1

(e) Fourth-order, grid index 2

(f) Hybrid scheme, grid index 2

Figure 5.15 Comparison of pressure contours between the fourth-order and hybrid schemes

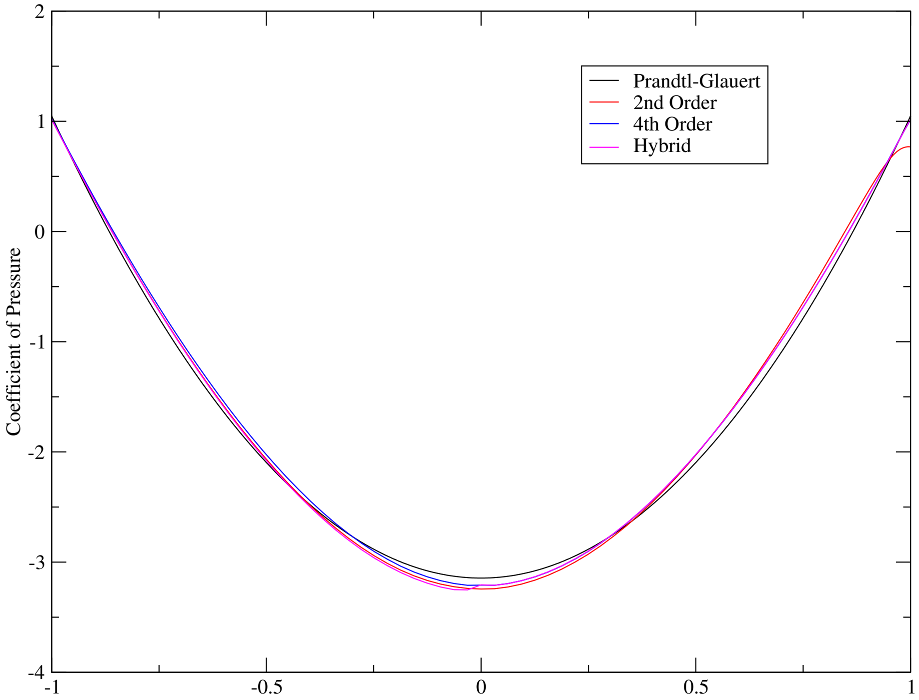


Figure 5.16 Coefficient of pressure for the index 2 grid with the hybrid scheme

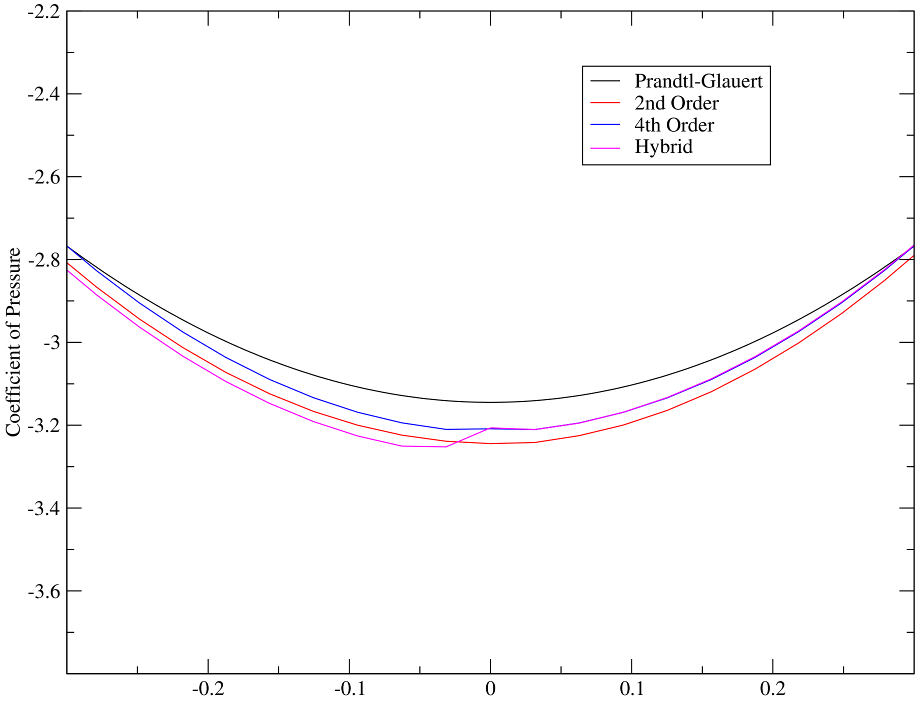


Figure 5.17 Detail of the coefficient of pressure for the index 2 grid with the hybrid scheme

NACA 0012 Airfoil

The final geometry studied in this research for steady state solutions is the NACA 0012 airfoil. Three unstructured grids are considered in the following test cases. The grid characteristics are given in Table 5.18. The farfield boundary is located 25 chord lengths away from the geometry. The leading and trailing edges of the airfoil are given additional refinement relative to the rest of the surface. The three grids are shown in detail around the airfoil in Figures 5.18, 5.19, and 5.20. As seen by the previous results, correct treatment of the curved boundaries is essential to the performance of the higher order methods. The previous examples of curved boundaries were defined by circular arcs that made the computation of the quadrature nodes and arc length facile. Wang and Sun [38] explored the possibility of approximating curved boundaries, including the NACA 0012 airfoil, using circular arcs that were computed locally for boundary points. This idea is particularly beneficial in that it can handle arbitrary geometries without knowledge of the exact representation. However, the approach undertaken in this research was to fully model the airfoil geometry.

Table 5.18 Grid characteristics for the NACA 0012 geometry

Grid Index	Number of nodes	Number of nodes on upper/lower surface
0	1325	125
1	3275	150
2	9188	200

The shape of the NACA 0012 airfoil is given by the equation

$$y(x) = \pm 0.6\,(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4)\,,\; x \in [0,1] \qquad (5.4)$$

To compute the quadrature nodes and the arc length, the equation is parameterized using the variable t. The arc length of a segment whose endpoints are given by x_a and x_b, with

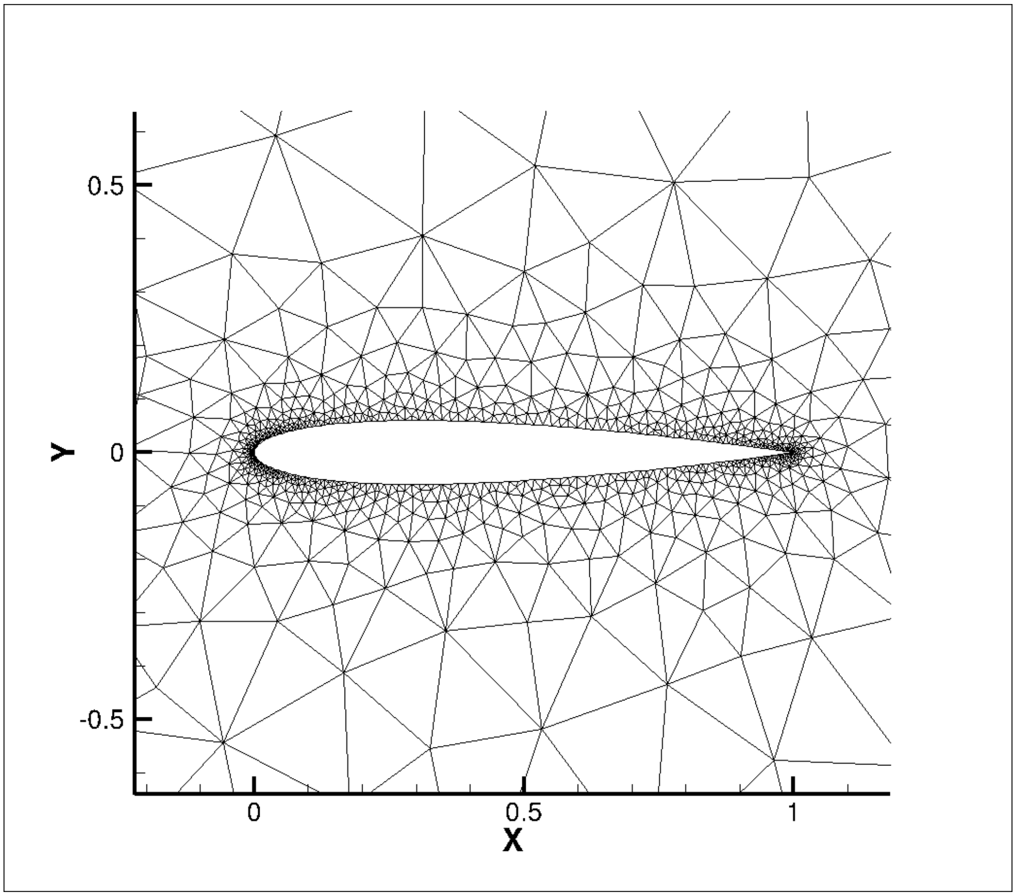


Figure 5.18 Index 0 grid for the NACA 0012 airfoil

$x_a < x_b$, is calculated as

$$L = \int_{x_a}^{x_b} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\, dt$$

Choosing the x to be the parameter, the arc length is computed as

$$L = \int_{x_a}^{x_b} \sqrt{1 + y'(x)^2}\, dt \tag{5.5}$$

The derivative of the airfoil equation is

$$y'(x) = \frac{0.08907}{\sqrt{x}} - 0.07560 - 0.42192x + 0.51174x^2 - 0.24864x^3 \tag{5.6}$$

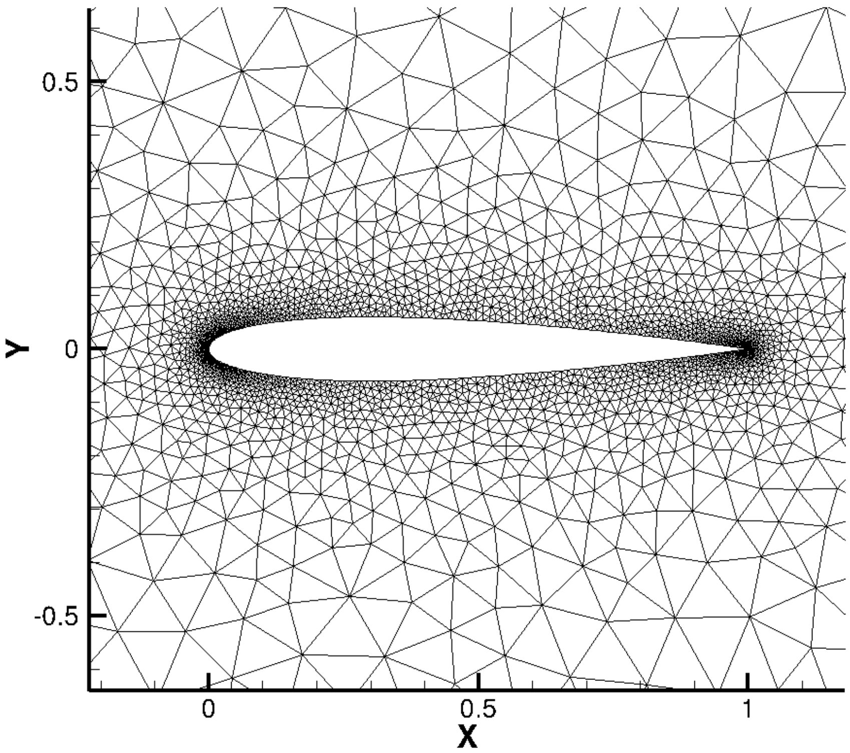


Figure 5.19 Index 1 grid for the NACA 0012 airfoil

Since the derivative in Equation 5.6 contains a singularity in the interval, Gaussian quadrature is no longer appropriate to integrate the arc length. To compute the integral, the GNU Scientific Library (GSL) is used. The GSL defines a function to integrate regions with singularities using an adaptive 21-point Gauss-Kronrod rule, which is used to compute the arc length of all segments describing the airfoil. The quadrature nodes are also found with the aid of this function.

Gaussian quadrature nodes are the roots of the appropriate Legendre polynomial in the interval $t \in [-1, 1]$, and for $k = 3$ are given as 0 and $\pm\sqrt{3/5}$. Rather than map the segment interval, $x \in [x_a, x_b]$, to the interval $[-1, 1]$, the approach taken in this research is to find $x_i \in [x_a, x_b]$ such that the arc length of Equation 5.5, with x_i replacing x_b in

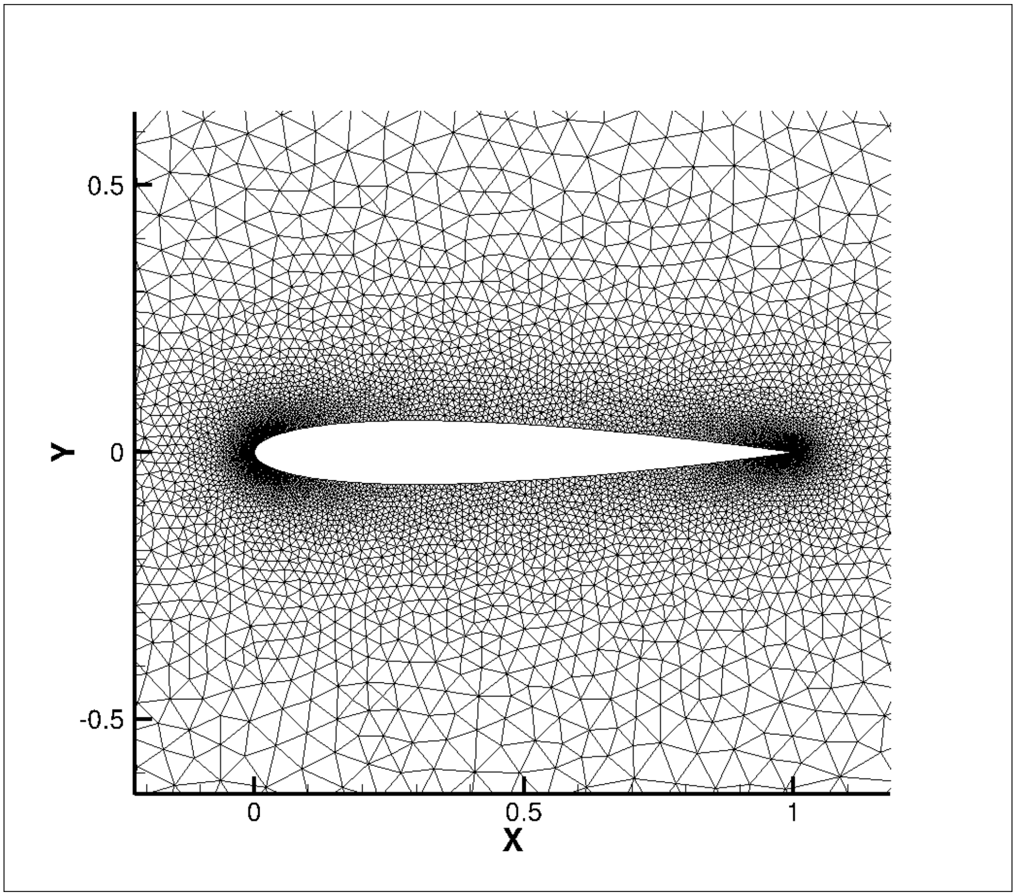


Figure 5.20 Index 2 grid for the NACA 0012 airfoil

the integration limit, has the correct fractional arc length. Consider the root $t = 0$ of the Legendre polynomial. This is the midpoint of the interval $[-1, 1]$. Thus, the corresponding quadrature node along the airfoil segment, $x_{t=0} \in [x_a, x_b]$, will satisfy the following condition:

$$\int_{x_a}^{x_i} \sqrt{1 + y'(x)^2}\, dt = \frac{1}{2}\int_{x_a}^{x_b} \sqrt{1 + y'(x)^2}\, dt$$

This condition is non-linear in the variable x, so must be solved with an appropriate non-linear solver, such as Newton's Method. The function Newton's Method will act upon is thus,

$$F(x) = \frac{1}{2}\int_{x_a}^{x_b} \sqrt{1 + y'(x)^2}\, dt - \int_{x_a}^{x_i} \sqrt{1 + y'(x)^2}$$

The other two quadrature nodes, $t = \pm\sqrt{3/5}$ are found similarly by replacing 1/2 with $(1 \pm \sqrt{3/5})/2$. The success of Newton's Method largely depends on the initial guess of the root. For the quadrature nodes, the initial guess used is the respective Legendre root linearly mapped onto the line segment from $(x_a, y(x_a))$ to $(x_b, y(x_b))$. This approach to calculating the quadrature nodes and arc lengths is more expensive than the way circular arcs are handled but is a one time expense since the results are computed and stored at the beginning of the solver run. The correctness of this method can be measured by how closed the control volumes are. From calculus, the integral of the normal vector around a closed boundary is zero. This procedure is computed for all nodes in the three grids from above. The largest non-closure in magnitude for all grids is 3.66205×10^{-15} which verifies that all CVs satisfy the expected result, within a very small tolerance.

Subsonic Solution with $M_\infty = 0.3$ and $\alpha = 0$

The first examined simulation on the NACA 0012 airfoil is a subsonic case with the freestream Mach number set at 0.3 and no angle of attack. This test case is useful for validating the effectiveness of the higher order methods on a grid with more complicated geometry. Furthermore, the flow is subsonic across the entire computational domain and is therefore isentropic. The performance of the different reconstruction methods can then be assessed with the condition that freestream entropy should be conserved. Any deviation from the value of freestream entropy is therefore expected to be the result of the reconstruction. The entropy itself is not calculated directly in the results from the flow solver; rather, the following condition from isentropic gas dynamics is used:

$$\frac{P}{\rho^\gamma} = Constant \tag{5.7}$$

The constant in the above equation is calculated using the freestream conditions for pressure and density. The error is then calculated for the different reconstruction schemes by integrating the difference between the computed value of Equation 5.7 and the freestream value over the control volumes.

The computed error norms from all reconstruction schemes are given in Table 5.19. The L_∞ is not given for these results since the maximal error in the grids was insensitive to refinement. The order of accuracy is not computed for these results since the grid refinement is not uniform. The geometric weight was set to zero for the third- and fourth-order schemes for these results which had the best convergence properties while still attaining a similar error on the finest grid. The results in the table show that all methods were able to reduce the error in the L_1 norm between the index 0 and 1 grids effectively. Though, all methods fail to reduce the error as much between the index 1 and 2 grids. The L_2 error norm results converge at a smaller ratio across all methods, being influenced by larger errors in the grids that are insensitive to grid refinement. The third- and fourth-order schemes both produce lower errors on the finest grid than the second-order schemes; nearly one half for the third-order scheme and one third for the fourth-order scheme in both error norms. The error in the CVs using the ratio of pressure and density from Equation 5.7 is displayed using the same color scale on the finest grids for all methods in Figure 5.21. The figure shows that in the lower order methods the largest errors occur along the boundary of the airfoil and continue beyond the trailing edge. For the higher order methods, the error about the airfoil has been minimized but persists in the region just behind the trailing edge.

Subsonic Solution with $M_\infty = 0.63$ and $\alpha = 2$

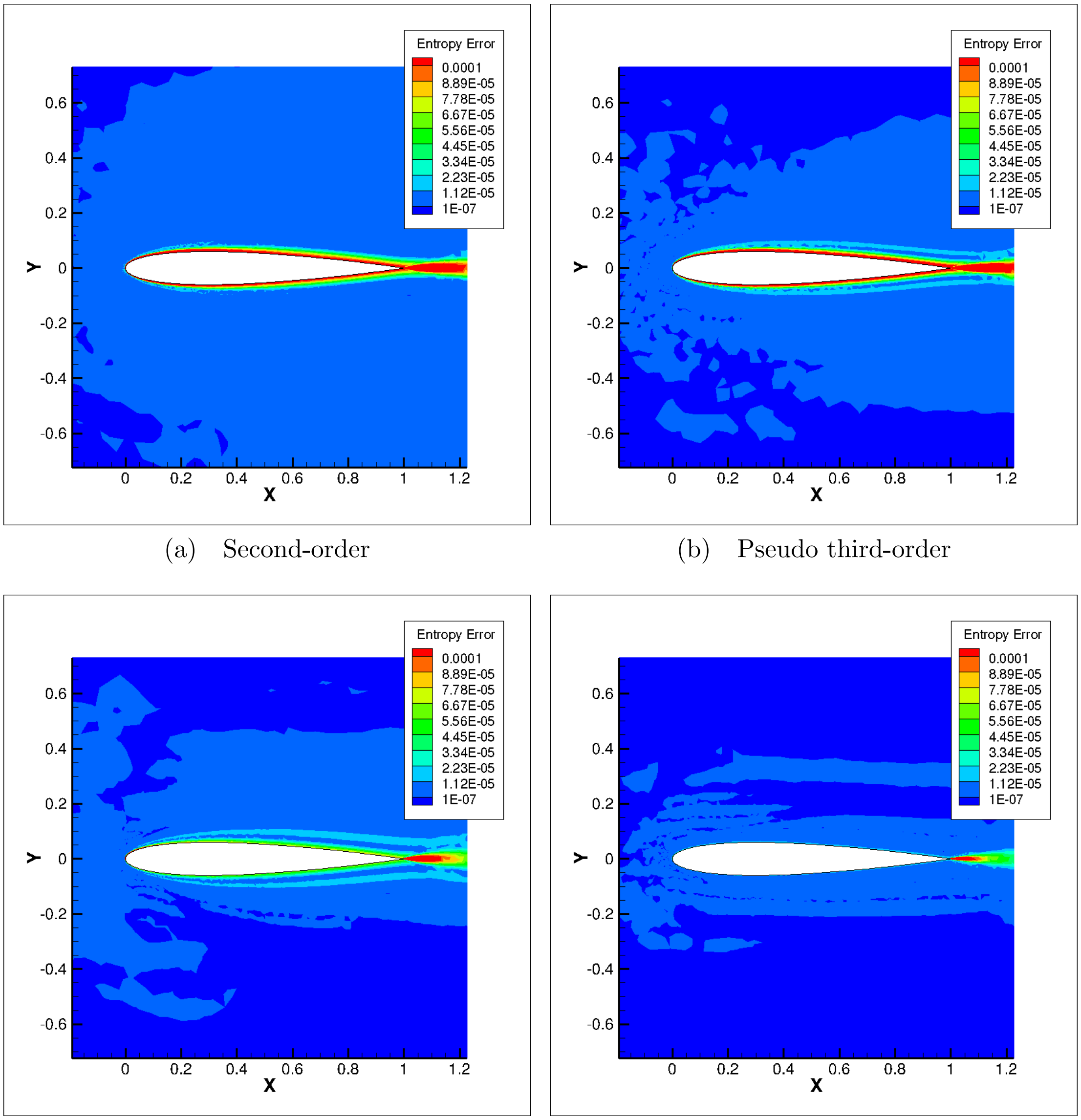
The next simulation case for the NACA 0012 airfoil is again a fully subsonic case with the freestream Mach number set at 0.63 and an angle of attack of two degrees. This test case has a more complicated set of features in the solution since the flow is no

Table 5.19 Error norms for all schemes on the NACA airfoil with $M_\infty = 0.3$ and $\alpha = 0$

2^{nd}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	1.7464e-06	—	7.6894e-06	—
1	3.0639e-07	5.70	1.8079e-06	4.25
2	9.5347e-08	3.21	7.9934e-07	2.26
pseudo 3^{rd}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	1.3393e-06	—	5.7809e-06	—
1	2.6109e-07	5.13	1.6837e-06	3.43
2	9.4097e-08	2.77	8.5558e-07	1.97
3^{rd}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	1.3535e-06	—	6.1695e-06	—
1	2.0160e-07	6.71	1.3325e-06	4.63
2	4.9108e-08	4.11	4.4664e-07	2.98
4^{th}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	2.7145e-06	—	1.1257e-05	—
1	1.4450e-07	18.78	8.6028e-07	13.09
2	3.2523e-08	4.44	2.8045e-07	3.07

longer symmetric over the upper and lower surfaces. Since the flow is still subsonic in the computational domain, the same analysis from the previous test case can be used to measure the performance of the reconstruction schemes.

The error norms from the four reconstruction schemes are given in Table 5.20 using the relation from Equation 5.7. In these results, the geometric weighting parameter for the higher order methods is set to two since it produced the lowest errors, albeit with poorer convergence. The table reveals that the errors are again lower for the third- and fourth-order schemes than the second-order schemes. The lower errors can be seen visually



(a) Second-order

(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

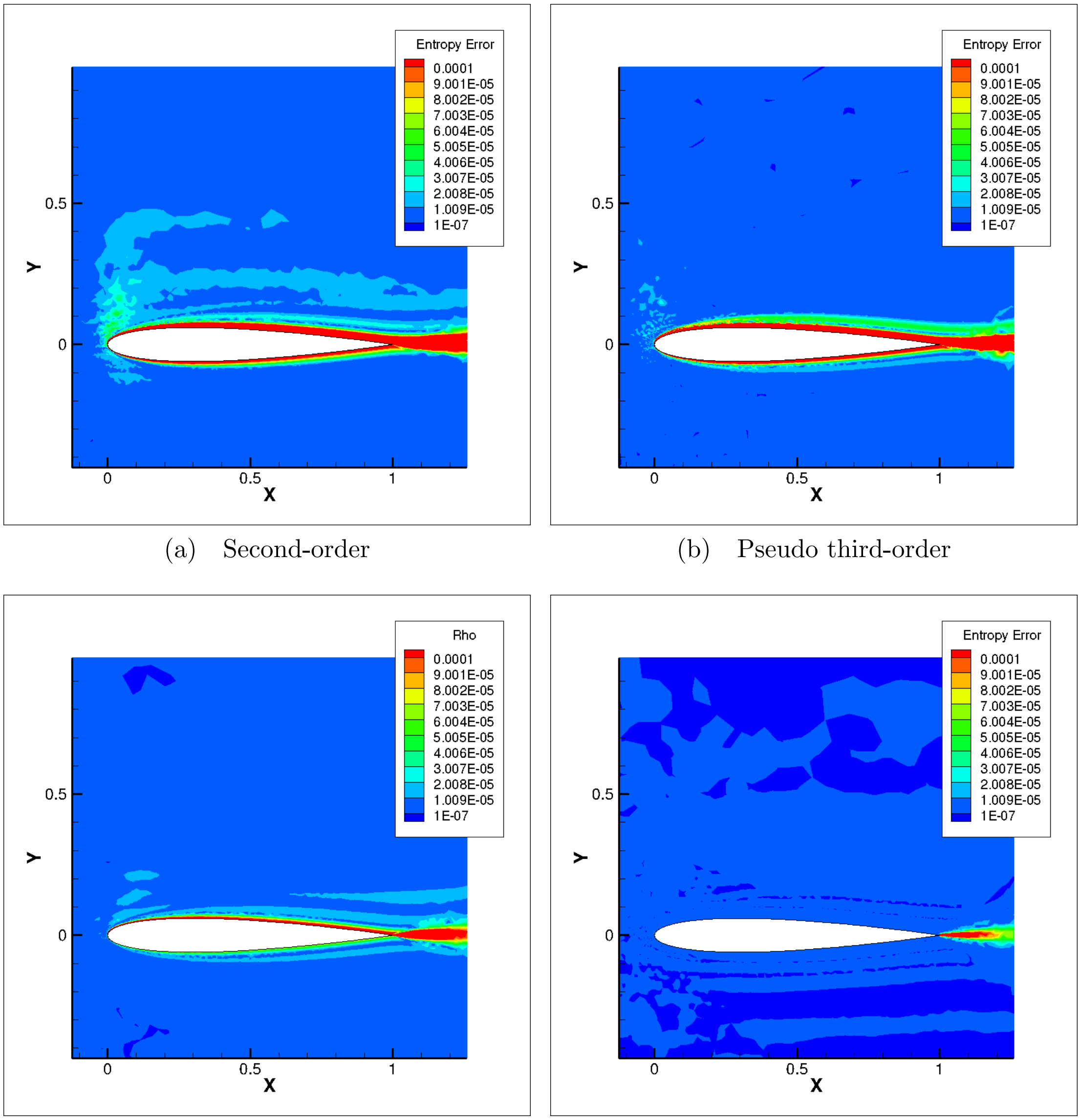
Figure 5.21 Deviation in entropy for all reconstruction schemes with $M_\infty = 0.3$ and $\alpha = 0$

from the solutions on the finest grid as shown in Figure 5.22. From the solution plots, the error around the airfoil can be seen to decrease as the reconstruction scheme's accuracy is increased. The error plots of the solution also partially illuminate the low pressure region along the upper surface of the airfoil. The entropy errors in the beginning of this region are larger for the second-order scheme and are successively reduced as the reconstruction order increases.

Table 5.20 Error norms for all schemes on the NACA airfoil with $M_\infty = 0.63$ and $\alpha = 2$

2^{nd}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	9.2434e-06	—	3.6432e-05	—
1	1.2421e-06	7.44	5.7566e-06	6.33
2	2.7544e-07	4.51	1.8911e-06	3.04
pseudo 3^{rd}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	6.6389e-06	—	2.6125e-05	—
1	9.1713e-07	7.24	4.7373e-06	5.51
2	2.4219e-07	3.79	2.0181e-06	2.35
3^{rd}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	2.8738e-06	—	1.2476e-05	—
1	7.0350e-07	4.08	3.4162e-06	3.65
2	1.4593e-07	4.82	9.5415e-07	3.58
4^{th}				
Grid Index	Error, L_1	Ratio	Error, L_2	Ratio
0	1.4951e-06	—	6.9765e-06	—
1	2.4243e-07	6.17	1.4260e-06	4.89
2	5.6406e-08	4.30	4.7306e-07	3.01

The pressure contours from the solution using the fourth-order reconstruction scheme on the index 2 grid is shown in Figure 5.23. At this grid resolution, the solutions from all four reconstruction schemes are similar enough that differences between them is difficult to render. The differences in the schemes can be more appropriately examined by their performance on the coarsest grid, index 0. The plot of the coefficient of pressure along the upper surface of the airfoil, that includes the low pressure region, for all schemes is given



(a) Second-order

(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

Figure 5.22 Deviation in entropy for all reconstruction schemes with $M_\infty = 0.63$ and $\alpha = 2$

in Figure 5.24. Including in the plot is a reference C_P distribution which is taken to be the second-order solution on the finest grid. The distribution of C_P for all schemes is nearly identical except in the region of lowest pressure towards the front of the upper surface. A detail plot of the regions is given in Figure 5.25. Examining the results in detail, it is seen that the C_P distribution approaches the reference distribution as the reconstruction order increases, though the pseudo third-order and formal third-order schemes appear to be roughly similar. The largest improvement, from the second-order to the fourth-order reconstruction, can be viewed in the pressure contours. In Figure 5.26, the pressure contours from these two solutions are show in detail. The feature of interest is highlighted by the rectangles, bordered in black. The fourth-order solution, in Figure 5.26b, shows better resolution in the contours for the lowest pressure region.

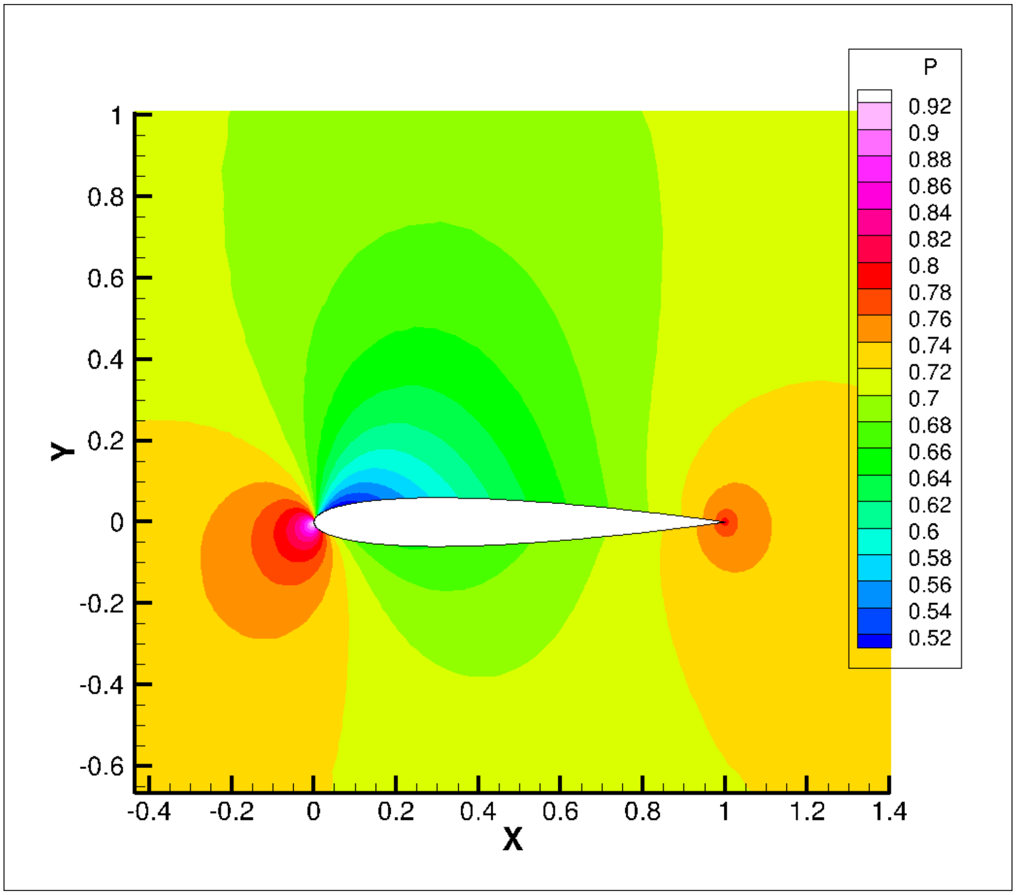


Figure 5.23 Pressure contours for the fourth-order solution on the NACA 0012 airfoil using $M_\infty = 0.63$ and $\alpha = 2$

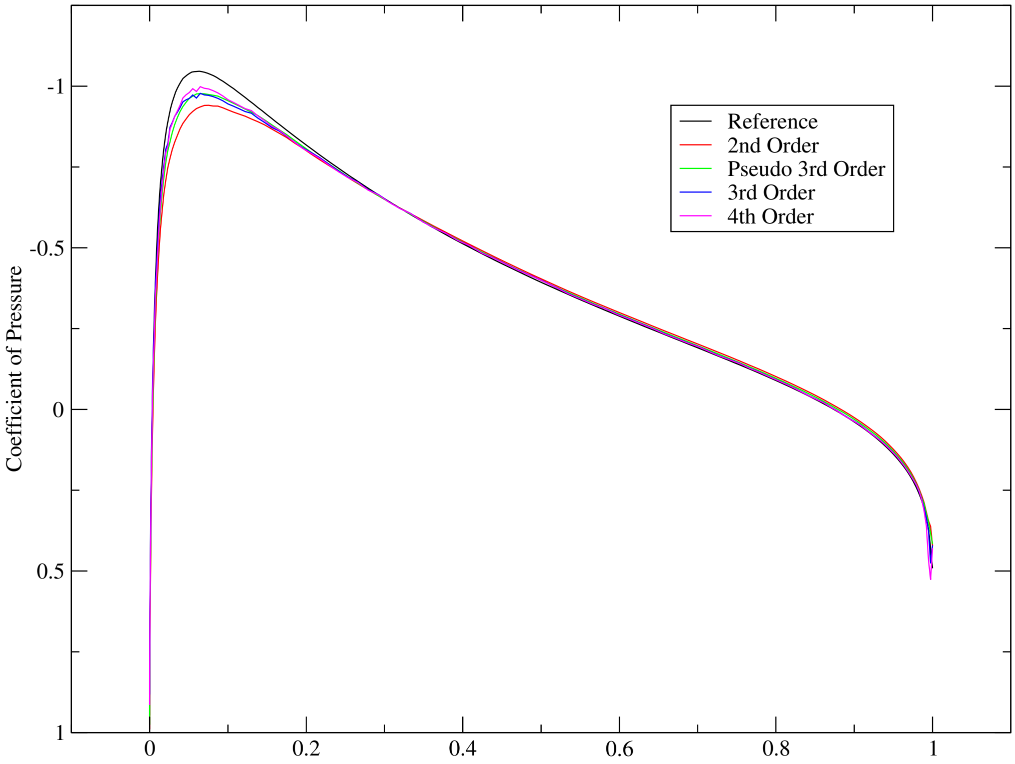


Figure 5.24 C_P plot over the upper airfoil surface on the index 0 grid using $M_\infty = 0.63$ and $\alpha = 2$

Transonic Solution with $M_\infty = 0.8$ and $\alpha = 1.25$

The final simulation of the NACA 0012 airfoil is a transonic solution with the freestream Mach number set at 0.8 and an angle of attack of 1.25 degrees. With these conditions, the flow field experiences two shocks across the airfoil surface; a stronger shock on the upper surface and a weaker shock on the lower surface. The appearance of shocks in the solution introduces oscillations in the reconstruction of the variables, giving an opportunity to investigate the inclusion of flux limiting. Since the flow is no longer isentropic, there is no attempt to quantify the performance of the schemes. Rather, the assessment of the schemes in this section will rely on qualitative discussions of the solutions. The flow solver in this section is restricted to solving the solution on the finest grid, index 2 from above.

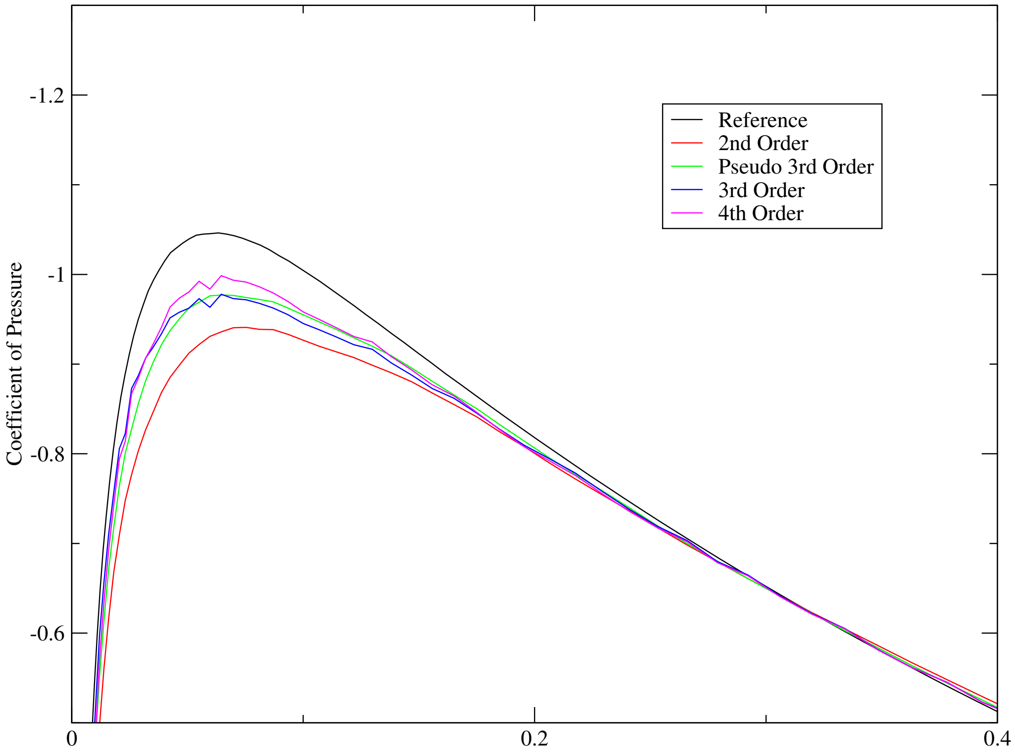
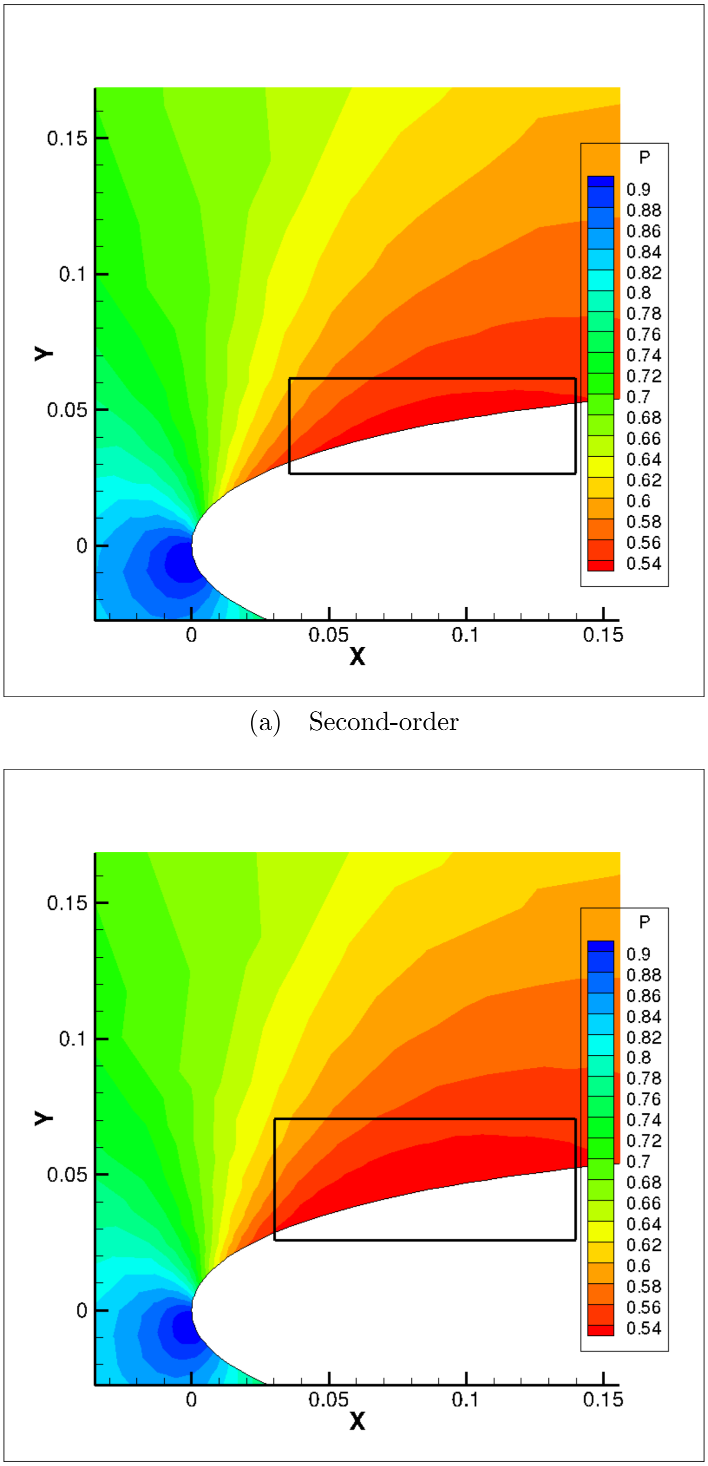


Figure 5.25 Detail of the C_P plot on the index 0 grid using $M_\infty = 0.63$ and $\alpha = 2$

The solution for all reconstruction schemes is initially computed with out a flux limiter. For the third- and fourth-order schemes, the geometric weight is set to two to help enforce solution locality. The presence of the shocks in the solution made it difficult for the solution to converge with respect to the L_2 norm of the update vector. Instead, the L_2 norm of the residual vector is used to assess when the solution has converged. The residual vector history for all four methods can be seen in Figure 5.27. The residual vectors show that, other than for the second-order method, all methods have difficulty in converging the solution. Oscillations in the solutions can be seen in a plot of the coefficient of pressure along the upper and lower surfaces in Figure 5.28. All solution methods produce similar C_P values and have the classic Gibbs-like oscillation at the shocks.

The solutions were then computed using flux limiters. For the second-order methods, Venkatakrishnan's limiter, as described by Equation 3.26, is used. The third- and fourth-order schemes use the limiter of Michalak and Ollivier-Gooch [17] as described in Chapter 3. This was found to produce better results than the limiter given by Nejat [28]. The third-order scheme used a geometric weight of two while the fourth-order scheme was restricted to a value of one for stability reasons. For the proceeding results, the parameter K, introduced in Chapter 3 with Venkatakrishnan's limiter, is set to a value of 1.0. This value was found to produce reasonable results for all reconstruction schemes. The use of the limiter did stall convergence, particularly in the higher order methods, which can be seen in the residual vector history shown in Figure 5.29 for all methods. The use of the limiter did positively affect the C_P distribution on the airfoil surface. The plot of the C_P values for all methods on both the upper and lower surfaces is given in Figure 5.30. While the pseudo third-order method still shows oscillations in the solution, all other reconstruction methods are seen to have lost the oscillations around the shocks. The pressure contours of the solution from all reconstruction methods is shown in Figure 5.31. The contours are very similar for all solutions, in part because of the interpolation scheme to render the contours and because the solutions are very similar on this refined grid.

While this is not an extensive test of the higher order reconstruction schemes' ability to compute a solution with shocks present, the results indicate that it is possible, even in more complicated geometries. The results are also suggestive that the higher order methods do not give as great a benefit as the lower order methods. The inclusion of the limiter does modify the reconstruction order, at least locally, making it difficult to comment on global accuracy. Taking the fourth-order scheme as an example, the regions where the limiter is active can be seen in Figure 5.32. The contours of ϕ show that it is active mostly near the airfoil surface and around the shocks. However, the contours of σ, which limit the second and higher derivatives, show that it acts in a more diffuse region around the airfoil.



(a) Second-order

(b) Fourth-order

Figure 5.26 Detail of pressure contours for second- and fourth-order reconstructions with $M_\infty = 0.63$ and $\alpha = 2$

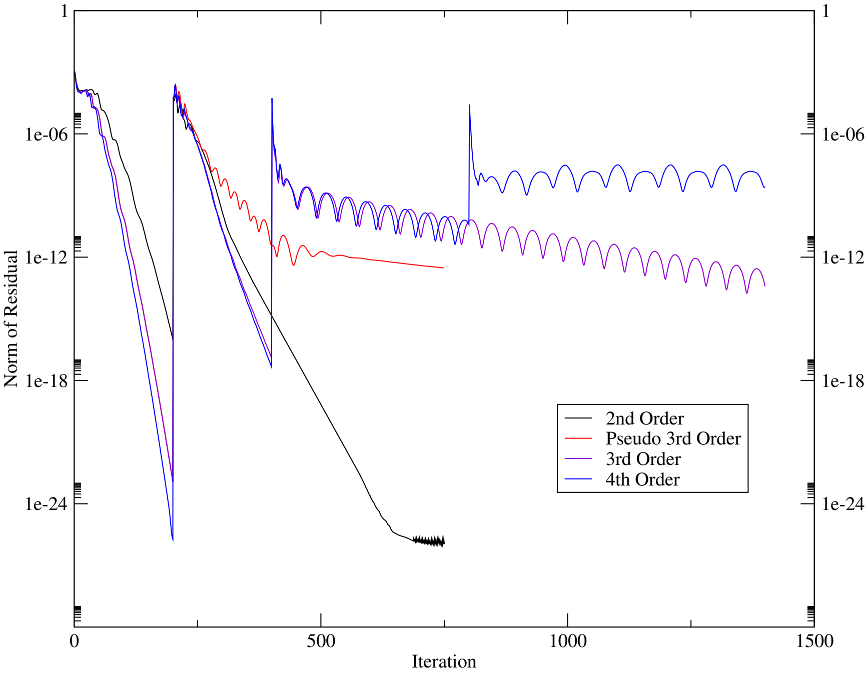


Figure 5.27 L_2 norm of the residual vector for the unlimited solution from the transonic airfoil case

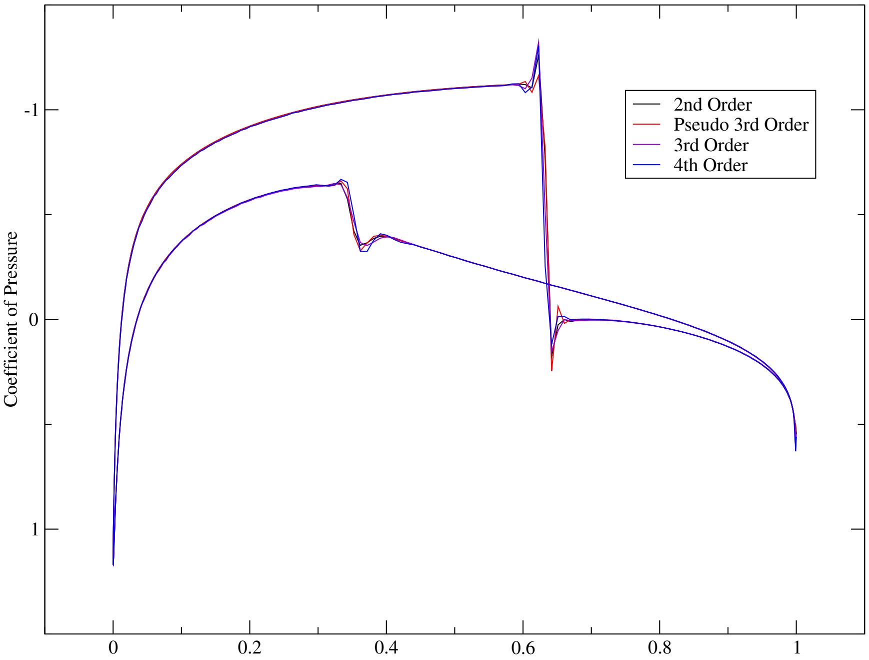


Figure 5.28 C_P plot from all methods with unlimited reconstruction on the upper and lower surfaces of the airfoil in transonic flow

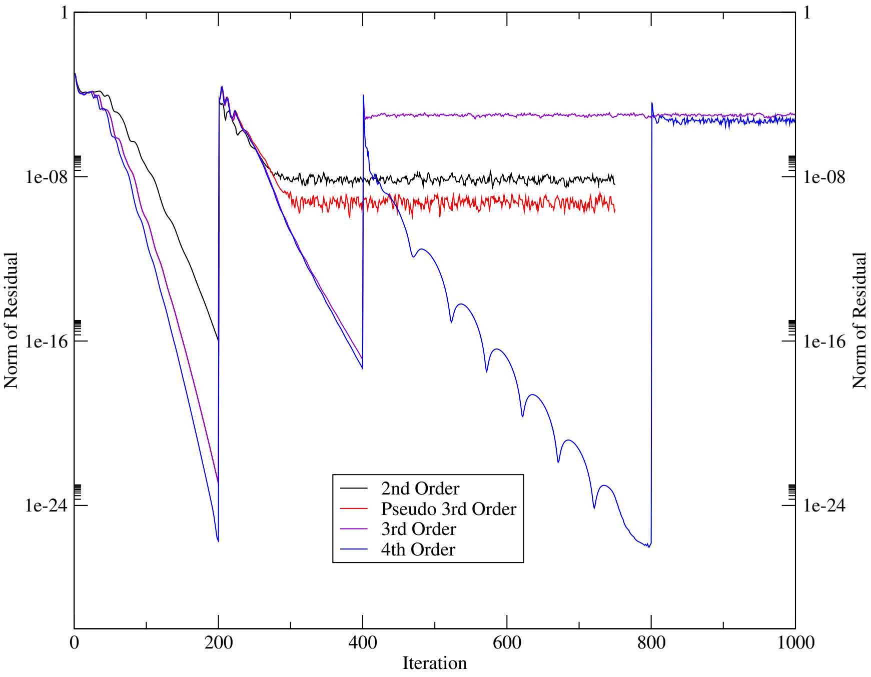


Figure 5.29 L_2 norm of the residual vector for the limited solution from the transonic airfoil case

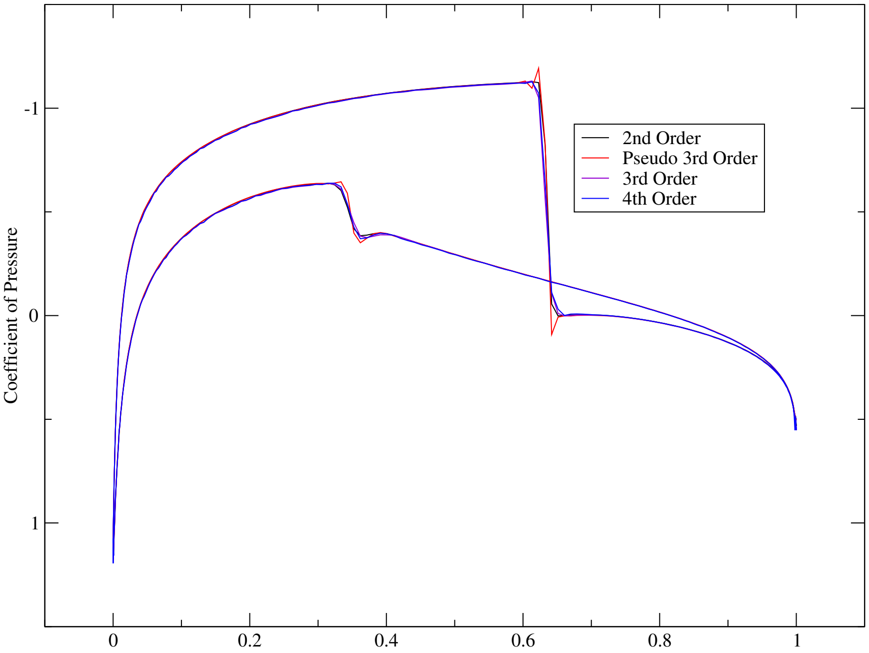
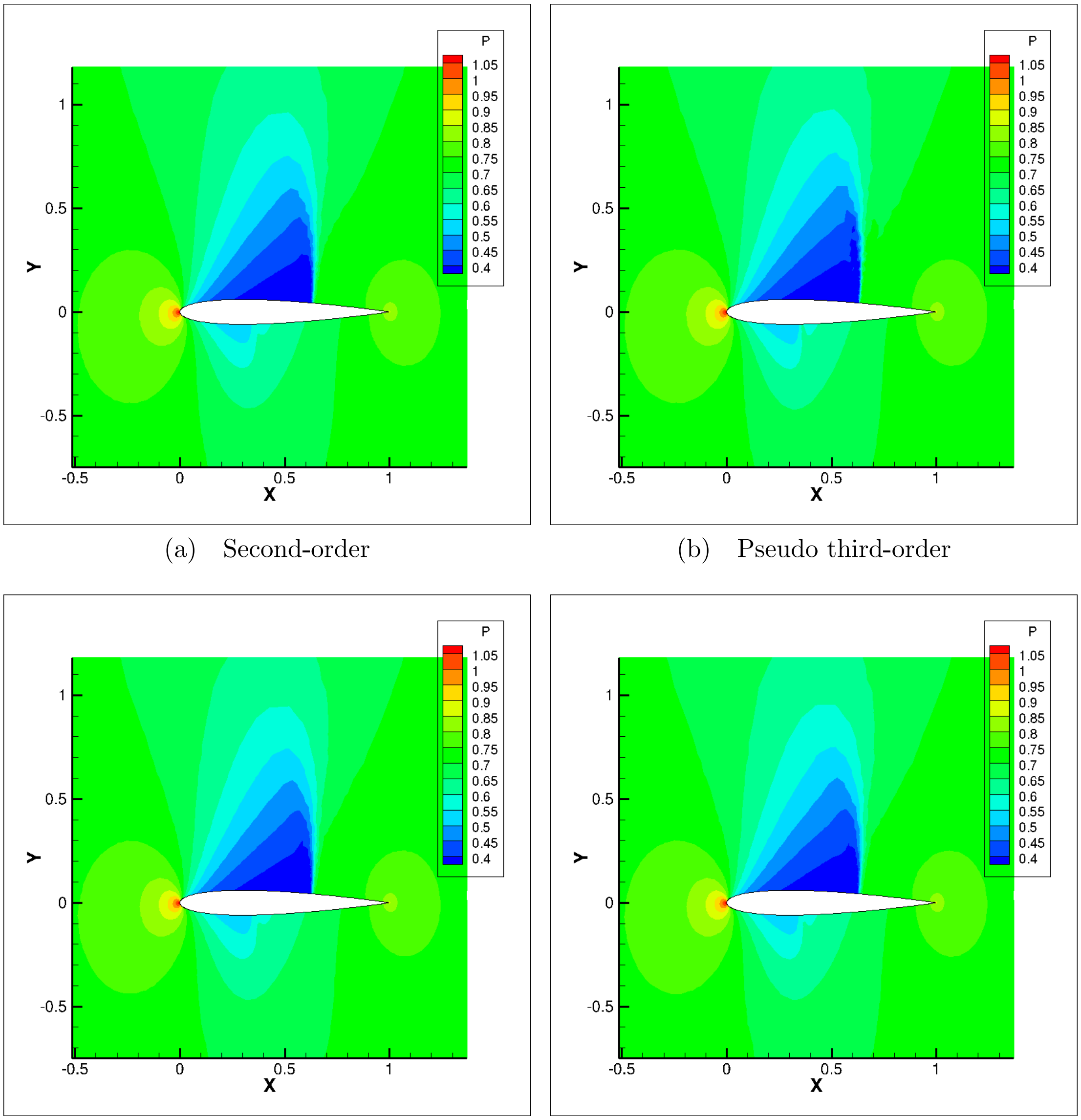


Figure 5.30 C_P plot from all methods with limited reconstruction on the upper and lower surfaces of the airfoil in transonic flow



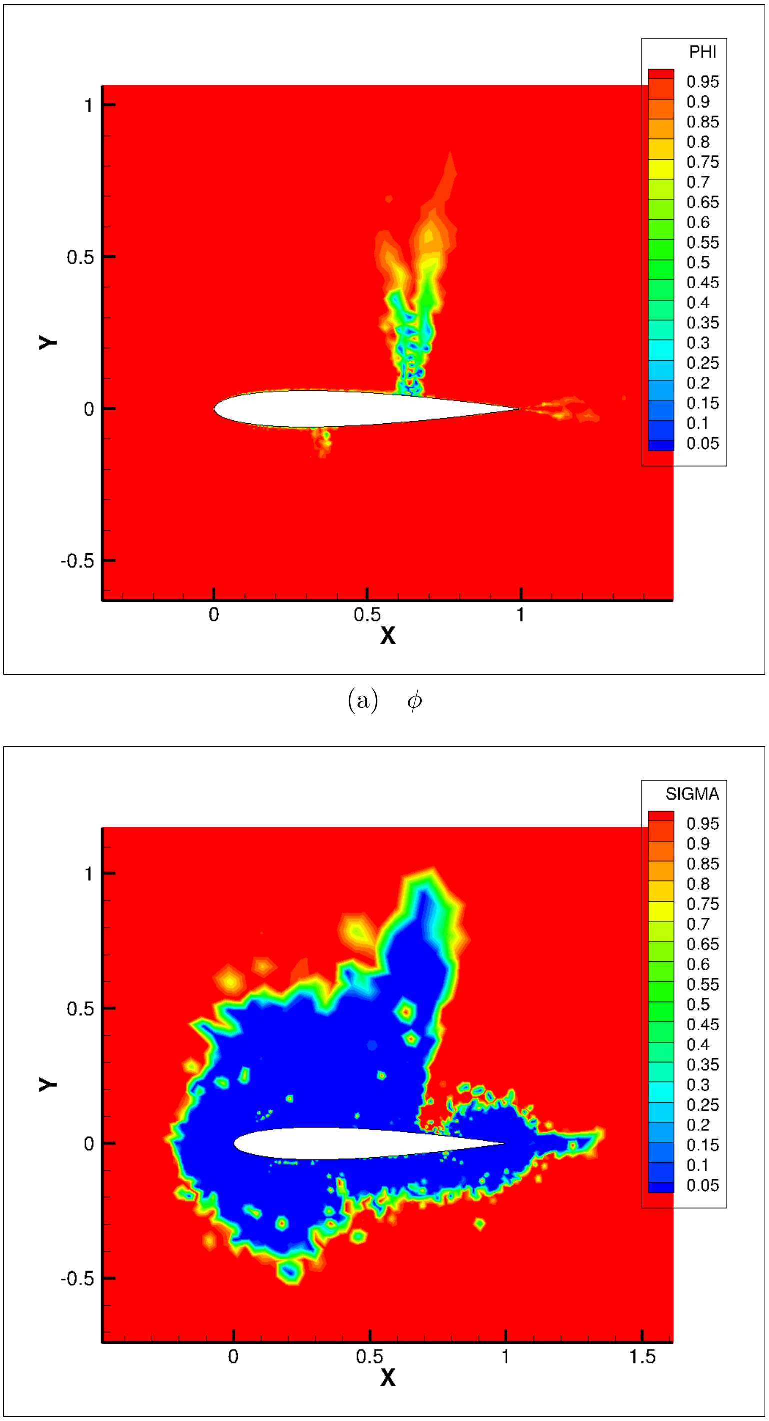
(a) Second-order

(b) Pseudo third-order

(c) Third-order

(d) Fourth-order

Figure 5.31 Pressure contours from the limited solution of the transonic airfoil case



(a) ϕ

(b) σ

Figure 5.32 Contour plot of minimal values of ϕ and σ for the fourth-order scheme on the transonic airfoil solution

CHAPTER 6

UNSTEADY FLOW SOLVER SIMULATION RESULTS

The emphasis of this chapter is the demonstration of the reconstruction algorithm's ability to simulate time-dependent solutions using an implicit time marching scheme. The effectiveness of the new algorithm is ascertained in part by comparison with the base second-order and pseudo third-order schemes. The first simulation is of passive vortex convection where an analytical solution exists that is used to quantify the errors generated by the numerical schemes. The second, and final, simulation is of vortex shedding. While an exact solution is not available, qualitative comparisons can be made to assess the performance of the schemes.

Vortex Convection

The first test case chosen to assess performance with unsteady flow is the two-dimensional convection of an isentropic vortex in freestream flow. This particular case is modeled after a similar work by Yee [39] with a slight modification to the initial temperature. The mean flow is given freestream conditions with $\rho_\infty = 1$, $M_\infty = 0.5$, $u_\infty = M_\infty$, $v_\infty = 0$, $P_\infty = 1/\gamma$, and $T_\infty = P_\infty/\rho_\infty$. The initial freestream conditions are then aggregated by an isentropic vortex using the following perturbations:

$$\begin{aligned} \delta u &= \frac{-\beta}{2\pi}\, e^{(1-r^2)/2}\,(y - y_0) \\ \delta v &= \frac{\beta}{2\pi}\, e^{(1-r^2)/2}\,(x - x_0) \\ \delta T &= \frac{-\beta^2(\gamma - 1)}{8\pi^2\gamma}\, e^{1-r^2} \end{aligned} \tag{6.1}$$

Here, β is the vortex strength, (x_0, y_0) are the coordinates of the vortex center, and r is the Euclidean distance from the center of the vortex. In the following results, β was set to five and the vortex was centered at $(x_0, y_0) = (5, 0)$. The grids utilized in this simulation span the space $(x, y) \in [0, 150] \times [-5, 5]$ and are comprised of quadrilateral elements. The details of the grids used are given in Table 6.1. The initial solution using these conditions can be seen in Figure 6.1. In all cases, the time step was fixed to $\Delta t = 0.0125$ and applied for 5000 steps.

Table 6.1 Grid details for the vortex convection simulation

Grid Index	Points in y	Points in x	Total Points	Δx
0	51	751	38301	0.2
1	61	901	54961	0.1667
2	71	1051	74621	0.1429
3	81	1201	97281	0.125

The constant x boundaries are subject to farfield conditions and the constant y boundaries are treated as inviscid surfaces. This is not a correct treatment since the initial conditions have support extending beyond the computational boundaries; however, this simplification is not so errant considering that the perturbations drop off rapidly away from the vortex center.

The simulations began by initializing the numerical solution to uniform flow added with the perturbations described in Equations 6.1. The initial conditions are integrated over the CVs using the Gaussian integration scheme over the constituent triangles to accurately compute the CV averaged values of the conserved variables. This is an important step for the high-order schemes since the assumption that the CV average value be the point value at the nodes will introduce an $O(h^2)$ error. Due to the nature of unsteady solutions, any initial error will propagate along in time further contaminating the final computed solution. Unlike the steady solutions of the previous chapter, the goal of the flow solver is to drive the unsteady residual on the right hand side of Equation 2.21 to zero. Since convergence to zero is not realistic, the target convergence level is set at 10^{-12}, which all schemes were able to achieve before convergence stalled. The number of Newton iterations required to converge the unsteady residual to the desired tolerance for the four reconstruction schemes on all grids is shown in Table 6.2.

Table 6.2 Number of Newton iterations for each reconstruction scheme

Order	Geom. Weight	51 pts. y	61 pts. y	71 pts. y	81 pts. y
2^{nd}	—	10	10	10	12
Pseudo 3^{rd}	—	10	12	12	14
3^{rd}	0	10	10	10	10
—	1	10	10	10	10
—	2	10	10	10	10
4^{th}	0	10	10	10	10
—	1	10	10	10	10
—	2	10	10	12	12

All grids were used to solve the convection of the vortex with all spatial reconstruction schemes. For the third- and fourth-order schemes, the geometric weighting parameter was set to zero, one, and two, although only results from the parameter set at two are presented as they produced the lowest error in a normed sense. The L_1 norm of the error in density for the four schemes on all the grids is plotted in Figure 6.2. From the plot, it can be seen that the pseudo third-order scheme is able to produce less error in the density compared to both the second-order scheme and the third-order scheme. The third-order scheme does have a slightly better observed order of accuracy, seen in Table 6.3, but still does not achieve the theoretical order of accuracy of three. The fourth-order schemes fares much better. Not only does it achieve the theoretical order of accuracy, the fourth-order scheme shows a dramatic improvement in reducing the L_1 norm of the error in density compared to all other schemes. In fact, the error on the coarsest grid is lower than the error on the finest grid for the other three reconstruction schemes. The error data in Table 6.3 further demonstrates that while the pseudo third-order scheme is able to produce lower error than the base second-order scheme it can only achieve an order of accuracy of two.

Table 6.3 L_1 norm of error for all schemes on the vortex grid sequence

2^{nd}	ρ		ρu		ρv		E	
Npts y	Error	Order	Error	Order	Error	Order	Error	Order
51	5.92e-04	—	1.35e-03	—	9.80e-04	—	1.52e-03	—
61	4.30e-04	1.75	1.01e-03	1.58	7.62e-04	1.38	1.11e-03	1.73
71	3.28e-04	1.77	7.75e-04	1.74	6.03e-04	1.52	8.44e-04	1.76
81	2.56e-04	1.84	6.07e-04	1.83	4.82e-04	1.67	6.61e-04	1.83
pseudo 3^{rd}	ρ		ρu		ρv		E	
Npts y	Error	Order	Error	Order	Error	Order	Error	Order
51	3.75e-04	—	8.01e-04	—	6.78e-04	—	9.98e-04	—
61	2.62e-04	1.97	5.45e-04	2.11	4.73e-04	1.97	6.84e-04	2.07
71	1.90e-04	2.07	3.93e-04	2.12	3.43e-04	2.10	4.98e-04	2.05
81	1.42e-04	2.18	2.96e-04	2.12	2.59e-04	2.11	3.75e-04	2.12
3^{rd}	ρ		ρu		ρv		E	
Npts y	Error	Order	Error	Order	Error	Order	Error	Order
51	4.66e-04	—	8.10e-04	—	6.82e-04	—	1.26e-03	—
61	3.20e-04	2.05	5.44e-04	2.18	4.70e-04	2.03	8.51e-04	2.14
71	2.25e-04	2.30	3.83e-04	2.27	3.41e-04	2.08	5.89e-04	2.39
81	1.69e-04	2.15	2.80e-04	2.35	2.52e-04	2.27	4.30e-04	2.35
4^{th}	ρ		ρu		ρv		E	
Npts y	Error	Order	Error	Order	Error	Order	Error	Order
51	1.10e-04	—	1.68e-04	—	1.61e-04	—	2.59e-04	—
61	5.46e-05	3.87	8.46e-05	3.77	8.38e-04	3.59	1.33e-04	3.64
71	2.87e-05	4.18	4.54e-05	4.03	4.46e-05	4.09	7.25e-05	3.95
81	1.62e-05	4.26	2.59e-05	4.20	2.52e-05	4.29	4.14e-05	4.20

The difference between the analytical and computed solutions using the four spatial reconstruction schemes for all conserved variables can be seen in Figures 6.3, 6.4, 6.5, and 6.6. The finest grid was used for the second-order scheme. For the pseudo third- and formal third-order schemes, the grid with 61 points along the y axis was used. Both schemes produced errors on this mesh comparable to the errors on the finest grid with second-order reconstruction, which is demonstrated by the error norms in Table 6.3. The fourth-order scheme was able to compute a lower error solution on the coarsest grid. This again validates the assertion that high-order schemes are able to compute comparable solutions on coarser grids. The fourth-order scheme is clearly superior to the other schemes in terms of maintaining the initial solution as the mean flow convects it downstream, even on relatively coarser grids. The cost incurred is that more calculations are performed per node.

The efficiency of the schemes can be compared by examining the time required to reach the final computed solution. For this comparison, the errors from the second-order scheme will be the target for the other schemes to match. The pseudo third- and formal third-order schemes, as mentioned above, reproduce similar errors on the grids with 61 points in the y direction. For the fourth- order scheme, a new grid with 41 points in the y direction is used. This new grid is used to drive the errors in the fourth-order reconstruction closer to those produced by the second-order scheme. The new grid required ten Newton iterations to converge the unsteady residual just as the other coarser grids needed with the fourth-order reconstruction scheme. The error data is collected in Table 6.4 to provide easier comparison. The assessment of efficiency is again measured by time results. For these simulations, the time results from a ten runs are sampled and averaged, as before with the annulus time results of Chapter 5. The timing results are tabulated in Table 6.5 which shows the total time required along with the total time averaged over the iterations, both time steps and time steps and Newton iterations, and nodes. The results illustrate that while the second-order scheme is used on a grid with the most points it still has the quickest time per node

Table 6.4 Comparisons of the L_1 norm of error for all schemes on their respective grids for the vortex convection problem

Order, Grid	ρ	ρu	ρv	E
2^{nd}, 81	2.56e-04	6.07e-04	4.82e-04	6.61e-04
pseudo 3^{rd}, 61	2.62e-04	5.45e-04	4.73e-04	6.84e-04
3^{rd}, 61	3.20e-04	5.44e-04	4.70e-04	8.51e-04
4^{th}, 41	2.05e-04	3.45e-04	3.32e-04	4.70e-04

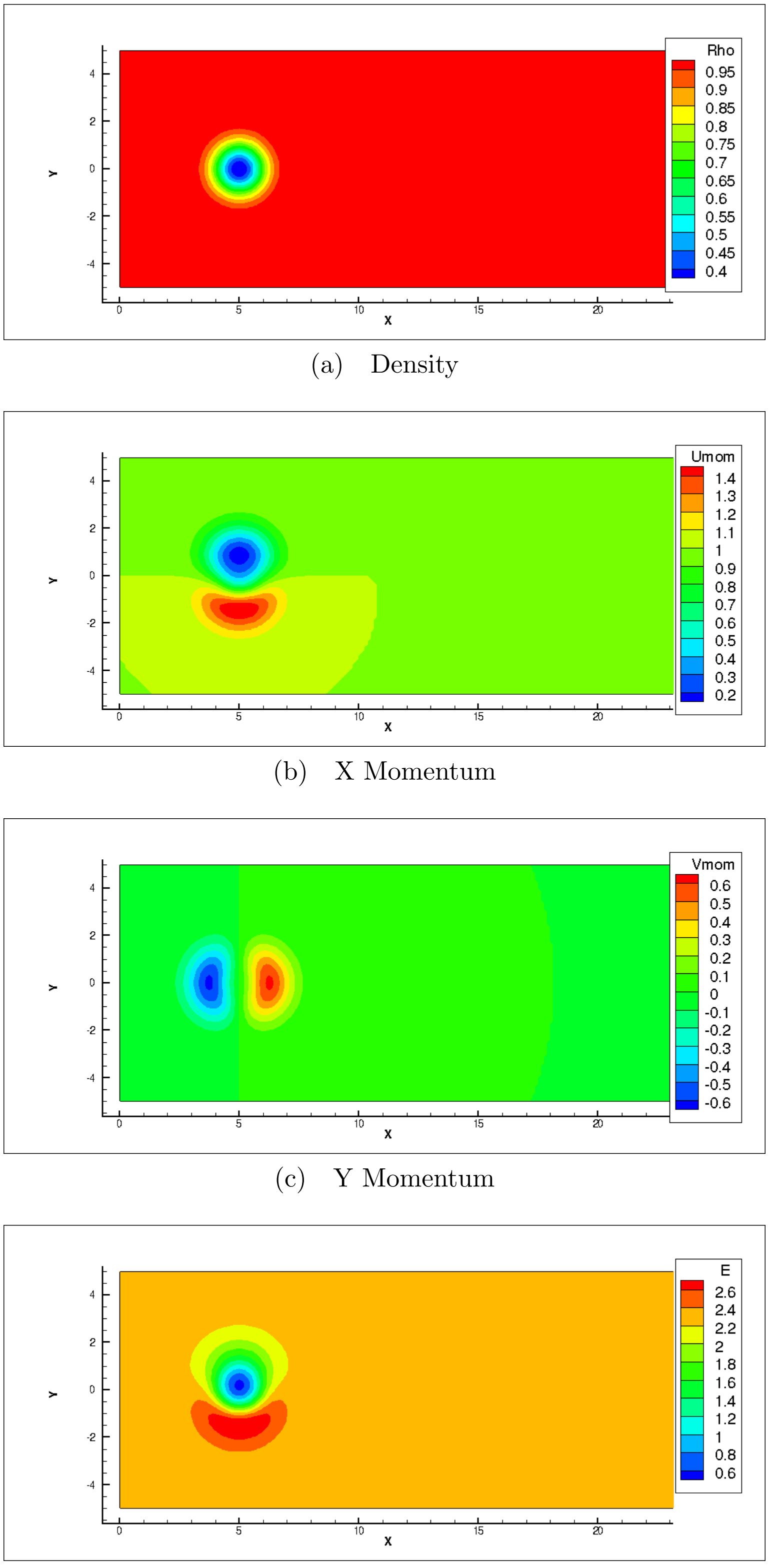
when averaged over the iterations. This is expected since the second-order scheme performs the fewest operations compared to the other methods. However, its total time is greater than the time required by all other schemes. The fourth-order reconstruction is able to out perform all other methods by producing the lowest error in all variables in the least total time. This illustrates that while the scheme requires more computing resources per node it is very competitive to low-order methods. The verdict is not so pleasant for the formal third-order scheme. It requires more time that the pseudo third-order scheme to produce a solution with comparable error to the second-order method. The errors between all spatial reconstruction schemes can also be seen in a plot of the pressure from the computed solution in Figure 6.7. The cut is taken along the x axis with $y = 0$, where the vortex center is initialized. The analytical solution is included to show the true pressure distribution along the cut. The plot illustrates that the third-order scheme shows additional dissipation of the pressure compared to the other schemes. The fourth-order scheme is seen to over-predict the pressure at in the vortex core but otherwise tracks the structure well. A plot of the core pressure tracked in computational time is shown in Figure 6.8. The exact solution of the convection problem retains the initialized core pressure and convects it downstream as time progresses. The oscillations in the core pressure for the numerical solutions are an artifact of the reconstruction procedure. Since the core pressure is not guaranteed to

Table 6.5 Timing results for the second-, third-, and fourth-order schemes in seconds for the vortex convection problem

Order, Grid	Total Time (s)[*]	Time/It./Newton It./node (s)
2^{nd}, 81	5.4370e+04	9.3150e-06
pseudo 3^{rd}, 61	3.1338e+04	9.5032e-06
3^{rd}, 61	4.8718e+04	1.7728e-05
4^{th}, 41	2.7499e+04	2.2320e-05

[*] Executed on an Intel® Core™ i5 750.

coincide with a grid point at a given time step, the core pressure is estimated from the reconstruction of the nearest control volume. The plot indicates that all numerical schemes begin by strengthening the vortex, demonstrated by a reduction in the pressure. However, all schemes reach a minimum pressure and begin to dissipate the vortex. The second-order schemes and the third-order scheme roughly agree on a minimum pressure, but the third-order scheme is seen to dissipate the core pressure more rapidly. The fourth-order scheme produces a much larger over-prediction in the core pressure and appears to posses a flatter slope as it begins to dissipate the core. The over-prediction of the core pressure by all schemes might be restrained by the inclusion of a slope limiter as described in Chapter 3.



(a) Density

(b) X Momentum

(c) Y Momentum

(d) Total Energy

Figure 6.1 Initial solution for the vortex convection problem

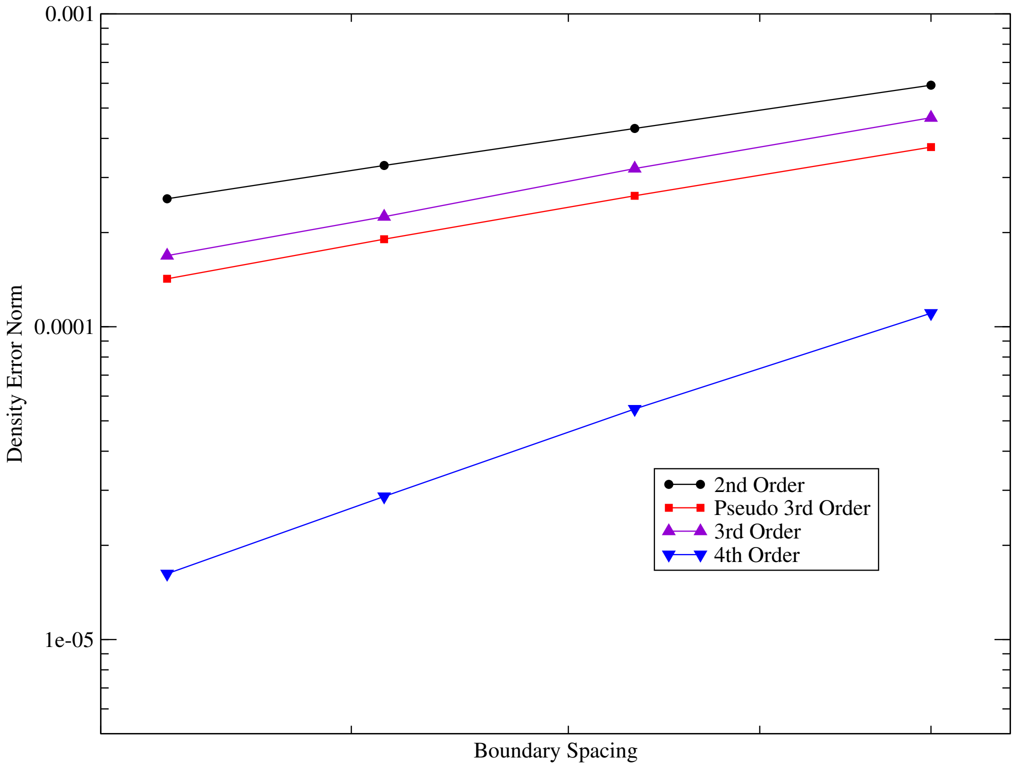
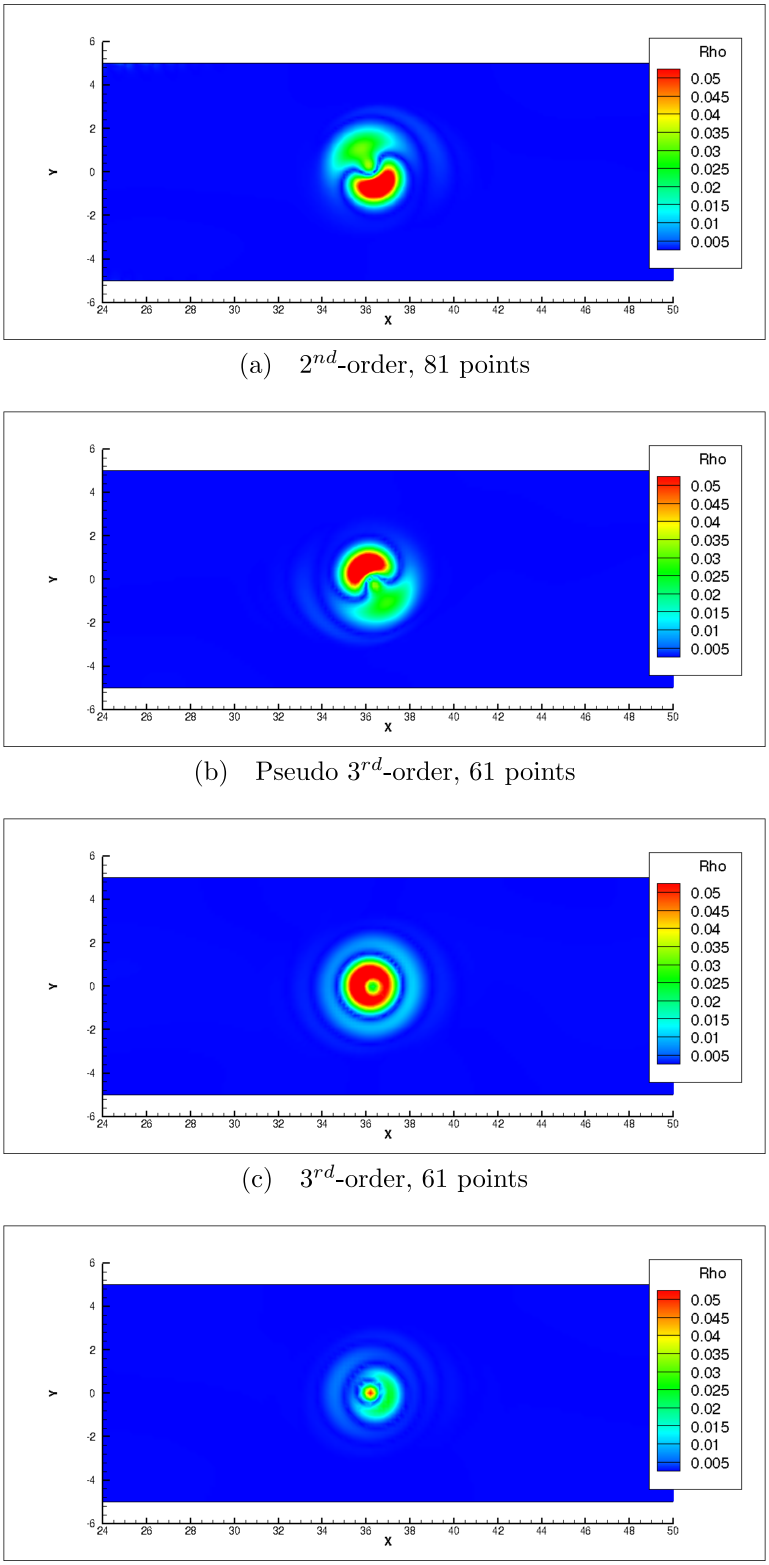


Figure 6.2 L_1 norm of density error for all schemes on the grid sequence for the vortex convection problem



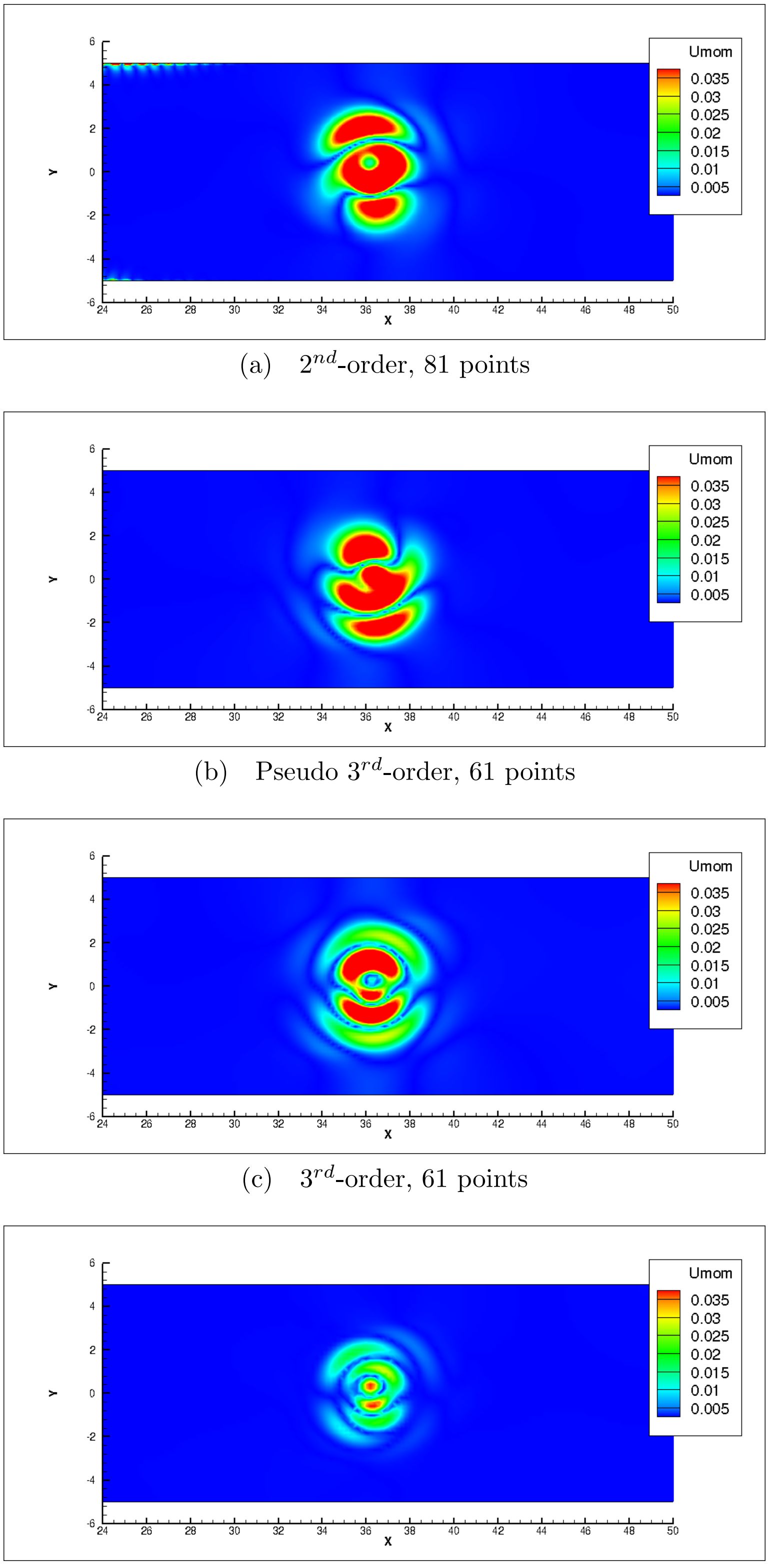
(a) 2^{nd}-order, 81 points

(b) Pseudo 3^{rd}-order, 61 points

(c) 3^{rd}-order, 61 points

(d) 4^{th}-order, 51 points

Figure 6.3 Error between the exact and numerical schemes for density



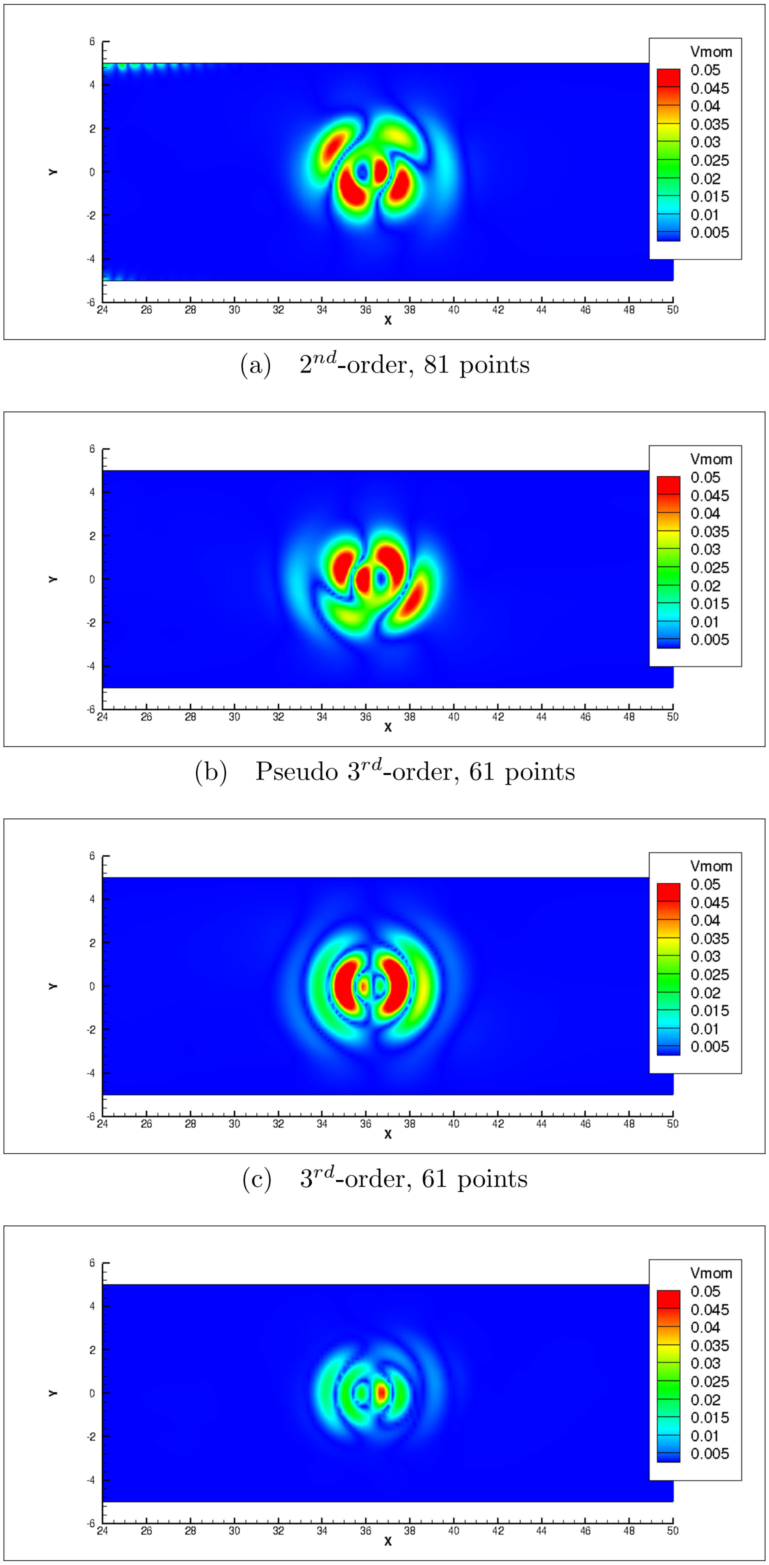
(a) 2^{nd}-order, 81 points

(b) Pseudo 3^{rd}-order, 61 points

(c) 3^{rd}-order, 61 points

(d) 4^{th}-order, 51 points

Figure 6.4 Error between the exact and numerical schemes for x momentum



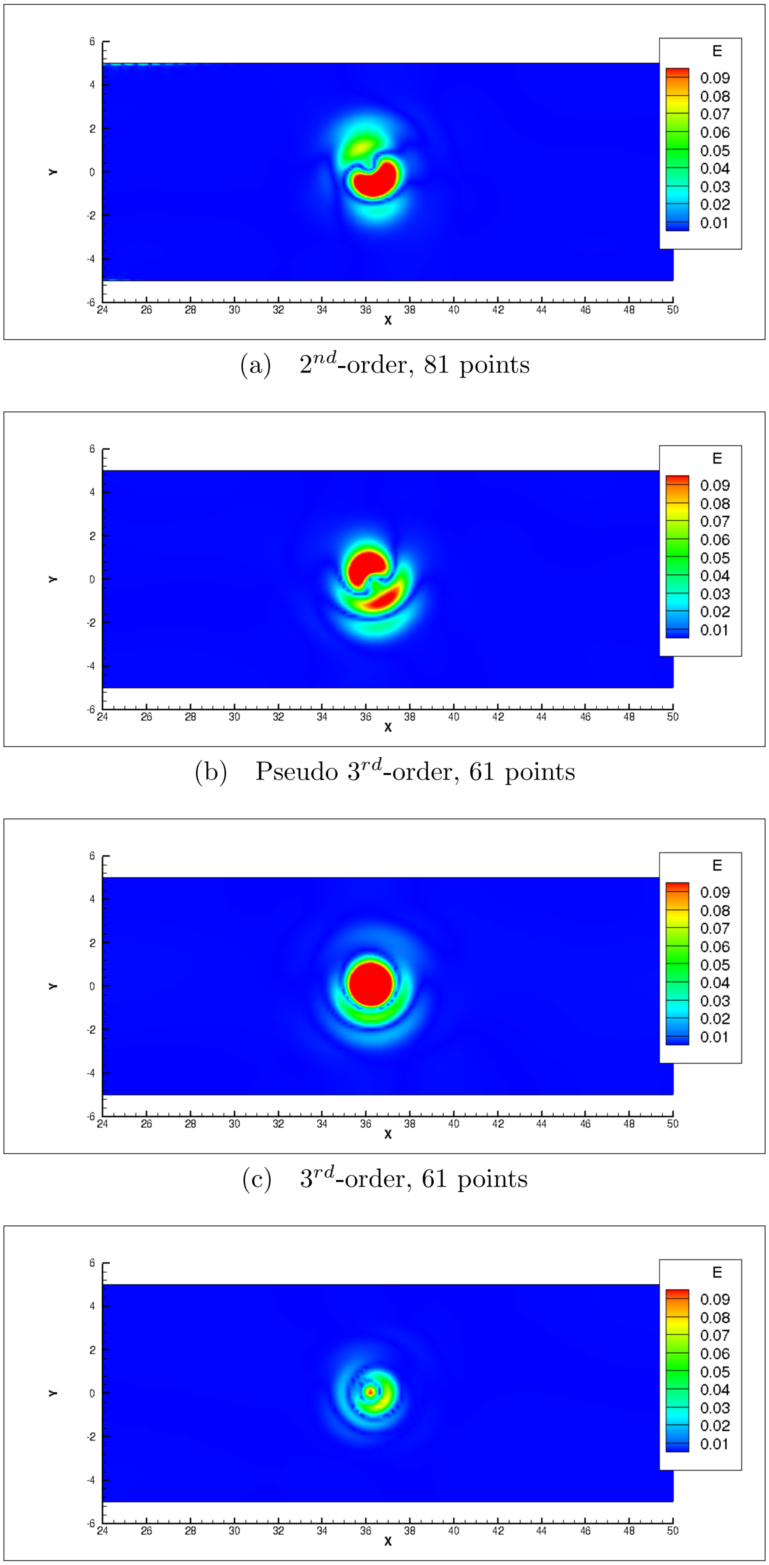
(a) 2^{nd}-order, 81 points

(b) Pseudo 3^{rd}-order, 61 points

(c) 3^{rd}-order, 61 points

(d) 4^{th}-order, 51 points

Figure 6.5 Error between the exact and numerical schemes for y momentum



(a) 2^{nd}-order, 81 points

(b) Pseudo 3^{rd}-order, 61 points

(c) 3^{rd}-order, 61 points

(d) 4^{th}-order, 51 points

Figure 6.6 Error between the exact and numerical schemes for total energy

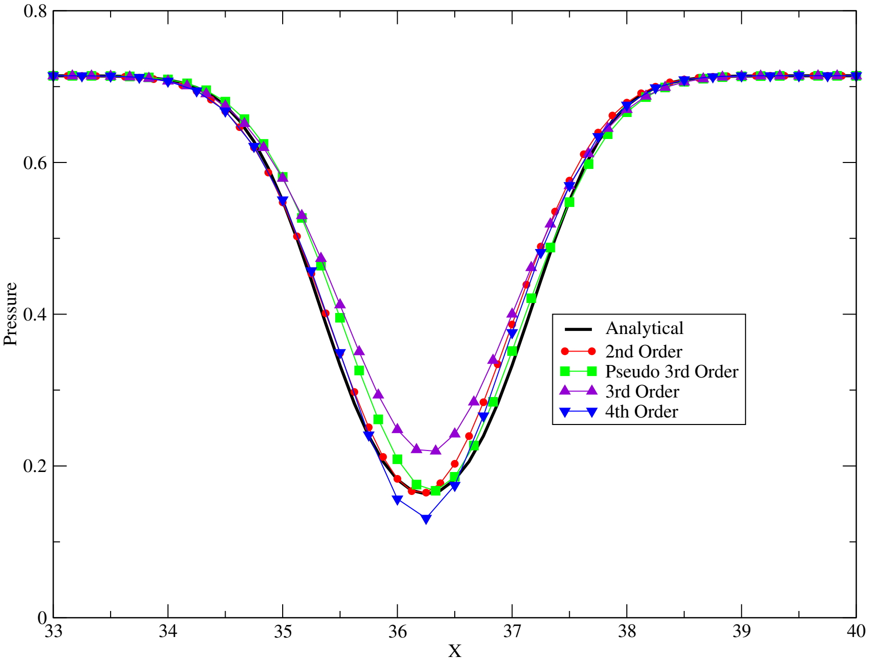


Figure 6.7 Pressure along a cut in the computed solution for vortex convection problem

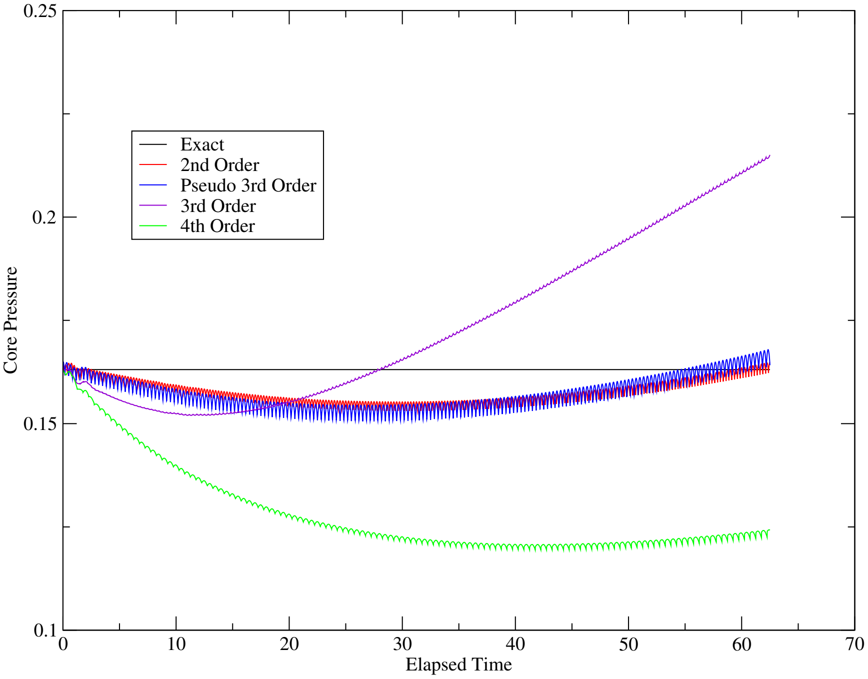


Figure 6.8 Core pressure plot as a function of elapsed computational time for the vortex convection problem

Vortex Shedding Over a Wedge

The final simulation discussed in this thesis is vortex shedding over a triangular wedge boundary. This test case was also used by Wang [40] to demonstrate qualitatively a high-order scheme's ability to capture and retain more solution detail, though in a finite-element solver in that work. The idea is to have freestream flow move over a triangular boundary; in this case, a unit equilateral triangle with one side parallel to the y axis placed so that one corner is directed into the flow. As a result of the geometry and placement, the flow experiences two sharp corners it must turn about. This geometrical condition coupled with numerical dissipation in the solver assures that vortices will be generated at the rear of the triangle and subsequently convected downstream. A view of the grid used in this case is shown in Figure 6.9. The grid is refined around the triangle boundary, which can be seen in detail in Figure 6.10. The back of the refined area is allowed to gradually transition to the coarser elements along the computational boundary. The grid is comprised of $41,217$ nodes and $82,211$ triangles.

The solver is initialized in a specific way for this geometry. Rather than start with freestream conditions, all examined spatial accuracies begin with a converged first-order solution. The first-order solution is computed using freestream conditions with $M_\infty = 0.2$ and allowed to converge, taking 800 iterations. Contours of density in the converged steady solution are shown in Figure 6.11. This solution is then used as the initial solution to the higher order reconstruction schemes. In each spatial accuracy case, the solver used a fixed time step of $\Delta t = 0.05$ and was ran for a total of 20,000 iterations, inclusive of the first-order iterations. The number of iterations was set high to ensure that enough vortices would be created to provide qualitative assessment of the reconstruction schemes. The freestream conditions were kept constant between the first-order and higher order simulations. Again, the third- and fourth-order solutions use two as the geometric weighting parameter.

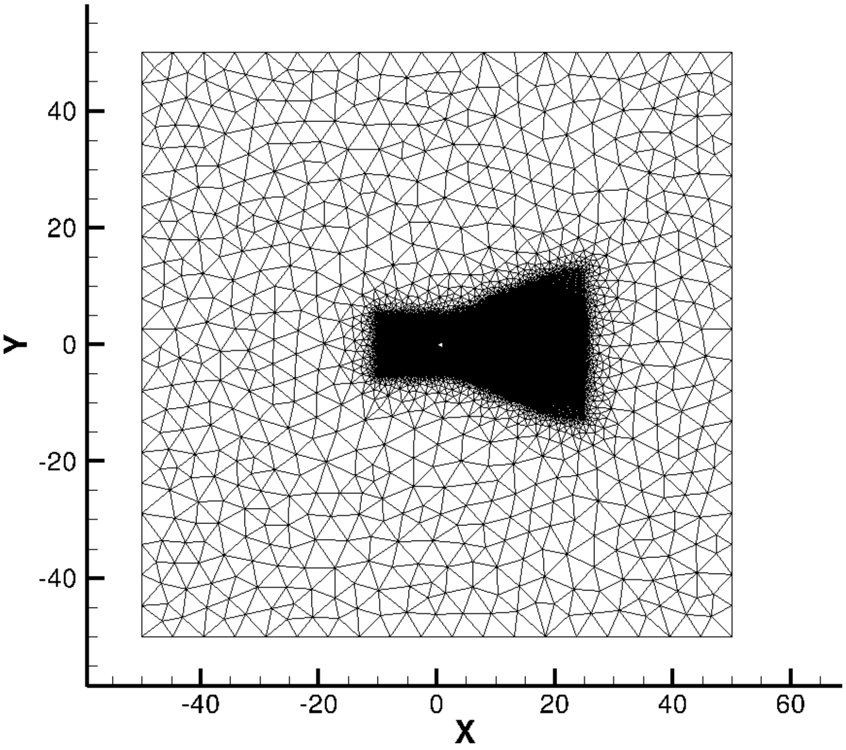


Figure 6.9 Grid for the triangular wedge

The numerical results are assessed by examining the contours of pressure for all reconstruction schemes given in Figures 6.12, 6.13, 6.14, and 6.15 for second-, pseudo third-, formal third-, and fourth-order schemes, respectively. The second-order results shows a few well resolved vortices close to the triangle, but the structure is quickly dissipated as they move away from the triangle. The pseudo third-order scheme is able to maintain some of the structure better than the second-order scheme. A somewhat surprising result is that the third-order scheme does not appear to capture the vortices any better than the pseudo third-order scheme and perhaps only slightly better than the second-order scheme. This result illustrates that a high-order scheme does not guarantee a better solution on a fixed

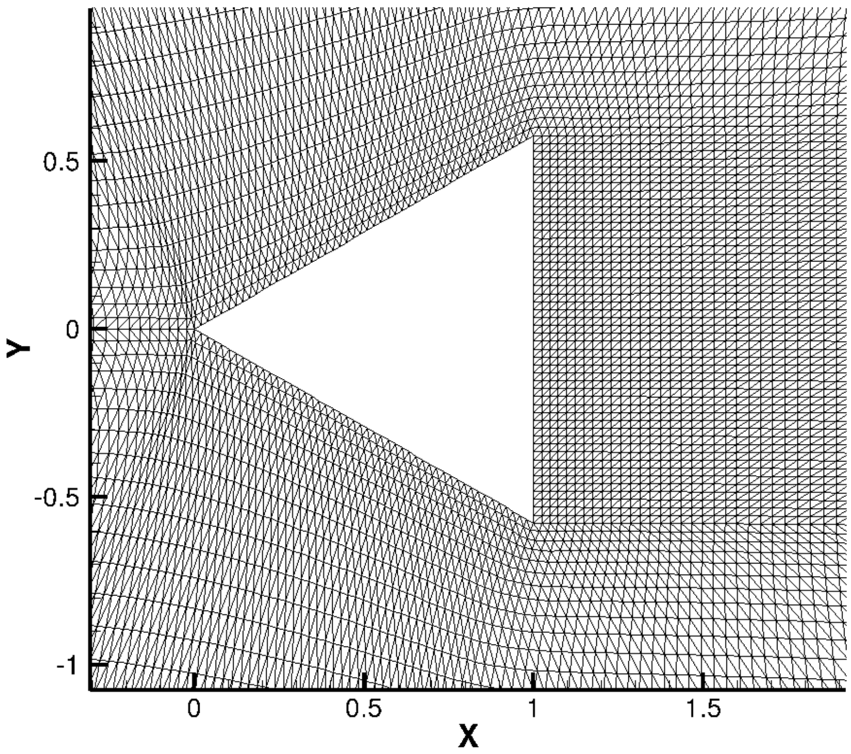


Figure 6.10 Detail of the triangular wedge grid

grid; rather, it predicts the reduction of error as the grid is refined. Fortunately, the fourth-order scheme is able to demonstrate marked improvement over the other three reconstruction schemes in maintaining the vortices.

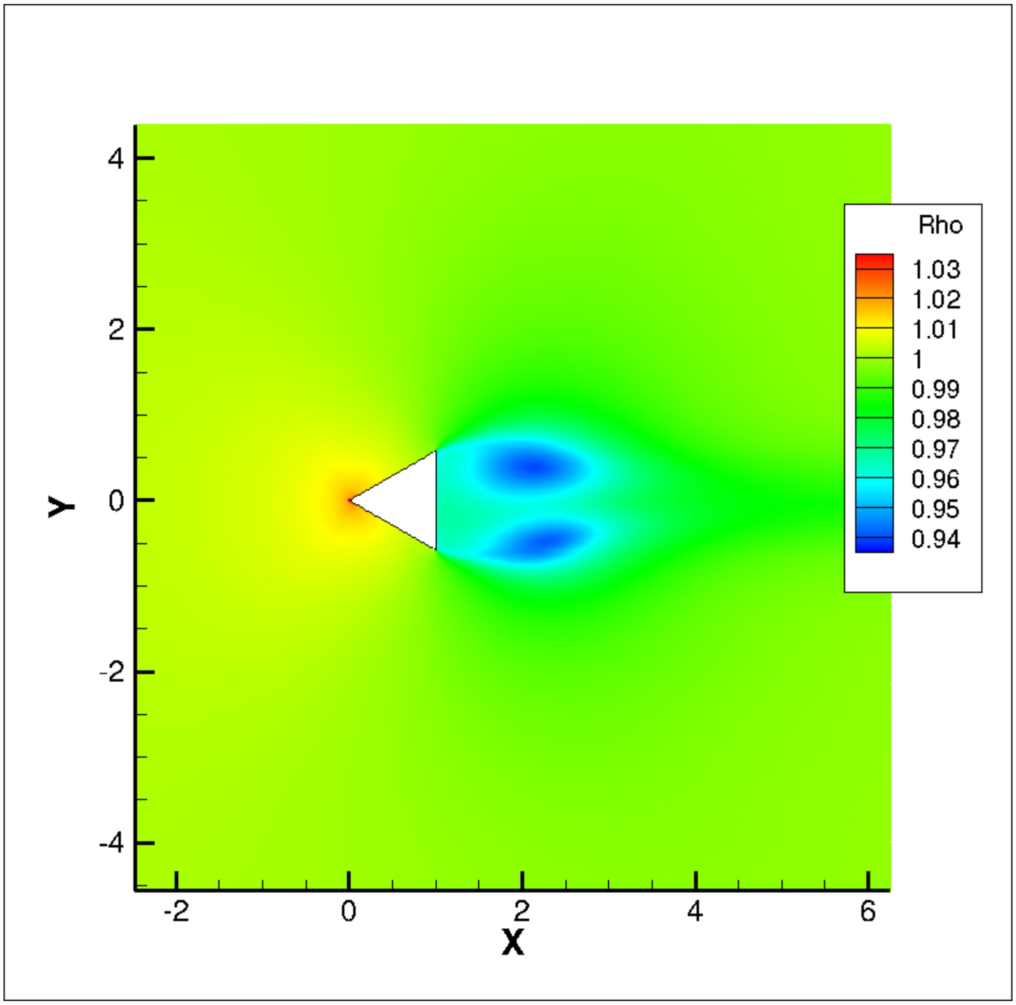


Figure 6.11 First-order solution density contours

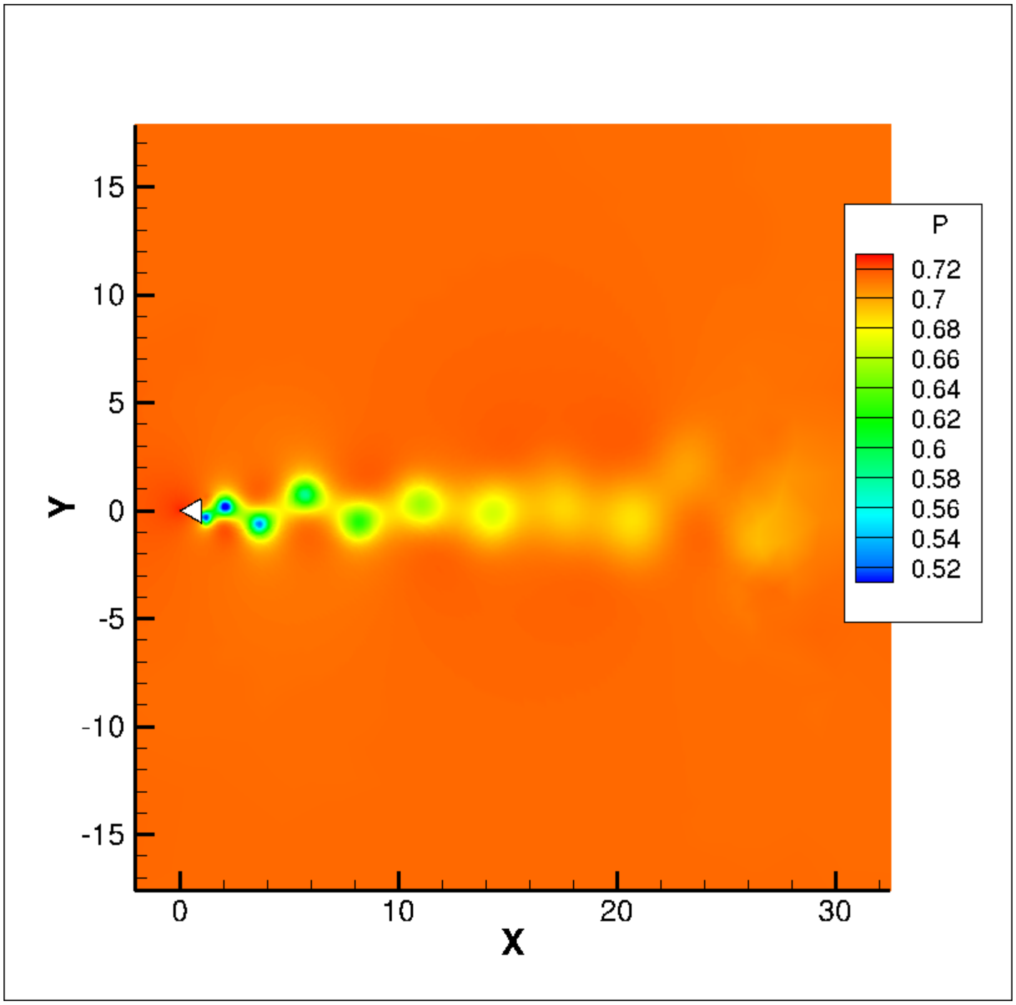


Figure 6.12 Pressure contours for the second-order solution

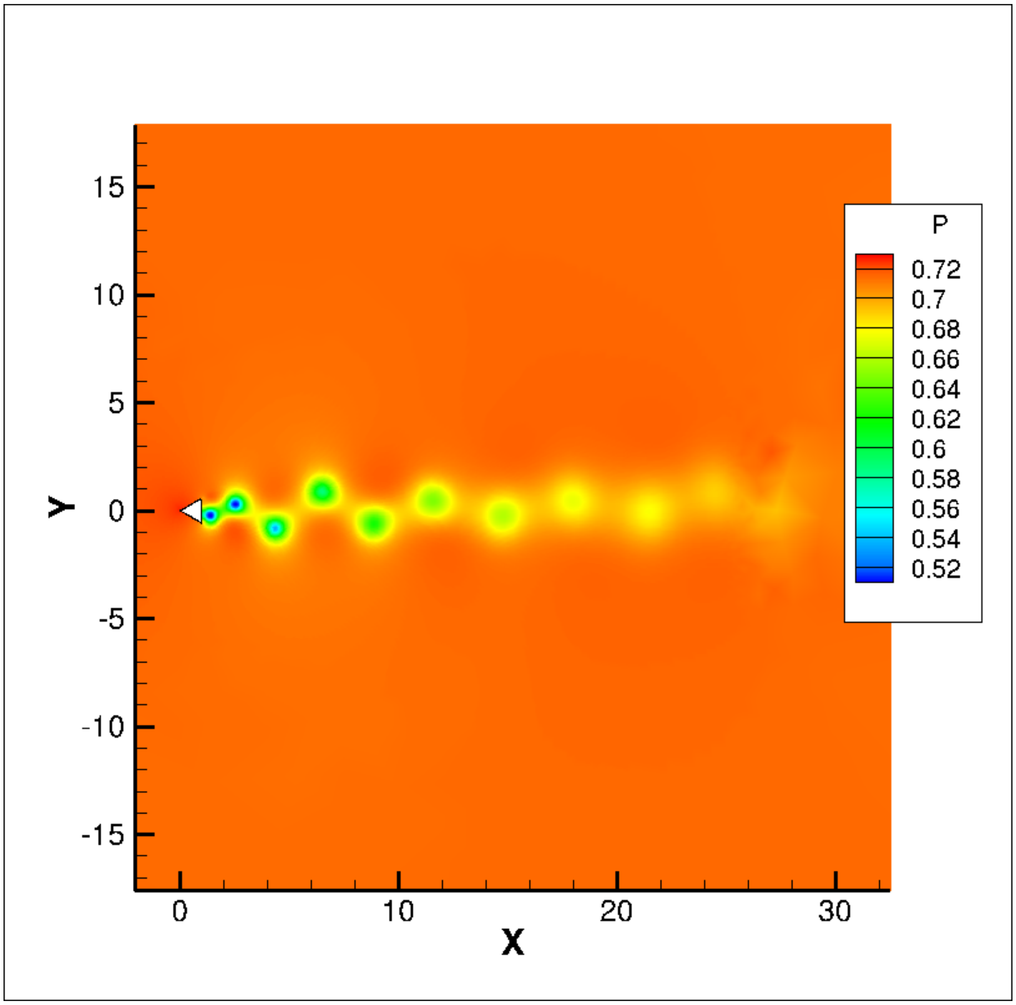


Figure 6.13 Pressure contours for the pseudo third-order solution

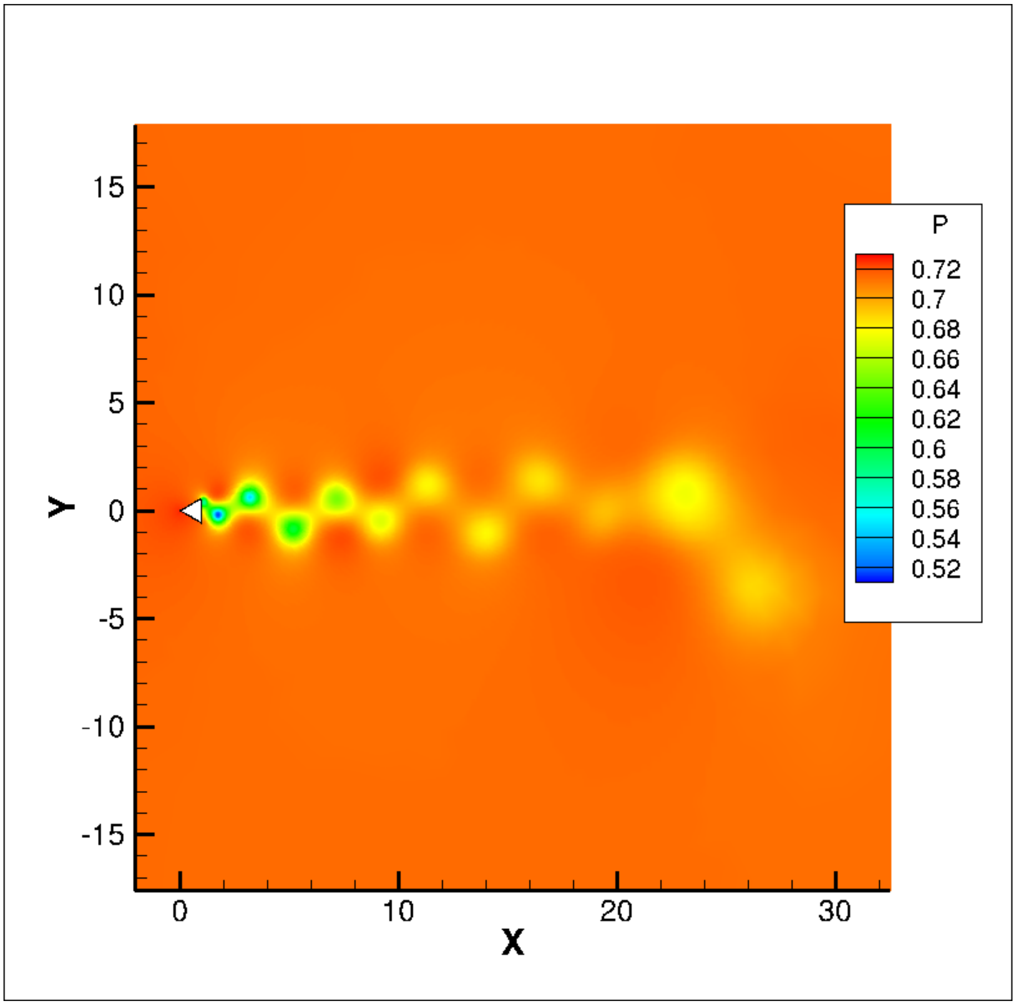


Figure 6.14 Pressure contours for the third-order solution

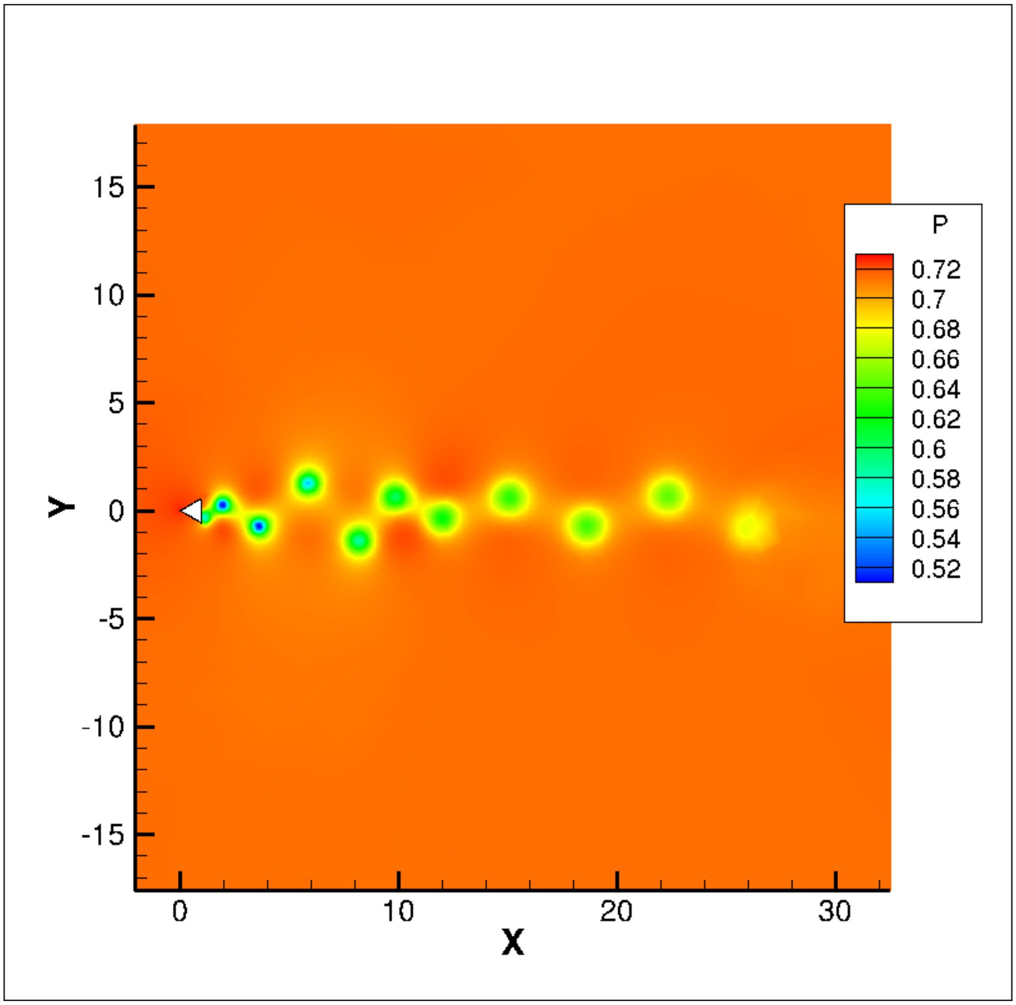


Figure 6.15 Pressure contours for the fourth-order solution

Timing results are presented in Table 6.6 to compare the efficiency of the various reconstruction schemes. The total times for all methods are comparatively high; approximately three and a half days for the second-order scheme, four and a quarter days for the pseudo third-order scheme, eight and a half days for the third-order scheme, and sixteen days for the fourth-order scheme. The long times required to run this grid is affected by the use of triangles to discretize the domain and additional Newton iterations to drive the unsteady residual to zero. Compared with quadrilateral elements, triangles generally increase the number of neighbors a node has. This increases the number of edges around a control volume, which drives up the cost of the flux residual. While it was not apparent in the vortex convection problem, the mismatch between the order of the flux on the right hand side and the first-order Jacobian system on the left hand side in Equation 2.21 penalizes the convergence of the nonlinear solver. The second-order scheme required 20 Newton iterations per time step to converge to the desired tolerance. This base number of iterations increased to 26 for pseudo third-order, 30 for third-order, and 40 for fourth-order reconstruction. The third- and fourth-order schemes further suffer from the time required to solve the least squares problem, which is partially mitigated by storing the factorization for each CV, and the application of Gaussian integration to the flux residual. The timing results also reveal a subtle detail of the lower reconstruction schemes; namely, the cost per node averaged over all iterations, temporal and Newton, is lower for the pseudo third-order scheme than for the second-order scheme. The reason for this result is that the reconstruction for the pseudo third-order scheme requires less access to data stored in arrays and fewer floating point operations to extrapolate the variables. Whereas the second-order scheme computes the quadrature node during the reconstruction, the reconstruction for the pseudo third-order scheme is partially precomputed and does not require that the quadrature node be computed.

Table 6.6 Timing results for the second-, third-, and fourth-order schemes in seconds for the triangular wedge problem

Order	Total Time (s)	Time/It./Newton It./node (s)
2^{nd}	3.0333e+05	1.9165e-05
pseudo 3^{rd}	3.6942e+05	1.7955e-05
3^{rd}	7.3572e+05	3.0989e-05
4^{th}	1.3726e+06	4.3361e-05

* Executed on an Intel® Xeon® X7560.

This test case is also the only one with timing results given for all methods on a fixed grid. This allows a more detailed examination of the incurred cost, as the reconstruction order is increased, in terms of time. In particular, the increase in time is examined for the computation of the residual and the derivative values. The time results for all methods is shown in Table 6.7.[*] The presented results have been averaged over the time step, Newton iterations, and number of nodes so that the additional cost can be compared for an average node as the order is increased. As expected, the second-order and pseudo third-order schemes have nearly identical results since they both use the same algorithm to compute derivative values and the flux residual computation is similar. There is a large increase in the time required to compute the derivatives by jumping to third-order accuracy, approximately a factor of 35 more time than the second-order schemes. This increase becomes nearly a hundredfold for the fourth-order scheme. However, the flux residual computation shows a more modest increase in required time, roughly twofold. Combining the times for both residual and derivative calculations approximately yields 9 μs for the second-order schemes, 22 μs for the third-order scheme, and 36 μs for the fourth-order scheme. Examining the times in this way implies that, for an average node, the third-order scheme is a little more

[*] Additional timing code was added after this test case was run; rather than run the full iterations again, these timing results are averaged over 1000 iterations.

than twice as expensive and the fourth-order scheme almost four times as expensive as the second-order schemes.

Table 6.7 Timing results for computation of flux residual and derivatives for all orders on the triangular wedge problem

Order	Average Derivative Time (μs)	Average Residual Time (μs)
2^{nd}	0.1690	8.8142
pseudo 3^{rd}	0.1708	8.4492
3^{rd}	6.1432	15.4497
4^{th}	19.4407	16.7242

* Executed on an Intel® Xeon® X7560.

CHAPTER 7

CONCLUSION

A high-order accurate spatial reconstruction algorithm has been examined and implemented in a two-dimensional unstructured finite volume flow solver for the Euler equations. Details of both the flow solver and reconstruction algorithm implementation are given. The order of accuracy of the reconstruction algorithm is verified for orders three and four using the Method of Manufactured Solutions. Supersonic flow in an annulus is used to show the order of accuracy in a grid convergence study. Subsonic flow over a circular cylinder is studied to demonstrate the superiority of the high-order methods at correctly resolving the pressure field along the back side of the cylinder. This test case also showed the potential for a hybrid solution where different orders of reconstruction are applied to different regions of the computational domain. Simulations were also presented for different flow conditions about the NACA 0012 airfoil geometry which demonstrate the effects of the slope limiter. This also included a discussion on the novel way in which quadrature points are correctly positioned along a curved boundary that has an analytical description.

The high-order reconstruction algorithm was further extended to time dependent simulations. The accuracy of the solver was verified by a grid convergence study on the problem of passive vortex convection. Computational results verified that the high-order reconstruction algorithm, particularly fourth-order, was able to successfully produce similar results to the second-order scheme on the finest grid using a grid with about one fourth the number of points. The results from the simulation of vortex shedding due to the presence of a triangular wedge illustrated that the fourth-order algorithm was able to capture the structure of the vortices better than the lower order algorithms.

Future Work

There are many areas in which further research may be investigated. The flow solver developed in this research can be extended to solve the full Navier-Stokes equations with the inclusion of the viscous contributions as was done by Michalak [27]. The solver could also be parallelized using message passing and domain decomposition techniques. This would allow faster simulation of unsteady cases, in particular.

There is also the extension of this reconstruction algorithm to Tenasi, a three-dimensional unstructured finite volume flow solver, to consider. This appears to be a challenging endeavor as much of Tenasi is edge-based (element edges), whereas the studied algorithm needs to operate on the facets generated by the median dual procedure. To mitigate some of the increased computational cost at the expense of formally preserving the order of accuracy, the solver might be extended using the idea of hybrid reconstruction, as discussed in Chapter 5. Rather than running the full simulation as higher order, regions of the computational domain might be tagged to run high-order reconstruction while the rest of the domain, in particular the boundary (to avoid complications of modeling curved boundaries in three dimensions), is run at a lower order. The tagged higher order regions will incur more computations as all integral must be computed with a high-order quadrature. The domain decomposition algorithm that partitions the full grid into smaller pieces must take the additional computations on tagged nodes into consideration to ensure proper load balancing. Fortunately, such algorithms exist in the literature, particularly the algorithm described by Karypis and Kumar [41].

A suitable use of such high-order regions in the domain would be to capture vortices that are shed from the modeled geometry. This is an idea that was explored by Hariharan [42] and further by Hariharan and Sankar [43]. The research documented in these papers examined the idea of using a hybrid fifth-/seventh-order ENO scheme for structured grids to resolve vortices generated by a wing tip. The approach the researches took was to utilize the overset grid methodology to embed a refinement region, where the solution is computed to seventh-order accuracy, in the full domain grid, which is limited to fifth-order accuracy. The researches noted significant improvement in resolving and maintaining critical parameters characterizing the structure of the vortices as compared to using only the fifth-order accurate scheme. These positive results help to validate the idea that further research into mixed order, hybrid schemes can yield solutions with improved resolution of essential flow features.

REFERENCES

[1] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R., "Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III," *Journal of Computational Physics*, Vol. 71, No. 2, 1987, pp. 231 – 303. 1

[2] Liu, X.-D., Osher, S., and Chan, T., "Weighted Essentially Non-oscillatory Schemes," *Journal of Computational Physics*, Vol. 115, No. 1, 1994, pp. 200 – 212. 1

[3] Reed, W. H. and Hill, T. R., "Triangular Mesh Methods for the Neutron Transport Equation," Tech. Rep. LA-UR-73-479, Los Alamos Scientic Laboratory, 1973. 2

[4] Berggren, M., "A Vertex-Centered, Dual Discontinous Galerkin Method," *Journal of Computational and Applied Mathematics*, Vol. 192, No. 1, 2006, pp. 175 – 181. 2

[5] Berggren, M., Ekström, S.-E., and Nordström, J., "A Discontinous Galerkin Extension of the Vertex-Centered Edge-Based Finite Volume Method," *Communications in Computational Physics*, Vol. 5, No. 2 - 4, Februrary 2009, pp. 456 – 468. 2

[6] Barth, T. J. and Frederickson, P. O., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," *28th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1990, Paper No. AIAA 90-0013. 2, 3, 25

[7] Harten, A. and Chakravarthy, S. R., "Multi-Dimensional ENO Schemes for General Geometries," Tech. Rep. NASA Contractor Report 187637 ICASE Report No. 91-76, ICASE-NASA Langley, ICASE, NASA Langley Research Center, Hampton VA 23665-5225, September 1991. 2, 3, 18

[8] Abgrall, R., "On Essentially Non-Oscillatory Schemes on Unstructured Meshes: Analysis and Implementation," Tech. Rep. NASA Contractor Report 191415 ICASE Report No. 92-74, ICASE-NASA Langley, ICASE, NASA Langley Research Center, Hampton VA 23681-0001, December 1992. 3

[9] Friedrich, O., "Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids," *Journal of Computational Physics*, Vol. 144, 1998, pp. 194–212. 3

[10] Hu, C. and Shu, C.-W., "Weighted Essentially Non-oscillatory Schemes on Triangular Meshes," *Journal of Computational Physics*, Vol. 150, 1999, pp. 97–127. 3

[11] Ollivier-Gooch, C. F., “High-Order ENO Schemes for Unstructured Meshes Based on Least-Squares Reconstruction,” *35th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1997, Paper No. AIAA 1997-0540. 4

[12] Ollivier-Gooch, C., “Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction,” *Journal of Computational Physics*, Vol. 133, May 1997, pp. 6 – 17. 4

[13] Ollivier-Gooch, C. and Altena, M. V., “A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation,” *Journal of Computational Physics*, Vol. 181, September 2002, pp. 729 – 752. 4, 25, 34

[14] Nejat, A., Ollivier-Gooch, C., and Michalak, K., “Accuracy Assessment Methodology for a High-Order Unstructured Finite Volume Solver,” *18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, 2007, Paper No. AIAA 2007-4194. 4, 25, 86

[15] Ollivier-Gooch, C., Nejat, A., and Michalak, K., “On Obtaining High-Order Finite-Volume Solutions to the Euler Equations on Unstructured Meshes,” *18th AIAA Computational Dynamics Conference*, Miami, FL, 2007, Paper No. AIAA 2007-4464. 4, 25

[16] Michalak, C. and Ollivier-Gooch, C., “Unstructured High-Order Accurate Finite-Volume Solutions of the Navier-Stokes Equations,” *47th Aerospace Sciences Meeting and Exhibit*, Orlando, FL, 2009, Paper No. AIAA 2009-0954. 4

[17] Michalak, K. and Ollivier-Gooch, C., “Limiters for Unstructured Higher-Order Accurate Solutions of the Euler Equations,” *46th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2008, Paper No. AIAA 2008-0776. 4, 49, 51, 123

[18] Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357 – 372. 11

[19] Hirsch, C., *Numerical Computation of Internal and External Flows, Volume 2*, John Wiley and Sons, 1990. 11

[20] Whitfield, D. L. and Janus, J. M., “Three-Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting,” *AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, Snowmass, CO, 1984, Paper No. AIAA 84-1552. 14

[21] Anderson, W. K. and Bonhaus, D. L., “An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids,” *Computers & Fluids*, Vol. 23, No. 1, 1994, pp. 1 – 21. 17

[22] Hyams, D. G., *An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies*, Ph.D. thesis, Mississippi State University, 2003. 17, 38

[23] Suresh, A. and Jorgenson, P. C. E., "Essentially Nonoscillatory (ENO) Reconstruction via Extrapolation," *33rd Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1995, Paper No. AIAA 95-0467. 25

[24] Haselbacher, A., "A WENO Reconstruction Algorithm for Unstructured Grids Based on Explicit Stencil Construction," *43rd Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2005, Paper No. AIAA 2005-0879. 25

[25] Ollivier-Gooch, C., Nejat, A., and Michalak, K., "Obtaining and Verifying High-Order Unstructured Finite Volume Solutions to the Euler Equations," *AIAA Journal*, Vol. 47, No. 9, September 2009, pp. 2105–2120. 38, 77, 86

[26] Demmel, J. W., *Applied Numerical Linear Algebra*, SIAM, 1997. 44, 46

[27] Michalak, C., *Efficient High-Order Accurate Unstructured Finite-Volume Algorithms for Viscous and Inviscid Compressible Flows*, Ph.D. thesis, The University of British Columbia, April 2009. 46, 155

[28] Nejat, A., *A Higher-Order Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows*, Ph.D. thesis, The University of British Columbia, 2007. 49, 50, 51, 52, 86, 105, 123

[29] Barth, T. J. and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," *27th Aerospace Sciences Meeting*, Reno, Nevada, 1989, Paper No. AIAA 89-0366. 49

[30] Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *Journal of Computational Physics*, Vol. 118, 1995, pp. 120–130. 50

[31] Salari, K. and Knupp, P., "Code Verification by the Method of Manufactured Solutions," Sandia Report SAND2000-1444, Sandia National Laboratories, Sandia National Laboratories Albuquerque, New Mexico 87185, June 2000. 53

[32] Murali, V. K., *Code Verification Using the Method of Manufactured Solutions*, Master's thesis, Mississippi State University, December 2002. 53, 54

[33] Burg, C. O. E. and Murali, V. K., "Efficient Code Verification Using the Residual Formulation of the Method of Manufactured Solutions," *34th AIAA Fluid Dynamics Conference and Exhibit*, Portland, Oregon, June 2004, Paper No. AIAA 2004-2628. 53, 54

[34] Erwin, T., Anderson, W. K., Kapadia, S., and Wang, L., "Three Dimensional Stabilized Finite Elements for Compressible Navier-Stokes," *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, 2011, Paper No. AIAA 2011-3411. 54

[35] Johnson, C., *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Dover, 2009. 69

[36] Aftosmis, M., Gaitonde, D., and Tavares, T. S., "Behavior of Linear Reconstruction Techniques on Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 11, November 1995, pp. 2038–2049. 77, 78

[37] Diskin, B. and Thomas, J. L., "Accuracy Analysis for Mixed-Element Finite-Volume Disctretization Schemes," Tech. Rep. NIA Report No. 2007-08, National Institute of Aerospace, 2007. 81

[38] Wang, Z. J. and Sun, Y., "A Curvature-Based Wall Boundary Condition for the Euler Equations on Unstructured Grids," *40th AIAA Aerospace Sciences Meeting and Exhibit*, January 2002, Paper No. AIAA 2002-0966. 110

[39] Yee, H. C., Sandham, N. D., and Djomehri, M. J., "Low-Dissipative High-Order Shock-Capturing Methods Using Charateristic-Based Filters," *Journal of Computational Physics*, Vol. 150, 1999, pp. 199–238. 129

[40] Wang, L., *Techniques for High-order Adaptive Discontinuous Galerkin Disctretizations in Fluid Dynamics*, Ph.D. thesis, The University of Wyoming, April 2009. 143

[41] Karypis, G. and Kumar, V., "Parallel Multilevel k-Way Partitioning Scheme for Irregular Graphs," *SIAM Review*, Vol. 41, No. 2, June 1999, pp. 278–300. 155

[42] Hariharan, N., "Rotrary-Wing Capturing: High-Order Schemes Toward Minimizing Numerical Vortex Dissipation," *Journal of Aircraft*, Vol. 39, No. 5, September-October 2002, pp. 822–829. 155

[43] Hariharan, N. and Sankar, L., "Uniformly High-Order Essentially Nonoscillatory Schemes for Vortex Convection Across Overset Interfaces," *Journal of Aircraft*, Vol. 45, No. 2, March-April 2008, pp. 546–559. 155

VITA

Shane Edmond Sawyer was born in Chattanooga, Tennessee in June of 1980. After graduating from East Ridge High School, he went on to complete a Bachelor's of Science degree in Applied Mathematics, graduating in May, 2006 magna cum laude. He then accepted a position at the UT SimCenter as a graduate research assistant. Shane plans on continuing his education by pursuing a Doctorate in Applied Mathematics with an emphasis on computational methods.