2008

An adaptive quadrature-free implementation of the high-order spectral volume method on unstructured grids

Robert Evan Harris
Iowa State University

Recommended Citation

Harris, Robert Evan, "An adaptive quadrature-free implementation of the high-order spectral volume method on unstructured grids" (2008). *Retrospective Theses and Dissertations*. 15691.
https://lib.dr.iastate.edu/rtd/15691

An adaptive quadrature-free implementation of the high-order spectral volume method on unstructured grids

by

Robert Evan Harris

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Program of Study Committee:
Z. J. Wang, Major Professor
Tom I-P Shih
Ambar Mitra
Thomas Rudolphi
Richard Pletcher

Iowa State University

Ames, Iowa

2008

DEDICATION

I would like to dedicate this thesis to my wonderful wife Robin and my family without whose love and support I would not have been able to complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my adviser, Dr. Z. J. Wang. His continuous stream of advice, his ability to communicate complex ideas in elegant and easily understandable ways, his willingness to let me explore my own ideas, and most of all his enthusiasm and optimism have been essential to the completion of this work. I am eternally grateful to Z. J. for helping me realize that this is what I want to do with my life. Many thanks are due to Dr. Tom I-P Shih, Dr. Ambar Mitra, Dr. Tom Rudolphi, and Dr. Richard Pletcher for serving on my doctoral committee. Special thanks are due to the Air Force and the Department of Energy. This study has been supported by the Air Force Office of Scientific Research (AFOSR) grant FA9550-06-1-0146, and the Department of Energy (DOE) grant DE-FG02-05ER25677.

Thanks go to my office mate Ravishekar Kannan for providing me with a much needed sounding board for new ideas. Special thanks go to Dr. Ashvin Mahajan for being the sole operator of the Underground Linux Group (ULG) at Iowa State, and for providing essential after hours computing support. Additional thanks go to my colleagues Brandon Williams and Jason Busch for appreciating my sense of humor, and for providing a much needed distraction from the rigor of day-to-day code development, and to Brandon for providing a formidable ping-pong opponent.

Last, but certainly not least, I thank my loving wife Robin for understanding the importance of this endeavor and for never ceasing to believe in me.

ABSTRACT

An efficient implementation of the high-order spectral volume (SV) method is presented for multi-dimensional conservation laws on unstructured grids. In the SV method, each simplex cell is called a spectral volume (SV), and the SV is further subdivided into polygonal (2D), or polyhedral (3D) control volumes (CVs) to support high-order data reconstructions. In the traditional implementation, Gauss quadrature formulas are used to approximate the flux integrals on all faces. In the new approach, a nodal set is selected and used to reconstruct a high-order polynomial approximation for the flux vector, and then the flux integrals on the internal faces are computed analytically, without the need for Gauss quadrature formulas. This gives a significant advantage over the traditional SV method in efficiency and ease of implementation. Fundamental properties of the new SV implementation are studied and high-order accuracy is demonstrated for linear and nonlinear advection equations, and the Euler equations.

The new quadrature-free approach is then extended to handle local adaptive hp-refinement (grid and order refinement). Efficient edge-based adaptation utilizing a binary tree search algorithm is employed. Several different adaptation criteria which focus computational effort near high gradient regions are presented. Both h- and p- refinements are presented in a general framework where it is possible to perform either or both on any grid cell at any time. Several well-known inviscid flow test cases, subjected to various levels of adaptation, are utilized to demonstrate the effectiveness of the method.

An analysis of the accuracy and stability properties of the spectral volume (SV) method is then presented. In general, the partitioning of an SV into CVs is not uniquely defined. It is of great importance to select a partition which yields favorable stability properties, and results in an interpolation polynomial of high quality. The current work seeks to address the issue of stability, as well as polynomial quality, in the design of SV partitions. A new approach is presented, which efficiently locates stable partitions by means of constrained minimization. This is motivated by the fact that, at present, an exhaustive search approach to SV partition design would be prohibitively costly and thus not feasible. Once stable partitions are located, a high quality interpolation polynomial is then assured by subsequently minimizing the dissipation and dispersion errors of the stable partitions. Preliminary results are given which indicate this to be an effective method for use in the design of stable and highly accurate SV partitions of arbitrary order.

CHAPTER 1. INTRODUCTION

1.1 Overview of computational fluid dynamics (CFD)

Computational fluid dynamics (CFD) is a branch of fluid dynamics that focuses on solving the governing equations of fluid flow using digital computers. Although many of the basic ideas behind CFD are very old, the modern development of CFD could not begin until the early 1950's with the advent of the digital computer. Problems that once required years of tedious hand calculations, could be solved in only seconds on a digital computer.

One of the fundamental issues arising in the numerical solution of the governing equations of fluid dynamics is that fluid is a continuous medium which must be represented in a discrete manner to be understood by a digital computer. The process of converting continuous models and equations into a discrete form is referred to as discretization, and is essential in CFD. Selection of a particular discretization technique is heavily dependent on the type of problem and solution being sought. Since the advent of CFD, the primary focus has been on three different discretization techniques: the finite difference (FD), finite element (FE), and finite volume (FV) methods. These methods are outlined in Section 1.1.1.

1.1.1 Some history

In the early twentieth century, the numerical methods which would become a basis for one of the most common discretization techniques, the finite difference (FD) method, were maturing. In the FD method, the domain of interest is discretized into a finite number of mesh or grid points and the equations are cast in their differential form. Taylor series expansions are then used to convert the governing partial differential equations (PDEs) into algebraic equations. Taylor series expansions can be carried out in many different ways which depend on the type of derivatives appearing in the governing equations, and on the desired accuracy of the approximation. For example, the equations could be derived for a 5-point stencil, or a 9-point stencil as shown in Figure 1. If the 5-point stencil is used, Taylor series expansions are taken about the grid point *i,j* and derivatives are then expressed as algebraic differences of the flowfield quantities evaluated at the grid points. If high-order accuracy is sought, more points could be added to the stencil in order to retain more terms of the Taylor series, or the differencing could be done in a way that is compact [37], so as to avoid using a large stencil. The solution quantities to be updated in the FD method are the values of the flow variables at each grid point in the problem domain.

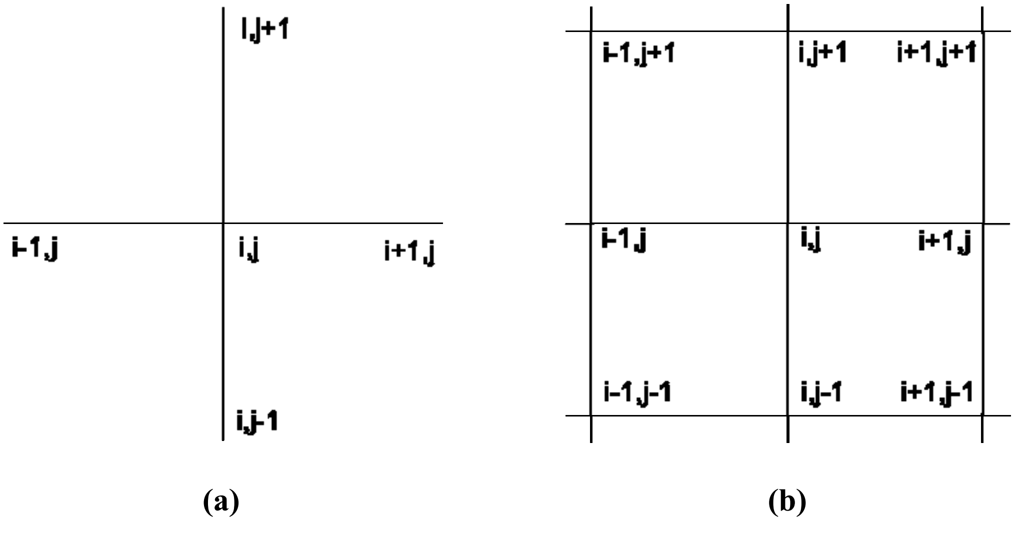


Figure 1. Possible stencils for use with the finite difference method; (a) 5-point; (b) 9-point.

Another common discretization technique which emerged in the early 1940’s is the finite element (FE) method. The FE method was originally developed for solving problems in elasticity, and structural analysis. In the FE method, the problem domain is subdivided into a number of smaller elements, and the governing equations are cast in their integral form. Basis functions are then selected, as defined by nodes on each element, and used to support a piecewise-continuous functional representation of the solution on the domain. These traditional FE methods are not well suited for general fluid flow problems. Because of their continuous nature, they are very well suited for structural analysis problems which mostly deal with solution quantities which are continuous. More recently, a new FE method which is ideal for fluid flow problems, called the discontinuous Galerkin (DG) method [10-13], has been under development. The DG method will be subsequently described in Section 1.4.

The final discretization method to be addressed here is the finite volume (FV) method. Like the FE method, in the FV method the equations are cast in their integral form, and the problem domain is subdivided into a number of smaller elements called volumes. The main difference between the FE and FV methods is that in the FE method, there is a unique set of nodes for which the solution is updated, while in the FV method, the solution quantities are volume-averages which vary from cell to cell. Also of significance is that in the FV method, there is no inherent continuity of the solution from one cell to another and it is thus equipped to handle problems which involve physical discontinuities of the flow variables in the problem domain. When discontinuities are present in the flow, a unique flux between cells does not exist. This problem is alleviated by using a Riemann flux to add the necessary upwinding and produce a unique flux across cell interfaces. The method outlined in the subsequent chapters borrows ideas from the FE and FV methods.

There are many relevant texts which were instrumental in the early years of CFD, and continue to be important references both for fundamental principles and for relevant historical information. In particular, the texts by Tannehill et al. [40], Anderson [2], and Hirsch [28,29] are very commonly referenced.

1.1.2 Current state-of-the-art

In recent years, CFD has become an indispensible tool in the design and analysis of vehicles which undergo motion in an atmosphere. Although there have been tremendous advances in digital computing which have permitted previously unrealizable calculations to be carried out, the speed and memory capacity of the digital computer continues to be the major limiting factor in CFD. Currently the direct numerical simulation (DNS) of turbulent flows for practical problems of interest, such as the three dimensional flow over an aircraft or spacecraft, is intractable. The numerical methods and understanding of the governing equations is mature, but resolution of the disparate length scales present in turbulent flows necessitates an extremely refined calculation which, at current, is not feasible. For this reason, much attention has gone into the development of different modeling techniques which avoid a direct solution of the Navier-Stokes equations. For example, The Reynolds-Averaged Navier-Stokes (RANS) equations aim to provide information of the flowfield in an averaged sense, while Large Eddy Simulation (LES) attempts to resolve the large scales while filtering out the smaller scales of the flow. While historically the aforementioned methods have been of low-order accuracy, and primarily used on structured grids, many researchers are extending these methods for use with high-order numerical schemes on unstructured grids. This is done because unstructured grids are able to handle more complex geometries, and high-order methods are capable of increased accuracy using far fewer grid cells. These techniques are explained in Sections 1.2 and 1.3, respectively.

1.2 Structured vs. unstructured grids in CFD

There are two major types of grids used in CFD analyses. These are referred to as either structured grids or unstructured grids. Structured grids take their name from the fact that they possess a discernible pattern which is characterized by an implicit connectivity between cells, or nodes. Conversely, unstructured grids may contain no discernible pattern and thus no implied connectivity exists between constituents.

Structured grid methods typically utilize quadrilateral elements in 2D, and hexahedral elements in 3D. These methods are known to offer better resolution of the boundary layer for viscous flow problems, because most grid skewness problems can be avoided. It is however very difficult to generate a good structured grid for problems involving complicated geometries in which the resolution of extremely fine details is critical, such as rough surfaces or ice shapes for example. In addition, the creation of a high-quality structured grid can take weeks or even months to achieve by a skilled engineer.

Unstructured grid methods typically utilize triangular elements in 2D, and tetrahedral elements in 3D, while many other cell types such as hexahedra, pyramids, and prisms are also popular. These methods are known to be very robust, and are capable of resolving extremely complicated geometries with very fine details. There are some circumstances where care must be taken to prevent an overly skewed grid, such as in the vicinity of a boundary layer or near a region of high curvature in the geometry. Unlike structured grids, the creation of a good unstructured grid can take only hours or even minutes to achieve, and can often be automated.

Many problems have some features which would best be handled using structured grids, and other features which demand unstructured grids. For this reason, hybrid grid methods, which seek to combine the strengths of both structured and unstructured grids, are becoming very common. Typically hybrid grids employ a structured grid in localized areas, while using an unstructured grid for the majority of the problem domain. These grids can be very difficult to generate, and often require significant time and expertise.

The appropriate choice of grid is heavily dependent on the type of problem under consideration, and the capabilities of the flow solver being used. Although each approach has its own strengths and weaknesses, all of the above approaches continue to see widespread used in CFD. This thesis concerns the development of unstructured grid CFD methods.

1.3 High-order vs. low-order methods

For a given problem, the difference between the governing partial differential equations (PDEs) and the discretized algebraic equations is referred to as the truncation error of the discretization. In CFD, this truncation error is proportional to the maximum grid spacing h, raised to some exponent p. The value of the exponent p is known as the order of the discretization. When the value of p is 2 or less, the scheme is referred to as being low-order accurate, while for values of p greater than 2, the scheme is high-order accurate. Although high-order schemes have been gaining popularity, the state-of-the-art commercial CFD codes are still predominantly low-order. Low-order methods have been extremely successful in engineering, and are sufficient for a wide range of applications given an adequately refined grid. Never-the-less, there are many problems of increasing interest which necessitate high-order methods on unstructured grids: Some examples include vortex-dominated flows, rotorcraft flow/blade-vortex interaction problems, aero-acoustic noise predictions, and LES/DNS for complex configurations to name just a few. Many of these problems involve vortex structures which must be captured and preserved in order to represent the correct aerodynamic behavior. In problems like this, low-order methods tend to over-dissipate the vortices, resulting in a loss of important flow physics. On the other hand, high-order methods have very low numerical dissipation and are thus exceptionally well suited for the capture and preservation of vortices and other detailed flow features. Here we improve and expand upon an existing high-order accurate unstructured grid method.

1.4 Review of popular CFD methods

The spectral volume (SV) method is a recently developed finite volume method for hyperbolic conservation laws on unstructured grids [50-54,32,39]. The SV method belongs to a general class of Godunov-type finite volume method [17,46], which has been under development for several decades, and is considered to be the current state-of-the-art for the numerical solution of hyperbolic conservation laws. For a more detailed review of the literature on the Godunov-type method, refer to [47,50,49,15], and the references therein. Many of the most popular numerical methods, such as the k-exact finite volume [4,14], the essentially non-oscillatory (ENO) [24,1], and weighted ENO [30,57] methods are also Godunov-type methods. The SV method is also closely related to the discontinuous Galerkin

(DG) [10-13,3,5,57] method, a popular finite-element method for conservation laws, and the spectral difference [38] method. Both the SV and DG methods employ multiple degrees of freedom within a single element. Each simplex in the SV method utilizes a "structured" set of sub-cells, thus resulting in more cell faces where Riemann problems must be solved. This inherent property of sub-cell resolution may give the SV method the ability to capture discontinuities with higher resolution than the DG method. The number of interior faces in 3D may be high, making the traditional SV method expensive in 3D. However, the methodology outlined in Section 3 dramatically reduces the cost of the 3D SV method. For more comparisons of the SV and DG methods, refer to [49,50,60].

1.5 Motivation, objectives, and accomplishments of current work

The additional cell faces present in the SV method, mentioned in Section 1.4, number in the dozens per SV for 2D and in the hundreds per SV for 3D. In the traditional implementation of the SV method, all face integrals are computed by means of Gauss quadrature formulas, which are appropriate to the shape and dimension of the face. For example, a pentagonal face existing in the partition of a tetrahedral SV (3D) is split into 3 triangles. To carry out the integration, a Gauss quadrature formula of appropriate precision is then employed for each triangle. This procedure can be efficient in 2D, where the required Gauss quadrature points number in the dozens per SV. However in 3D, the partition of a tetrahedron can be so complicated that hundreds or thousands of Gauss quadrature points per SV may be necessary to compute the face integrals to the desired precision. Therefore, the first objective of this study is to increase the efficiency of the SV method by avoiding the use of Gauss quadrature formulae in the computation of flux integrals. This will be referred to as the quadrature-free (QF) implementation of the SV method. Additionally, we will evaluate the new QF approach for the solution of both linear and nonlinear advection equations and the Euler equations in 2D and 3D.

Although the new QF approach is much more efficient than the traditional approach, additional work still needs to be done to allow the solution of practical problems within a reasonable amount of time. To this end, the second main objective of this study is to extend the new QF approach to handle solution-based grid and polynomial adaptation, or local adaptive hp-refinement. We thus aspire to solve problems with complicated flow features while avoiding a global grid refinement study, in favor of adaptive local refinement techniques, to achieve grid independent solutions. We will develop and evaluate this technology in 2D and employ several well known inviscid flow test cases for validation.

Finally, the data reconstruction in the SV method, and the resulting stability properties and polynomial quality for the scheme, is heavily dependent on the partitioning of the SV into CVs. Although some recent work has been done to develop new SV partitions with improved stability and accuracy properties [43-45], we feel that this study is very much incomplete and the ideal SV partitions have yet to be presented. For these reasons, we will develop a new methodology for determining stable and accurate SV partitions using ideas from constrained minimization. Preliminary results will be presented which show great potential of the method for use in the development of stable and highly accurate SV partitions.

The accomplishments of this work include the formulation, development, and application of an efficient new quadrature-free implementation of the high-order spectral volume method in both 2D and 3D. This objective also included the development of data limiters for problems involving discontinuities, and curved boundary treatments for the high-order implementation of slip wall boundary conditions. The second major accomplishment of this work involved the formulation, development, and application of a solver for the quadrature-free spectral volume method which utilizes solution-based grid adaptation to efficiently resolve important flow features. This included the implementation of both h- and p-refinement techniques in a unified formulation where it is possible to perform either or both on any grid cell at any time. The final accomplishment of this work is the formulation and development of a new method for obtaining stable and highly accurate reconstruction stencils for the spectral volume method. This new approach utilizes constrained minimization techniques to locate stable reconstructions with very low dissipation and dispersion errors.

1.6 Outline of dissertation

This thesis is organized as follows. In Chapter 2, we provide the basic framework for the SV method. After that, an efficient new quadrature-free implementation is described in detail in Chapter 3, including data limiting procedures and curved boundary treatment. Some results from accuracy studies and several inviscid flow test cases are also given in Chapter 3. In Chapter 4, the methodology for local adaptive hp-refinement is given, with a study of different adaptation criteria, and relevant results including several supersonic cases with strong shock waves. Chapter 5 outlines the methodology for analyzing the stability properties of the SV method, including partition generation and optimization via constrained minimization, as well as some preliminary results. Finally, a summary and some possibilities for future work are discussed in Chapter 6.

CHAPTER 2. FRAMEWORK OF SPECTRAL VOLUME (SV) METHOD

2.1 Overview of spatial and temporal discretization

Consider the multidimensional conservation law

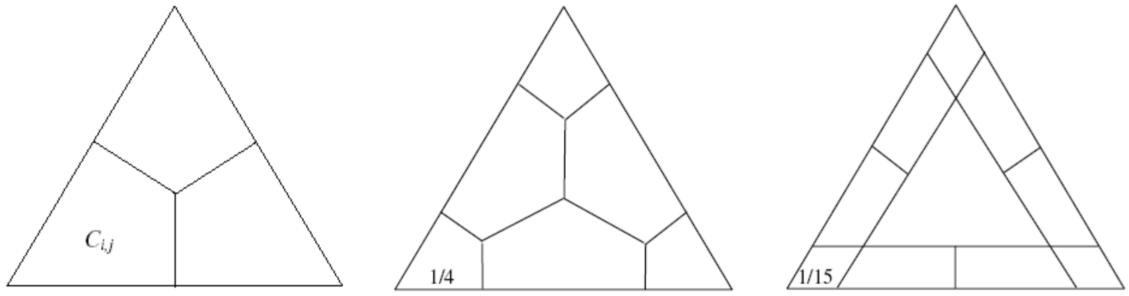
$$\frac{\partial Q}{\partial t}+\frac{\partial f(Q)}{\partial x}+\frac{\partial g(Q)}{\partial y}+\frac{\partial h(Q)}{\partial z}=0, \tag{2.1a}$$

on domain $\Omega \times [0,T]$ and $\Omega \subset R^3$ with the initial condition

$$Q(x,y,z,0)=Q_0(x,y,z), \tag{2.1b}$$

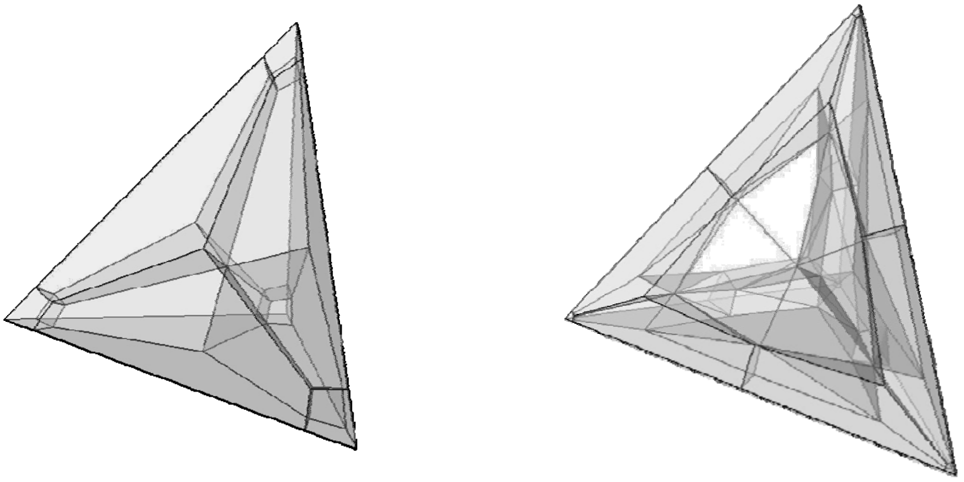
and appropriate boundary conditions on $\partial\Omega$. In (2.1), x, y, and z are the Cartesian coordinates and $(x,y,z) \in \Omega$, $t \in [0,T]$ denotes time, Q is the vector of conserved variables, and f, g and h are the fluxes in the x, y and z directions, respectively. Domain Ω is discretized into I non-overlapping triangular (2D), or tetrahedral (3D) cells. In the SV method, the simplex grid cells are called SVs, denoted S_i, which are further partitioned into CVs, denoted $C_{i,j}$, which depend on the degree of the polynomial reconstruction. Examples of partitions supporting linear, quadratic and cubic reconstructions are shown in Figure 2 (2D), and quadratic and cubic reconstructions are shown in Figure 3 (3D). The partitions shown in Figure 3 were taken from Chen [9]. Volume-averaged conserved variables on the CVs are then used to reconstruct a high-order polynomial inside the SV. To represent the solution as a polynomial of degree m, we need N pieces of independent information, or degrees of freedom (DOFs), calculated as follows:

$$N=\frac{(m+1)(m+2)\cdots(m+d)}{d!}, \tag{2.2}$$



(a) (b) (c)

Figure 2. Partitions of a triangular SV supporting linear, quadratic and cubic data reconstructions, shown in (a) , (b) and (c), respectively.



(a) (b)

Figure 3. Partitions of a tetrahedral SV supporting quadratic and cubic data reconstructions, shown in (a) and (b), respectively.

where *d* is the spatial dimension of the problem. The DOFs in the SV method are the volume-averaged conserved variables at the *N* CVs. Define the CV-averaged conserved variable for $C_{i,j}$ as

$$\overline{Q}_{i,j} = \frac{1}{V_{i,j}} \int_{C_{i,j}} Q dV \,, \quad j=1,\ldots,N, \quad i=1,\ldots,I, \tag{2.3}$$

where $V_{i,j}$ is the volume of $C_{i,j}$. Given the CV-averaged conserved variables for all CVs in S_i, a polynomial $p_i(x,y,z) \in P^m$ (the space of polynomials of at most degree m) can be reconstructed such that it is a $(m+1)^{\text{th}}$ order accurate approximation to $Q(x,y,z)$ inside S_i.

$$p_i(x,y,z) = Q(x,y,z) + O\left(h^{m+1}\right), \qquad (x,y,z) \in S_i\,, \quad i=1,\ldots,I, \tag{2.4}$$

where h is the maximum edge length of all the CVs. This reconstruction can be solved analytically by satisfying the following conditions:

$$\frac{1}{V_{i,j}} \int_{C_{i,j}} p_i(x,y,z)dV = \overline{Q}_{i,j}\,, \quad j=1,\ldots..,N. \tag{2.5}$$

This polynomial $p_i(x,y,z)$ is the $(m+1)^{th}$ order approximation we are looking for as long as the solution $Q(x,y,z)$ is smooth in the region covered by S_i. The reconstruction can be expressed more conveniently as

$$p_i(x,y,z) = \sum_{j=1}^{N} L_j(x,y,z)\overline{Q}_{i,j}\,, \tag{2.6}$$

where $L_j(x,y,z) \in P^m$ are the shape functions which satisfy

$$\frac{1}{V_{i,j}} \int_{C_{i,j}} L_n(x,y,z)dV = \delta_{jn}\,. \tag{2.7}$$

Some example plots of the shape functions for a degree 3 polynomial are shown in Figure 4. Integrating (2.1a) in $C_{i,j}$, we obtain

$$\frac{d\overline{Q}_{i,j}}{dt} + \frac{1}{V_{i,j}} \sum_{r=1}^{K} \int_{A_r} \left(\vec{F} \cdot \vec{n}\right) dA = 0\,, \qquad j=1,\ldots,N, \qquad i=1,\ldots,I, \tag{2.8}$$

where $\vec{F} = (f,g,h)$, A_r represents the r^{th} face of $C_{i,j}$, $\vec{n}$ is the outward unit normal vector of A_r, and K is the number of faces in $C_{i,j}$. More details of this can be found in [51]. If $\vec{F}$ is a

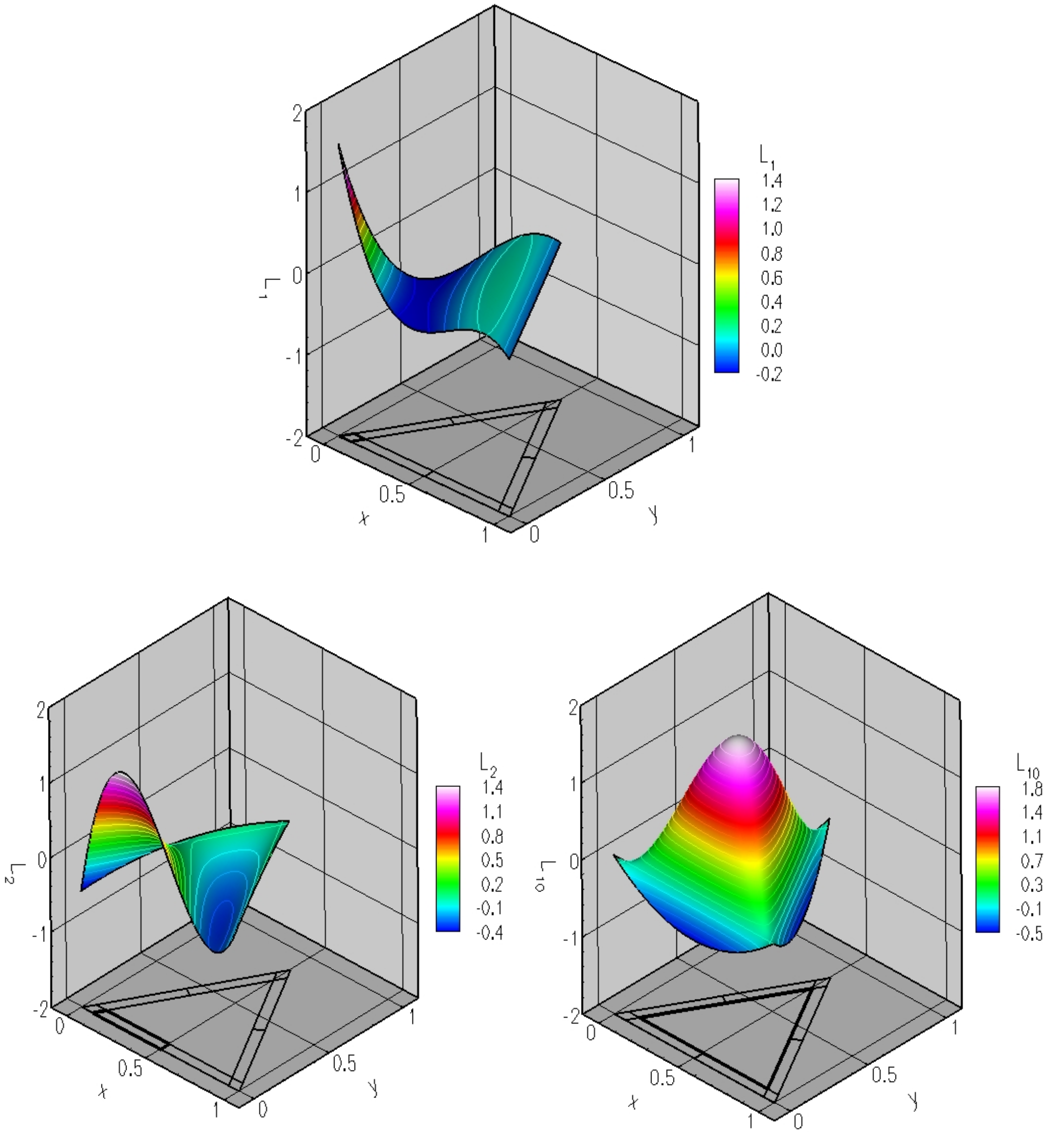


Figure 4. Shape functions to support a degree 3 polynomial reconstruction for the partition shown in Figure 1c. The partition is projected below the shape function plot, with the corresponding CV in outlined in bold.

nonlinear function of the conserved variables, then the surface integration on each face is performed with a $(m+1)^{th}$-order-accurate Gauss quadrature formula; i.e.,

$$\int_{A_r} (\vec{F}\cdot\vec{n})dA \cong \sum_{q=1}^{J} w_{rq}\vec{F}(Q(x_{rq},y_{rq},z_{rq}))\cdot\vec{n}_r A_r \ , \tag{2.9}$$

where J=integer[(m+2)/2] is the number of quadrature points on the r^{th} face in 2D, w_{rq} are the Gauss quadrature weights, and (x_{rq},y_{rq},z_{rq}) are the Gauss quadrature points. Since the reconstructed solution is discontinuous across SV interfaces, the interface flux is not uniquely defined. Therefore the normal flux in (2.9) is replaced by a "numerical" Riemann flux, such as the Rusanov [36], Roe [35] or HLLC [25,41,42,6] flux. For example, the Rusanov flux can be written as

$$\hat{F}(Q_L,Q_R,\vec{n}) \equiv \frac{1}{2}\left[\vec{F}(Q_L)\cdot\vec{n} + \vec{F}(Q_R)\cdot\vec{n} - \alpha(Q_R - Q_L)\right] \cong \vec{F}(Q)\cdot\vec{n} \equiv F_n, \tag{2.10}$$

where Q_L and Q_R are the reconstructed solution at the left (inside $C_{i,j}$) and right side (outside $C_{i,j}$) of A_r, α is the maximum absolute eigenvalue of the flux Jacobian in $\vec{n}$ direction, evaluated at an average state of Q_L and Q_R. In [50], it was shown that the above SV scheme is $(m+1)^{\text{th}}$ order accurate. For time integration, we use the 3rd order Strong Stability-Preserving (SSP) Runge-Kutta scheme [18], as shown in (2.11):

$$\begin{aligned} \overline{Q}^{(1)} &= \overline{Q}^{n} + \Delta t R\left(\overline{Q}^{n}\right), \\ \overline{Q}^{(2)} &= \frac{3}{4}\overline{Q}^{n} + \frac{1}{4}\left[\overline{Q}^{(1)} + \Delta t R\left(\overline{Q}^{(1)}\right)\right], \\ \overline{Q}^{n+1} &= \frac{1}{3}\overline{Q}^{n} + \frac{2}{3}\left[\overline{Q}^{(2)} + \Delta t R\left(\overline{Q}^{(2)}\right)\right]. \end{aligned} \tag{2.11}$$

2.2 The data reconstruction

One of the most important aspects of the SV method, as with any CFD method, is the data reconstruction. This plays a vital role in determining the accuracy and stability properties of the scheme, and thus should be given careful attention. The reconstruction stencil for the SV method is defined by the partitioning of the SV into CVs as shown in Figure 2 and Figure 3. Once a particular partition has been selected, the shape functions defined by that partition are used to build a high-order polynomial for each SV, thus completing the reconstruction. These shape functions are computed as

$$L(x,y) = e(x,y)R^{-1}, \tag{2.12}$$

where $e(x,y)$ is the basis function vector $[e_1(x,y), e_2(x,y), e_3(x,y),\ldots, e_N(x,y)]$ which contains the complete polynomial basis given the dimension of the approximation space, and R is the reconstruction matrix computed by

$$R = \begin{bmatrix} \frac{1}{V_1}\int\limits_{C_{,1}} e_1(x,y)dV & \cdots & \frac{1}{V_1}\int\limits_{C_{,1}} e_N(x,y)dV \\ \vdots & \ddots & \vdots \\ \frac{1}{V_N}\int\limits_{C_{,N}} e_1(x,y)dV & \cdots & \frac{1}{V_N}\int\limits_{C_{,N}} e_N(x,y)dV \end{bmatrix}, \tag{2.13}$$

where $C_{,i}$ denotes the i^{th} control volume in the standard element. In general, the shape and partition of an SV can be arbitrary as long as a nonsingular reconstruction matrix is obtained. However, if a geometrically similar partition is applied to a simplex, the result is that even though the shapes of the SVs may be different, they all utilize the same reconstruction where the functional values of the shape functions at similar points are exactly the same. Thus, we focus our efforts solely on simplex SVs which are triangular (2D) or tetrahedral (3D), in which all CVs are comprised of planar polygons (2D) or planar polyhedra (3D).

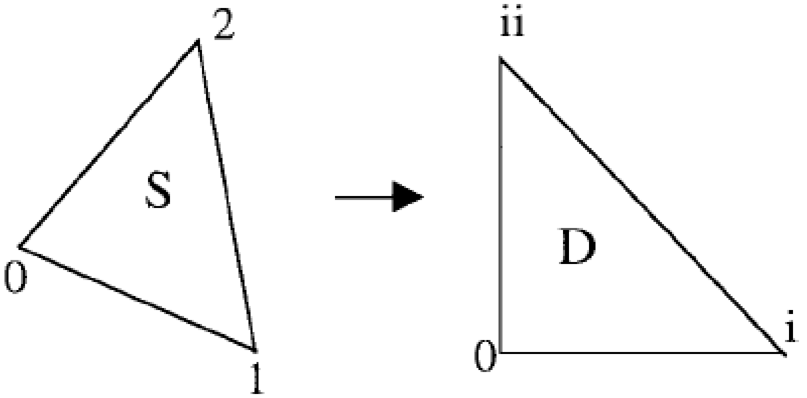


Figure 5. Schematic illustrating mapping from a physical triangle to a standard right triangle.

To take advantage of the universality of the reconstruction for the SV method applied to a simplex, we must define a transformation from a physical simplex to a standard simplex. For example, consider the transformation *Ψ*: *S*→*D*, shown in Figure 5, which transforms an arbitrary triangle *S* to a right triangle *D*, the three dimensional analog of which is a transformation from an arbitrary tetrahedron to a right tetrahedron. Let us use (x, y) to denote coordinates in *S* and (ξ, η) to denote coordinates of *D*. Then, assume that $\boldsymbol{r}_0=(x_0,y_0)$, $\boldsymbol{r}_1=(x_1,y_1)$ and $\boldsymbol{r}_2=(x_2,y_2)$ define the triangle in *S*, corresponding to (0,0), (1,0), and (0,1) in *D*, respectively. We can then write the transformation as

$$\Psi : r = r_0 + (r_1 - r_0)\xi + (r_2 - r_0)\eta, \quad \xi \geq 0, \eta \geq 0, \quad \text{and} \quad \xi + \eta \leq 1. \tag{2.14}$$

Thus, since the above transformation is linear, it can be easily shown that

$$e(x, y) = e(\xi, \eta)T, \tag{2.15}$$

where *T* is the transformation matrix containing only the nodal positions of *S*. It can also be shown that

$$dV = dxdy = 2Vd\xi d\eta, \tag{2.16}$$

for the above 2D transformation, or

$$dV = dxdydz = 6Vd\xi d\eta d\zeta \,, \tag{2.17}$$

for a 3D transformation, where *V* is the volume of *S*. Substituting (2.15) and (2.16) into (2.12), yields

$$L = \left[e_1(\xi,\eta) \quad \cdots \quad e_N(\xi,\eta)\right]$$

$$\begin{bmatrix} \int\limits_{C_{,1}} e_1(\xi,\eta)d\xi d\eta & \cdots & \int\limits_{C_{,1}} e_N(\xi,\eta)d\xi d\eta \\ \vdots & \ddots & \vdots \\ \int\limits_{C_{,N}} e_1(\xi,\eta)d\xi d\eta & \cdots & \int\limits_{C_{,N}} e_N(\xi,\eta)d\xi d\eta \end{bmatrix}^{-1} \begin{bmatrix} \frac{V_{,1}}{2V} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{V_{,N}}{2V} \end{bmatrix} . \tag{2.18}$$

We thus have a universal reconstruction formula for evaluating the conserved variables at similar points. This reconstruction needs only to be carried out once, and this can be done analytically using a program such as Mathematica [55], or in the case of adaptive polynomial refinement it may be desirable to perform the reconstruction numerically on-the-fly as the simulation is running. An important distinction between the SV method and the FV method is that all CVs within an SV utilize the same data reconstruction, and thus the memory- and CPU-intensive reconstructions employed in a FV method can be solved analytically and no extra storage is required.

It was shown in [51] that the so-called Lebesgue constant computed as

$$\left\|\Gamma_\Pi\right\| = \max_{\xi,\eta} \sum_{j=1}^{N} \left|L_j(\xi,\eta)\right| , \tag{2.19}$$

gives a simple method of bounding the interpolation polynomial. Thus the smaller the Lebesgue constant $\left\|\Gamma_\Pi\right\|$, the better the interpolation polynomial in terms of error bounds. However, although this criteria can be helpful in finding accurate SV partitions, it carries no information about the stability properties of the resulting scheme, and is thus a necessary but not sufficient tool for designing SV partitions.

2.3 Review of recent work

Recently, several different researchers have contributed to the development and application of the SV method. In particular, Haga et al. [19] implemented the 3D SV method in parallel and utilized the resources of the Earth simulator in an aerospace application. In Van den Abeele et al. [43-44], the stability properties of the SV method were analyzed in both 1D and 2D, respectively. In that work, several known SV schemes were found to suffer from weak instabilities, and new stable SV schemes were proposed. In Van den Abeele et al. [45], the stability properties of the 3D SV method were investigated, and again, some weak instabilities were revealed. In Chapter 5, we propose an improvement to the above stability analysis which uses a constrained minimization approach in the design of new stable SV partitions.

CHAPTER 3. QUADRATURE-FREE IMPLEMENTATION OF SV METHOD

3.1 Formulation of efficient new approaches

In consideration of the potentially prohibitive cost of the SV method in 3D, we present two different approaches to compute the face integral in (2.8) more efficiently. These approaches are also described in Harris et al. [22,23], and Yang et al. [58]. In each approach, a nodal set is defined within each SV to support a polynomial reconstruction for the flux vector. The shape functions defined by this nodal set are then integrated analytically over a standard element, resulting in an analytical representation for the flux integral on all faces. This procedure is independent of the problem being solved, as opposed to a recently presented quadrature-free approach for the DG method [3] which relies on a flux expansion which is heavily problem dependent.

For the methods presented here, the reconstruction is universal for all SVs if a nodal set is distributed in a geometrically similar manner for all SVs. The first approach employs the above methodology for the internal faces only, while using the Gauss quadrature formulas for faces bounding an SV. This approach will be referred to as the partial quadrature (PQ) approach. The second approach employs the above methodology for all faces, and is thus referred to as the quadrature-free (QF) approach. Both approaches are shown to be stable and convergent.

3.1.1 Quadrature-free (QF) approach

In the QF approach, a nodal set, such as those shown in Figure 6, is selected from Hesthaven [26-27]. This nodal set is then used to reconstruct a high-order polynomial approximation for the flux vector, and then the flux integrals are computed analytically, without the need for Gauss quadrature formulas. If we assume *Q*, *f,* *g* and *h* to be all polynomials in (2.1a), obviously *f*, *g* and *h* should be one degree higher than *Q*. Therefore, a polynomial of degree *m*+1 is employed to reconstruct the flux vector.

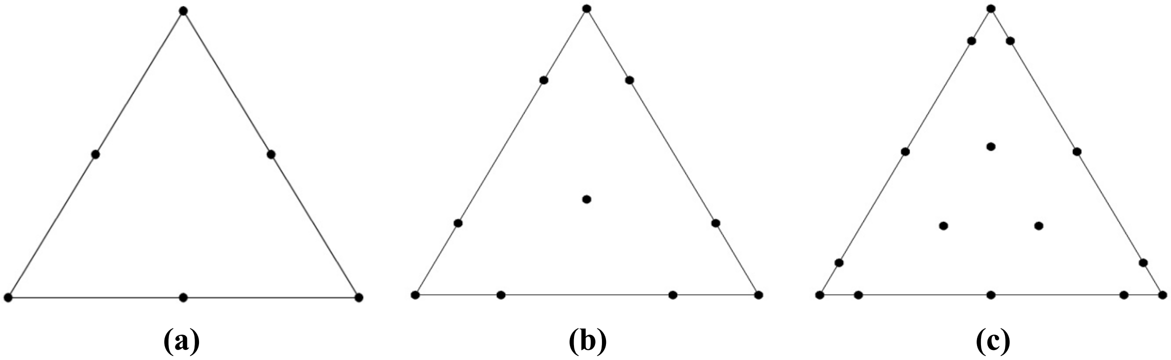


Figure 6. Nodal sets in a triangular SV supporting quadratic, cubic and quartic data reconstructions for the flux vector, shown in (a), (b) and (c), respectively.

The flux vector $\vec{F}$ can be computed at any point (*x,y,z*) by the following

$$\vec{F}(x,y,z)=\sum_{i=1}^{N_S} M_i(x,y,z)\vec{F}_i\,, \tag{3.1}$$

where N_s is the number of nodes in the nodal set, calculated from (2.2), and $\vec{F}_i$ is the flux vector at node *i* computed using the conserved variables at node *i* given by (2.6). Again, *Q*(*x,y,z*) and $\vec{F}(x,y,z)$ are taken to be continuous polynomials within each SV, thus (2.6) and (3.1) are valid within a given SV and not across SV boundaries, so interpolations are only carried out locally within a given SV. $M_i(x, y, z)$ are the shape functions defined by the nodal set which satisfy

$$M_n\left(x_j, y_j, z_j\right) = \delta_{jn}. \tag{3.2}$$

These are calculated by a procedure similar to that presented in Section 2.2, except that we are now dealing with nodal values of the solution instead of CV-averaged values. The above shape functions are thus computed as

$$M = \left[e_1(\xi,\eta) \quad \cdots \quad e_{N_S}(\xi,\eta)\right]\begin{bmatrix} e_1(\xi_1,\eta_1) & \cdots & e_{N_S}(\xi_1,\eta_1) \\ \vdots & \ddots & \vdots \\ e_1\left(\xi_{N_S},\eta_{N_S}\right) & \cdots & e_{N_S}\left(\xi_{N_S},\eta_{N_S}\right) \end{bmatrix}^{-1}. \tag{3.3}$$

where (ξ_i , η_i) is the location of node i in the standard element. Some representative 2D examples of the shape functions are shown in Figure 7 and Figure 8.

(3.1) is integrated over each internal face, resulting in an analytical representation for the flux integral on all internal faces. This is done using a program capable of symbolic calculus, such as Mathematica [55]. Or, as will be discussed in Chapter 4, in the case of local adaptive p-refinement, it is desirable to perform the integration of (3.1) numerically and on-the-fly as the grid and solution adapt. Regardless of the method of calculation, this allows for the flux integral on each internal face to be computed as a weighted average of the flux evaluated at the nodal set, i.e.;

$$\int_{A_r} \vec{F} \cdot \vec{n} dS = A_r \sum_{i=1}^{N_S} \overline{M}_i \vec{F}_i \cdot \vec{n}_r, \tag{3.4}$$

where $\overline{M}_i$ are the face-averaged shape functions for face r. In practice, face-averaged shape functions are computed during preprocessing for a standard element and then multiplied by the physical face area for computation. This eliminates the need to store all of the Jacobians.

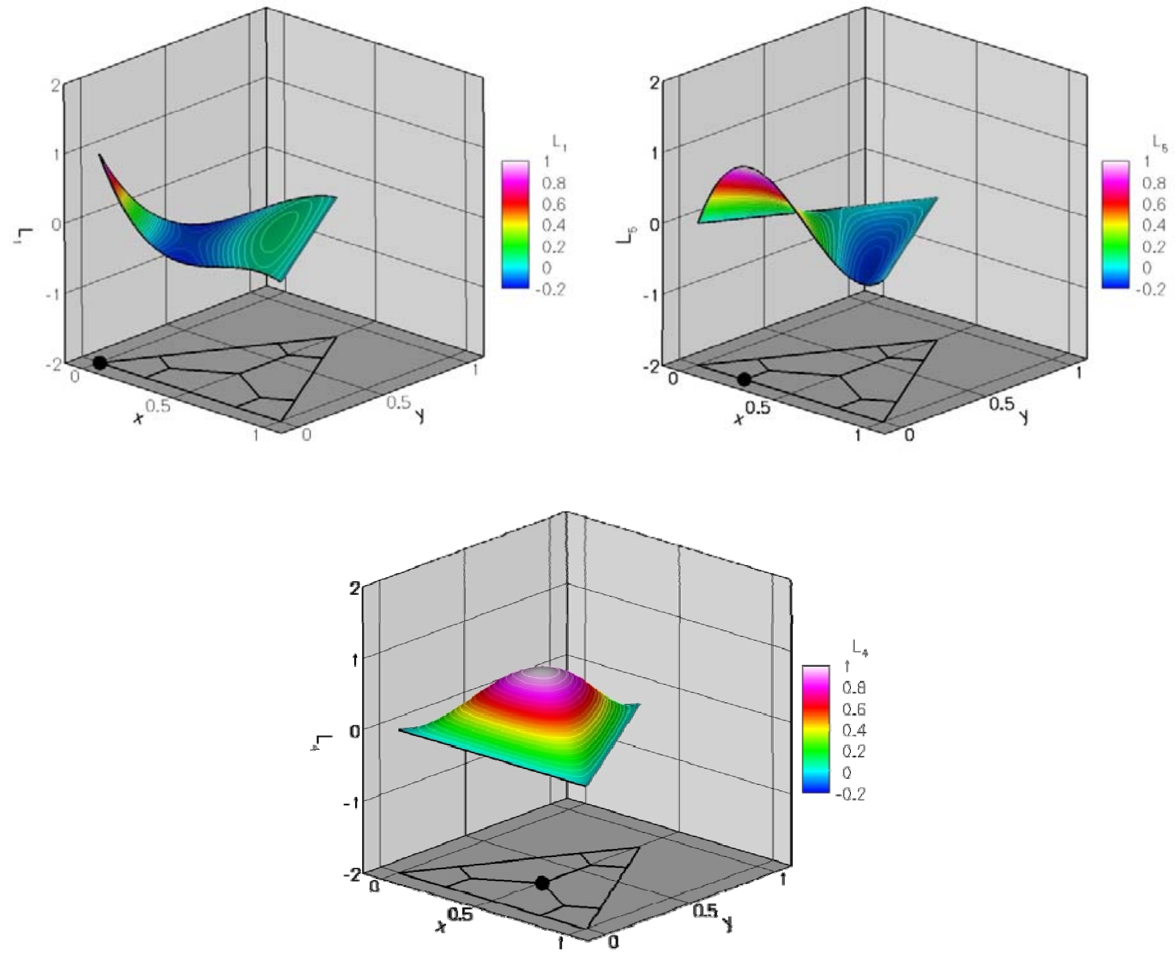


Figure 7. Shape functions in a triangular SV supporting a cubic data reconstruction for the flux vector. The node which each function corresponds to is shown in bold.

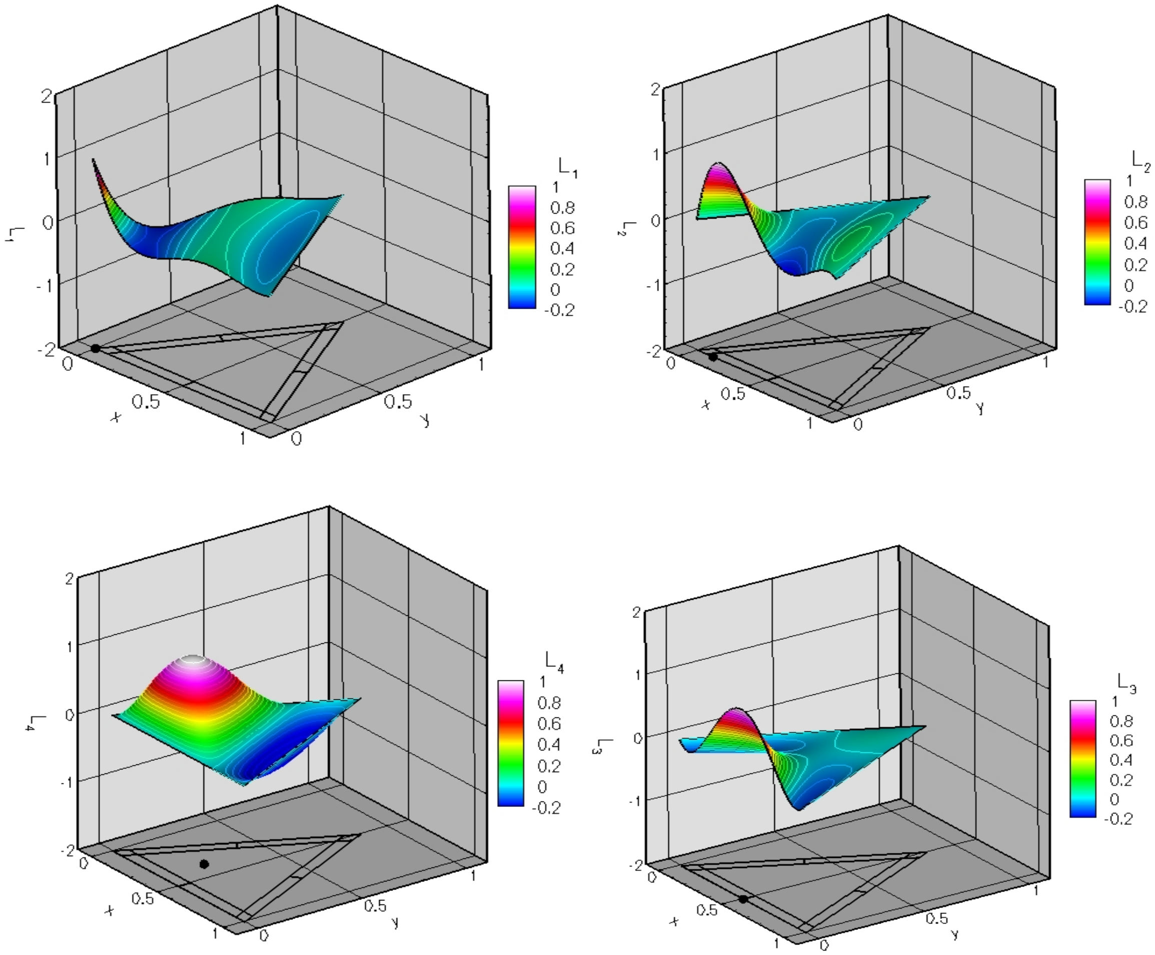


Figure 8. Shape functions in a triangular SV supporting a quartic data reconstruction for the flux vector. The node which each function corresponds to is shown in bold.

Although the face integrals on internal faces are evaluated as above, some care is required for the faces bounding the SV. Here, (2.10) is integrated over each SV-bounding CV face, and the resulting face integral can be expressed as the integral of a Riemann flux as follows:

$$\int_{A_r} \vec{F}\cdot\vec{n}dS \cong \frac{A_r}{2}\left[\overline{F}_{n,L}+\overline{F}_{n,R}-\alpha_c\left(\overline{Q}_R-\overline{Q}_L\right)\right], \tag{3.5}$$

where $\overline{F}_{n,L}$ and $\overline{F}_{n,R}$ denote the face-averaged normal component of the flux vector due to the SV to the left and right of the interface, respectively, and α_c is taken as either the maximum absolute eigenvalue as in the Rusanov flux [36], or the dissipation matrix as in the Roe flux [35], which are evaluated at the face center. $\overline{Q}_R$ and $\overline{Q}_L$ are the face-averaged conserved variables due to the SV to the right and left of the interface, respectively. (3.5) can be deduced from the following analysis, which assumes a Rusanov flux. Since $p_i(x, y, z) = Q(x, y, z) + O(h^{m+1})$, the following is also true,

$$\begin{aligned} Q_R - Q &= O(h^{m+1}), \\ Q_L - Q &= O(h^{m+1}), \\ Q_R - Q_L &= O(h^{m+1}), \end{aligned} \tag{3.6a}$$

and the maximum absolute eigenvalue can be expanded according to its value at the face center, i.e., $\alpha = \alpha_c + O(h)$. So we can say that

$$\alpha(Q_R - Q_L) = \alpha_c(Q_R - Q_L) + O(h^{m+2}). \tag{3.6b}$$

Integrating (3.6b) over a face, we obtain

$$\int_{A_r} \alpha(Q_R - Q_L) dA = A_r \alpha_c (\overline{Q}_R - \overline{Q}_L) + O(A_r h^{m+2}). \tag{3.7}$$

At each node shown in Figure 6, the flux is computed based on the reconstructed solution polynomial. Therefore, the flux error at each node of the nodal set is of order $O(h^{m+1})$, i.e.,

$$\begin{aligned} F_{n,L} &= F_n + O(h^{m+1}), \\ F_{n,R} &= F_n + O(h^{m+1}), \\ (F_{n,L} + F_{n,R})/2 &= F_n + O(h^{m+1}), \end{aligned} \tag{3.8}$$

where F_n is the (unknown) “exact” normal flux at the face. Integrating the 3rd equation in (3.8) over a face, we obtain

$$\int_{A_r} \tfrac{1}{2}\left(F_{n,L}+F_{n,R}\right)dA = \tfrac{1}{2}A_r\left(\overline{F}_{n,L}+\overline{F}_{n,R}\right) = \overline{F}_n A_r + O\left(A_r h^{m+1}\right). \tag{3.9}$$

So summarizing (3.7) and (3.9), we obtain

$$\int_{A_r} \vec{F}\cdot\vec{n}dA \equiv \overline{F}_n A_r = \frac{A_r}{2}\left[\overline{F}_{n,L}+\overline{F}_{n,R}-\alpha_c\left(\overline{Q}_R-\overline{Q}_L\right)\right]_r + O\left(A_r h^{m+1}\right). \tag{3.10}$$

If $\vec{F}$ = constant, the following identity exists:

$$\sum_{r=1}^{K}\int_{A_r}\vec{F}\cdot\vec{n}dA = 0. \tag{3.11}$$

Therefore, we will gain an extra order of accuracy if we sum up the surface integrals for the faces of $C_{i,j}$; i.e.,

$$\sum_{r=1}^{K}\int_{A_r}\vec{F}\cdot\vec{n}dA = \sum_{r=1}^{K}\frac{A_r}{2}\left[\overline{F}_{n,L}+\overline{F}_{n,R}-\alpha_c\left(\overline{Q}_R-\overline{Q}_L\right)\right]_r + O\left(A_r h^{m+2}\right). \tag{3.12}$$

Since $O(V_{i,j}) = O(A_r\, h)$, we have

$$\frac{d\overline{Q}_{i,j}}{dt} + \frac{1}{V_{i,j}}\sum_{r=1}^{K}\frac{A_r}{2}\left[\overline{F}_{n,L}+\overline{F}_{n,R}-\alpha_c\left(\overline{Q}_R-\overline{Q}_L\right)\right]_r = O\left(h^{m+1}\right). \tag{3.13}$$

Thus, if the surface integral in (2.8) is evaluated using (3.5), spatial accuracy of order m+1 is assured.

To summarize, the first and second terms in (3.5) are computed as a weighted average of the flux evaluated at the nodal set of the left, and right SV, respectively using (3.4), and the third and fourth terms are computed as a weighted average of the cell-averaged conserved variables of the right and left SV, respectively, i.e.

$$\int_{A_r} Q dS = A_r \sum_{i=1}^{N} \overline{L}_i \overline{Q}_i, \qquad (3.14)$$

where $\overline{L_i}$ are the face-averaged CV-based shape functions for face *r*. Again, these are computed during preprocessing for a standard element and then multiplied by the physical face area for computation.

3.1.2 Partial-quadrature (PQ) approach

The PQ approach uses a similar methodology as in the QF approach, namely (3.4) is used to evaluate the internal face integrals in (2.8), but for computation of the SV-bounding face integrals, the traditional Gauss quadrature approach via. (2.9) is used in favor of (3.5). This approach is introduced primarily as an additional validation case for the new methodology.

3.1.3 Cost comparison of different approaches for flux integration

In a quick comparison of the approaches, a 2D 3rd order SV method using the traditional "full quadrature" or FQ approach with 2 quadrature points per face necessitates 36 flux calculations per SV, the PQ approach requires 28 flux calculations per SV, and the QF approach requires only 10 flux calculations per SV. Similarly, for a 4th order SV method, the number of flux calculations per SV is 54, 39, and 15 for the FQ, PQ, and QF approaches, respectively. So, the PQ and QF approaches are slightly more efficient than the FQ approach in 2D, but are much less costly to evaluate in 3D where the faces number in the hundreds. Table 1 summarizes the costs for the PQ and QF approaches in 2D and 3D.

Table 1. Comparison of costs for PQ and QF approaches for vortex propagation problem (defined in Section 3.4) on 80x80x2 regular grid with single-stage time integration and Rusanov flux.

Order	Approach	CPU time(μs)/SV/iteration	Num. of flux calcs./SV/iteration	
		2D	2D	3D
3	PQ	31.63	28	200
	QF	25.42	10	20
4	PQ	51.28	39	371
	QF	46.92	15	35

It is evident that the QF approach is slightly less expensive than the PQ approach in 2D in terms of CPU time, as expected. Although the real savings presents itself in 3D, which can be seen from the required flux calculations presented in Table 1.

3.2 Data limiting for the QF approach

For the non-linear Euler equations, it is necessary to perform data limiting to maintain stability if the solution contains discontinuities. There are many different ways of applying limiters in the system setting. Here we choose the primitive variable approach and enforce monotonicity using a TVD limiter. To this end, we first establish the following numerical monotonicity criterion for each SV

$$\overline{\psi}_i^{\min} \le \psi_i(x_{rn}, y_{yn}, z_{rn}) \le \overline{\psi}_i^{\max}, \tag{3.15}$$

where $\overline{\psi}_i^{\min}$ and $\overline{\psi}_i^{\max}$ are the minimum and maximum SV-averaged primitive variables $\{p,\rho,u,v,w\}$ among all the neighboring SVs sharing a face (called face neighbors), or sharing a node (called node-neighbors), with S_i, including the SV under consideration, and $\psi_i(x_{rn}, y_{rn}, z_{rn})$ are the reconstructed primitive variables at the nodal set of the SV under consideration. The SV-averaged primitive variables are computed by

$$\overline{\psi}_{i,j} = \sum_{k=1}^{N_S} \overline{M}_{i,j,k} \psi_k . \tag{3.16}$$

and

$$\overline{\psi}_i = \frac{1}{V_i} \sum_{j=1}^{N} \overline{\psi}_{i,j} V_{i,j} , \tag{3.17}$$

where $\overline{\psi}_{i,j}$ are the CV-averaged primitive variables, $\overline{M}_{i,j,k}$ are the CV-averaged node-based shape functions, ψ_k are the primitive variables evaluated at node *k*, and V_i is the volume of SV *i*. The CV-averaged node-based shape functions, $\overline{M}_{i,j,k}$, are computed during preprocessing for a standard element.

In the cases considered here, only *p* and *ρ* are used in determining whether a given SV is to be limited to guarantee the positivity of *p* and *ρ*. Additionally, $\overline{\psi}_i^{\min}$ and $\overline{\psi}_i^{\max}$ are scaled by (1-ϵ) and (1+ϵ), respectively, where $\epsilon \ll 1$. This prevents unnecessary limiting based purely on numerical round-off. Here we use $10^{-8} < \epsilon < 10^{-3}$. Obviously if ϵ is large, the limiter is no longer strictly TVD.

If (3.15) is violated for any node in the nodal set of the SV under consideration, then it is assumed that the SV is close to a discontinuity, and all primitive variables in the SV are linear, i.e.,

$$\nabla \psi_i = Const, \quad \forall \boldsymbol{r} \in S_i . \tag{3.18}$$

The magnitude of this gradient is maximized subject to the monotonicity condition given in (3.15). The initial guess of the gradient $\nabla \psi_i$ for each SV is computed as

$$\nabla \psi_i \cong \sum_{j=1}^{N} \nabla \overline{L}_{i,j} \overline{\psi}_{i,j} \tag{3.19}$$

where $\nabla \overline{L}_{i,j}$ represents the SV-averaged gradient of the CV-based shape functions. Previous work has used the gradient at the center of the SV as the guess gradient [51]. In general, for high-order polynomials, the gradient at the SV center may be inadequate for this procedure, so (3.19) is used instead. Still, the gradient $\nabla \psi_i$ in (3.19) may not satisfy (3.15). Therefore it is limited by multiplying by a scalar $0 \le \varphi \le 1$. The scalar is computed from

$$\varphi = \begin{cases} \min\left(1, \dfrac{\overline{\psi}_i^{\max} - \overline{\psi}_i}{\psi_i(x_{rn}, y_{rn}, z_{rn}) - \overline{\psi}_i}\right) & \textit{if } \psi_i(x_{rn}, y_{rn}, z_{rn}) > \overline{\psi}_i \\ \min\left(1, \dfrac{\overline{\psi}_i^{\min} - \overline{\psi}_i}{\psi_i(x_{rn}, y_{rn}, z_{rn}) - \overline{\psi}_i}\right) & \textit{if } \psi_i(x_{rn}, y_{rn}, z_{rn}) < \overline{\psi}_i . \\ 1 & \textit{otherwise} \end{cases} \tag{3.20}$$

This concludes the formulation for a TVD limiter. If necessary, the above primitive variable formulation can be converted to a conservative variable formulation if Ψ is replaced with Q. This turns out to work better for some situations, as will be mentioned in Section 4.3.

Although the above limiting procedure guarantees the positivity of p and ρ provided their SV-averages are positive, a negative p or ρ could still result from an update step in the time integration. So, and additional measure needs to be in place to prevent this catastrophic situation from occurring. This is a moot point for steady simulations, since the pressure or density could just be overwritten with small positive values before the eigenvalue computation, and the results would likely still converge to the correct steady-state solution. However, we are interested in solving unsteady flow problems, so such a quick fix is not sufficient. Here when we update the solution for each time-step, and sub time-step, we check to see if the update produced a negative p or ρ at any point where an eigenvalue, or dissipation matrix as in Roe flux, calculation must be performed. If negative p or ρ are detected in this manor, we go back and locally redo the update step with a decreased time step size. Since this correction is only activated to avoid catastrophic oscillations, the error introduced should be negligible.

3.3 Curved boundary implementation for QF approach

One of the biggest advantages of high-order methods over low-order methods is their ability to achieve better solution accuracy using far fewer degrees-of-freedom when the solution is smooth. However, for problems involving curved wall boundaries which are represented by line-segments or planar-facets, an inordinate amount of elements may be necessary just to preserve the geometry with adequate precision. A much more desirable approach is to represent curved boundaries with high-order polynomials that are compatible with the order of the data reconstruction. We present three different methodologies to accomplish this, the isoparametric element approach [54], a new simplified approach, and the approach of Krivodonova and Berger [31].

3.3.1 Isoparametric element approach

The first approach, and most expensive, involves the use of high-order SVs with one or more curved sides. These physically nonlinear SVs are referred to as isoparametric elements, which are commonly used in the finite element community. The main complication arising from the application of this technique to the SV method is that the traditional definition of the cell-averaged conserved variables, as given in (2.3), must now be modified to account for CVs which are no longer planar polygons (2D) or planar polyhedra (3D).

This procedure involves a number of steps. First, the wall boundary which is to be treated as curved is identified. Then, a cubic-spline routine is employed to fit a piecewise cubic polynomial along the entire curved surface. Now that we have a cubic polynomial defined for each SV face on a curved boundary, we have essentially created new nonlinear SVs on the curved boundary, or isoparametric elements. Following the usual practice [54], isoparametric SVs can be used to map SVs with curved boundaries into the standard SV. Assume that a one-to-one transformation exists between a general SV in the physical space (x,y) and the standard triangle in the computational domain (ξ,η) as shown in Figure 9, i.e.,

$$\begin{aligned} \xi &= \xi(x, y), \\ \eta &= \eta(x, y). \end{aligned} \tag{3.21}$$

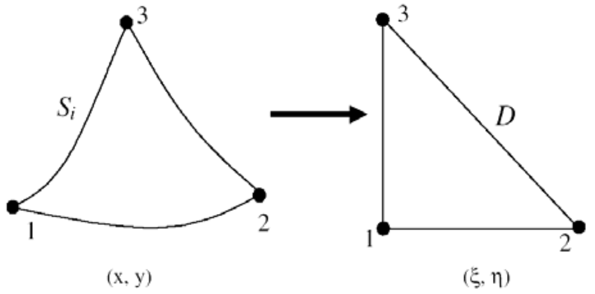


Figure 9. Transformation of a general SV in the physical domain to the standard triangle in the computational domain.

The partition of the SV is performed in the standard triangle, and the partition in the physical domain is the result of the inverse transformation from the computational domain back to the physical domain as shown in Figure 10.

$$\begin{aligned} x &= x(\xi,\eta), \\ y &= y(\xi,\eta). \end{aligned} \tag{3.22}$$

Therefore, a necessary condition for a valid transformation is that the inverse transformation exists.

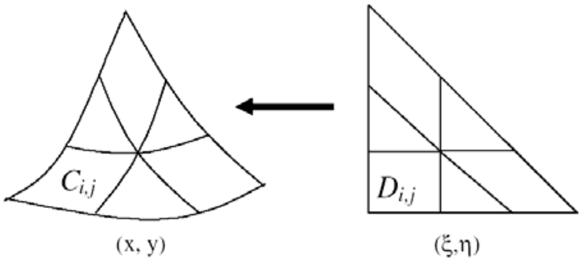


Figure 10. Transformation of the partition from the standard triangle in the computational domain to the general SV in the physical domain.

The DOFs for the general SV are then given as

$$\overline{Q}_{i,j} = \frac{\int\limits_{C_{i,j}} Q(x,y)dxdy}{\int\limits_{C_{i,j}} dxdy} = \frac{\int\limits_{D} Q(\xi,\eta)|J|d\xi d\eta}{\int\limits_{D} |J|d\xi d\eta}, \tag{3.23}$$

where J is the Jacobian matrix of the transformation, i.e., $J = \dfrac{\partial(x,y)}{\partial(\xi,\eta)}$. The reconstruction problem then becomes: given the CV-averaged conserved variables of an SV, build a degree m polynomial $p_i(\xi,\eta)$ such that

$$\int\limits_{D} p_i(\xi,\eta)|J|d\xi d\eta = \overline{Q}_{i,j}\int\limits_{D} |J|d\xi d\eta. \tag{3.24}$$

In the most general case, a quadratic isoparametric SV requires the specification of 6 nodes, as shown in Figure 11a. The transformation can be expressed as

$$r = \sum_{j=1}^{N} C_j(\xi, \eta) r_j , \tag{3.25}$$

where r=(x, y). The shape functions for a quadratic SV in 2D can be written as

$$\begin{aligned}
C_1(\xi,\eta) &= (1-\xi-\eta)(1-2\xi-2\eta), \\
C_2(\xi,\eta) &= \xi(2\xi-1), \\
C_3(\xi,\eta) &= \eta(2\eta-1), \\
C_4(\xi,\eta) &= 4\xi\eta, \\
C_5(\xi,\eta) &= 4\xi(1-\xi-\eta), \\
C_6(\xi,\eta) &= 4\eta(1-\xi-\eta).
\end{aligned} \tag{3.26}$$

However in most cases it is not necessary to use the most general isoparametric SV since only one boundary of the SV is usually curved as shown in Figure 11b and Figure 11c. As a result, simplified curved SVs (SCSV) can be used. The shape functions for the quadratic SCSV are given as

$$\begin{aligned}
C_1(\xi,\eta) &= 1-3\xi+2\xi(\xi+\eta)-\eta, \\
C_2(\xi,\eta) &= -\xi+2\xi(\xi+\eta), \\
C_3(\xi,\eta) &= \eta, \\
C_4(\xi,\eta) &= 4\xi(1-\xi-\eta),
\end{aligned} \tag{3.27}$$

and the shape functions for the cubic SCSV, as shown in Figure 11c, are given as

$$\begin{aligned}
C_1(\xi,\eta) &= 1-5.5\xi-\eta+9\xi(\xi+\eta)-4.5\xi(\xi+\eta)^2, \\
C_2(\xi,\eta) &= \xi\left[1-4.5(\xi+\eta)+4.5(\xi+\eta)^2\right], \\
C_3(\xi,\eta) &= \eta, \\
C_4(\xi,\eta) &= 9\xi\left[1-2.5(\xi+\eta)+1.5(\xi+\eta)^2\right], \\
C_5(\xi,\eta) &= \xi\left[-4.5+18(\xi+\eta)-13.5(\xi+\eta)^2\right]
\end{aligned} \tag{3.28}$$

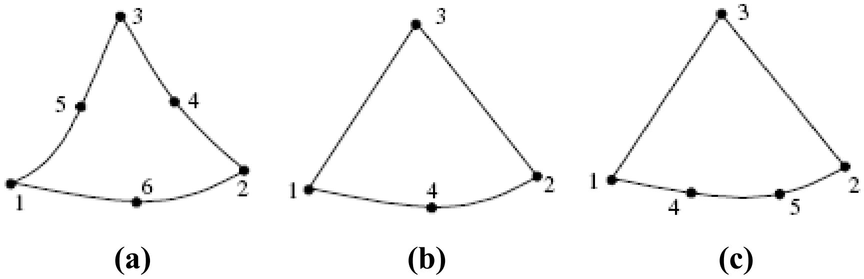


Figure 11. Quadratic (a), simplified quadratic (b), and simplified cubic (c) SVs with three and one curved boundaries, respectively.

More information on the isoparametric element approach can be found in Wang and Liu [54].

3.3.2 Simplified approach

One drawback of the isoparametric element approach is that it necessitates computation and storage of a separate reconstruction for each and every wall-bounding cell. For large problems, and especially in 3D, the memory requirement can become large. Here we present an approach in which the geometry is represented with high fidelity, but a separate reconstruction for every wall-bounding cell is avoided. Similar approaches have been recently presented by Krivodonova and Berger [31], and Luo et al. [33]. The basic idea employed here is that the flux integrals are carried out on the physically curved faces using the polynomial reconstruction from a linear SV, as opposed to a curved SV. This allows us to employ the same reconstruction for the SVs on curved boundaries that is already in use for all other SVs. Since the curvature will usually be very small, this is a reasonable approximation.

As with the previous method, a cubic-spline routine is employed to fit a piecewise cubic polynomial along the entire curved surface. Then (3.25) and (3.27) or (3.28) can be used to map points from a standard SV, into an SV with one quadratic or cubic side. Since we are not computing a new reconstruction based on isoparametric SVs, the purpose of this transformation is to modify the SVs such that the endpoints of the faces of wall-bounding CVs lie on the curved SV face. This is shown in Figure 12 for clarity. The node locations and face endpoints are computed using (3.25) and (3.27) or (3.28), and the face endpoints are all connected with straight lines.

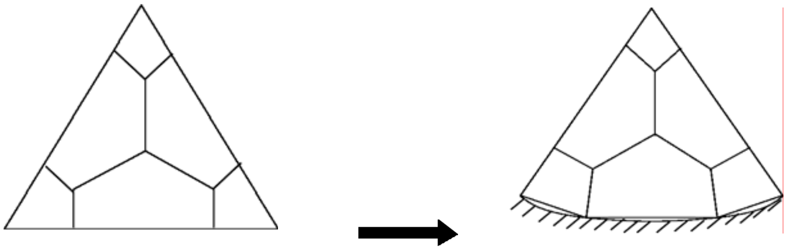


Figure 12. Traditional SV partition transformed to accommodate curved-wall boundary.

Although the CV faces themselves remain linear, with the CV face endpoints situated on the curved SV face, the flux integration can be carried out along the curved face and still satisfy conservation. Since the reconstruction is only universal for geometrically similar partitions and node distributions, there is some error inherent to the geometrical dissimilarity of the SVs on curved wall boundaries. In most situations, the SVs on curved wall boundaries will be small compared to all other SVs, and accordingly the degree of nonlinearity of these SV faces will also be small. It is thus reasoned that any error introduced in the reconstruction due to the presence of geometrically dissimilar SVs on curved wall boundaries will likely be small.

Let the equation of the r^{th} wall-bounding face of $C_{i,j}$ in the standard SV be

$$\eta = 0, \quad \xi_{r1} < \xi < \xi_{r2}. \tag{3.29}$$

Then the surface integral in (2.8) can be written as

$$\int_{A_r} \left(\vec{F} \cdot \vec{n}\right) dS = \int_{\xi_{r1}}^{\xi_{r2}} \vec{F} \cdot \left(\frac{dy}{d\xi}, -\frac{dx}{d\xi}\right) d\xi. \tag{3.30}$$

This line integral in the standard element can be evaluated using the standard Gauss quadrature formula

$$\int_{\xi_{r1}}^{\xi_{r2}} \vec{F} \cdot \left(\frac{dy}{d\xi}, -\frac{dx}{d\xi}\right) d\xi = \left(\xi_{r2} - \xi_{r1}\right) \sum_{q=1}^{J} w_{rq} F_N\left(\xi_{rq}\right), \tag{3.31}$$

where w_{rq} represent the Gauss quadrature weights, and $F_N = \vec{F} \cdot \left(\frac{dy}{d\xi}, -\frac{dx}{d\xi}\right)$ denotes the dot product of the flux vector with the area normal vector. More information on this approach can be found in Harris et al. [22].

3.3.3 Krivodonova and Berger approach

Here we follow an approach by Krivodonova and Berger [31] originally developed for the DG method in which the geometry is represented with high fidelity, but a separate reconstruction for every wall-bounding cell is avoided. The basic idea employed here is that straight-sided SVs, rather than curved SVs, are employed at wall boundaries, but a no-flow boundary condition is enforced for the physical boundary as opposed to the computational (piecewise-linear or planar-faceted) boundary. This allows us to utilize the same reconstruction for the SVs on curved boundaries that is already in use for all other SVs.

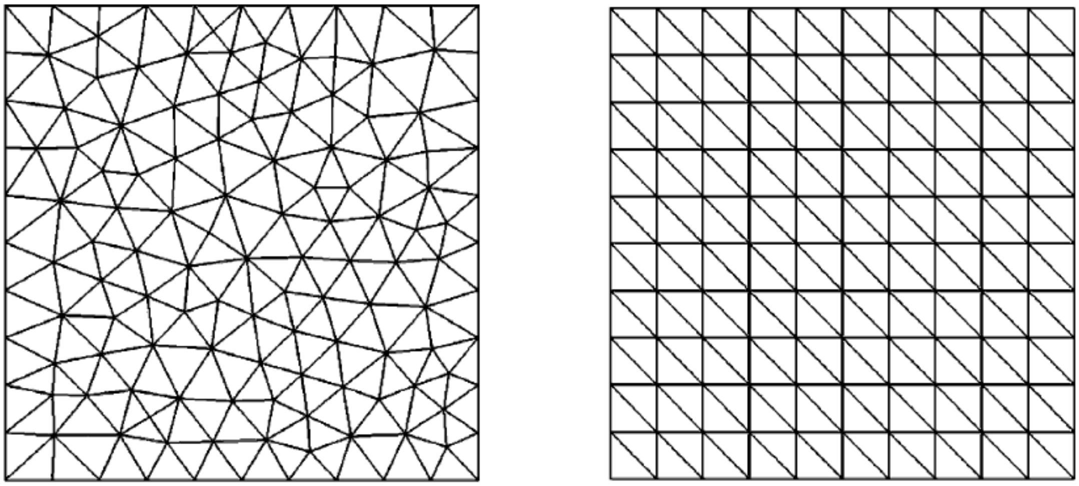
As with the previous method, a cubic-spline routine is employed to fit a piecewise cubic polynomial along the entire curved surface. Then (3.25) and (3.27) or (3.28) are used to map points from a standard SV, into an SV with one quadratic or cubic side. The unit-normal vector at any point on the curved face is then computed by

$$\vec{n} = \frac{\frac{dy}{d\xi}, -\frac{dx}{d\xi}}{\sqrt{\left(\frac{dy}{d\xi}\right)^2 + \left(\frac{dx}{d\xi}\right)^2}}. \tag{3.32}$$

It is important to note that the vector $\vec{n}$ is used only to define a ghost state at quadrature points, or nodes, located on curved-wall boundaries. The numerical flux is still computed through the straight-sided boundary. Thus, for the PQ approach, $\vec{n}$ must be computed and stored for each quadrature point on every curved-wall bounding SV. Whereas, in the QF approach, $\vec{n}$ must be computed and stored for each node in the nodal set of wall-bounding SVs that lies on the wall-bounding face. Additionally, $\vec{n}$ must be computed and stored at the CV face centers for curved-wall bounding SVs, to be used for computing the eigenvalue or dissipation matrix. Note that for simple geometries such as a cylinder, the above process is unnecessary as the normals can be computed directly.

3.4 Results

In this section, the efficient QF implementation of the SV method is evaluated for 2D scalar advection and Burger's equations, and for the 2D Euler equations. Both 3rd-order and 4th-order partitions are evaluated. The 3rd-order partition, and the corresponding nodal set to support a cubic reconstruction for the flux vector are shown in Figure 2b, and Figure 6b, respectively. The 4th-order partition, and the corresponding nodal set to support a quartic reconstruction for the flux vector are shown in Figure 2c, and Figure 6c, respectively. The numerical error is assessed using both regular and irregular grids, as shown in Figure 13. The finer grids are generated recursively by cutting each courser grid cell into four finer grid cells.



(a) **(b)**

Figure 13. Regular and irregular grids for validation cases: (a) regular (10x10x2); (b) irregular (10x10x2).

The third-order SSP Runge-Kutta scheme is used for time integration in all cases. Also, all errors presented are time-step independent because the time step Δt was made small enough so that the errors are dominated by the spatial discretization. The initial CV-averaged solutions were computed using the CV-averaged node-based shape functions. In this way, the solution is initialized at the nodal set, and the CV-averaged solutions are computed as a weighted average of these nodal values. This avoids the need for a high-order Gauss quadrature initialization. The solution is taken as converged when the L_2 norm of the residual is reduced by 10 orders of magnitude in all cases. The L_2 norm of the density residual is used for the Euler equation simulations.

3.4.1. Accuracy study with 2D linear advection equation

In this case, we test the accuracy of the SV method on the 2D linear advection equation:

$$\frac{\partial Q}{\partial t}+\frac{\partial Q}{\partial x}+\frac{\partial Q}{\partial y}=0\,,\quad -1\le x\le 1\,,\ -1\le y\le 1, \tag{3.33}$$

$$Q(x,y,0)=\sin\pi(x+y)\,,\ \text{periodic boundary condition.}$$

The numerical solutions were carried out until t=1 for this time-accurate problem. In Table 2 and Table 3, we present the L_1 and L_∞ errors and orders for regular and irregular grids, respectively.

Table 2. Accuracy of 2D linear advection equation at *t*=1 on regular grids.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	10x10x2	1.33e-3	-	3.29e-3	-
	20x20x2	1.73e-4	2.94	4.47e-4	2.88
	40x40x2	2.19e-5	2.98	5.77e-5	2.95
	80x80x2	2.76e-6	2.99	7.29e-6	2.98
4	10x10x2	7.10e-5	-	2.44e-4	-
	20x20x2	4.38e-6	4.02	1.56e-5	3.97
	40x40x2	2.73e-7	4.00	9.78e-7	3.99
	80x80x2	1.71e-8	4.00	6.12e-8	4.00

Grid refinement is carried out until both the L_1 and L_∞ orders are sufficiently demonstrated. For this case, the PQ results are identical to the QF results, as expected for a linear equation, so just the QF results are presented. The formal order of accuracy is observed, using both regular and irregular grids.

Table 3. Accuracy of 2D linear advection equation at *t*=1 on irregular grids.

Order of accuracy	Grid	L$_1$ error	L$_1$ order	L$_\infty$ error	L$_\infty$ order
3	10x10x2	3.12e-3	-	1.57e-2	-
	20x20x2	4.09e-4	2.93	1.93e-3	3.02
	40x40x2	5.36e-5	2.93	3.37e-4	2.52
	80x80x2	6.99e-6	2.94	6.18e-5	2.45
4	10x10x2	2.62e-4	-	1.50e-3	-
	20x20x2	1.56e-5	4.07	1.13e-4	3.73
	40x40x2	9.67e-7	4.02	8.23e-6	3.77
	80x80x2	6.03e-8	4.00	5.38e-7	3.94

3.4.2. Accuracy study with Burger's equation

In this case, we test the accuracy of the SV method on the 2D Burger's equation:

$$\frac{\partial Q}{\partial t}+\frac{\partial Q^2/2}{\partial x}+\frac{\partial Q^2/2}{\partial y}=0, \qquad -1\le x\le 1, \quad -1\le y\le 1, \tag{3.34}$$

$Q(x,y,0)=\frac{1}{4}+\frac{1}{2}\sin\pi(x+y)$, periodic boundary condition.

The numerical solutions were carried out until *t*=0.1 for this time-accurate problem. At this time the solution is still smooth, so data limiting is not necessary. In Table 4 and Table 5, we present the L_1 and L_∞ errors and orders for regular and irregular grids, respectively. Grid refinement is carried out until both the L_1 and L_∞ orders are sufficiently demonstrated. Results from the PQ and QF approaches are not identical for this case, but they are very similar, so again only the QF results are shown. The formal order of accuracy is again observed, using both regular and irregular grids.

Table 4. Accuracy of 2D Burger's equation at *t*=0.1 on regular grids.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	10x10x2	3.96e-4	-	1.63e-3	-
	20x20x2	6.49e-5	2.61	3.97e-4	2.03
	40x40x2	1.03e-5	2.66	7.01e-5	2.50
	80x80x2	1.55e-6	2.73	1.12e-5	2.65
4	10x10x2	3.28e-5	-	3.77e-4	-
	20x20x2	2.39e-6	3.78	3.26e-5	3.53
	40x40x2	1.74e-7	3.77	2.89e-6	3.49
	80x80x2	1.23e-8	3.83	2.08e-7	3.80

Table 5. Accuracy of 2D Burger's equation at *t*=0.1 on irregular grids.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	10x10x2	6.42e-4	-	3.35e-3	-
	20x20x2	1.17e-4	2.45	9.96e-4	1.75
	40x40x2	1.88e-5	2.65	2.66e-4	1.90
	80x80x2	2.91e-6	2.69	6.12e-5	2.12
	160x160x2	4.41e-7	2.72	1.14e-5	2.43
	320x320x2	6.51e-8	2.76	1.88e-6	2.59
4	10x10x2	8.35e-5	-	1.16e-3	-
	20x20x2	6.20e-6	3.75	1.43e-4	3.02
	40x40x2	4.57e-7	3.76	1.26e-5	3.51
	80x80x2	3.28e-8	3.80	1.06e-6	3.56

3.4.3. Accuracy study with 2D vortex propagation problem

The unsteady 2D Euler equations in conservative form can be written as

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0\,, \qquad (3.35)$$

where Q is the vector of conserved variables, F and G are the inviscid flux vectors given below:

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{Bmatrix}, \qquad F = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{Bmatrix}, \qquad G = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{Bmatrix}. \tag{3.36}$$

Here ρ is the density, u and v are the velocity components in x and y directions, p is the pressure, and E is the total energy. The pressure is related to the total energy by

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right), \tag{3.37}$$

with ratio of specific heats γ. In all of the simulations in this paper, γ is taken to be 1.4. This is an idealized problem for the Euler equations in 2D. The mean flow is $\{\rho, u, v, p\} = \{1,1,1,1\}$. An isotropic vortex is then added to the mean flow, i.e. with perturbations in u, v, and temperature $T = p/\rho$, and no perturbation in entropy $S=p/\rho^\gamma$:

$$(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5\left(1-r^2\right)}\left(-\bar{y}, \bar{x}\right), \qquad \delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \qquad \delta S = 0,$$

where $r^2 = \bar{x}^2 + \bar{y}^2$, $\bar{x} = x - 5$, $\bar{y} = y - 5$, and the vortex strength $\varepsilon = 5$. If the computational domain extends to infinity, the exact solution of the Euler equations with the above initial conditions is just the passive convection of the isotropic vortex with mean velocity (1, 1). In the following accuracy study, the computational domain is taken to be [0, 10] x [0, 10], with characteristic inflow and outflow boundary conditions imposed on the boundaries. The numerical solution is carried out until $t = 2$ on both regular and irregular grids. No limiters are employed in this study since the problem is smooth, and the Rusanov flux is used in all simulations. In Table 6 and Table 7, we present the L_1 and L_∞ errors and orders in the CV-averaged density for the regular grids using the PQ and QF approaches, respectively.

Table 6. Accuracy of 2D vortex propagation problem at *t*=2 on regular grids using PQ approach.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	10x10x2	1.41e-3	-	4.09e-2	-
	20x20x2	2.51e-4	2.49	4.88e-3	3.06
	40x40x2	4.30e-5	2.54	1.05e-3	2.22
	80x80x2	6.88e-6	2.65	1.95e-4	2.42
	160x160x2	1.05e-6	2.71	3.16e-5	2.63
4	10x10x2	6.08e-4	-	9.98e-3	-
	20x20x2	4.95e-5	3.62	1.12e-3	3.16
	40x40x2	3.03e-6	4.03	1.14e-4	3.30
	80x80x2	1.92e-7	3.98	7.28e-6	3.96
	160x160x2	1.36e-8	3.82	6.22e-7	3.55

Table 8 and Table 9 show the L_1 and L_∞ errors and orders for the irregular grids using the PQ and QF approaches, respectively. Comparison of the PQ and QF approaches for this case show very similar behavior. The formal order of accuracy is demonstrated for all cases.

Table 7. Accuracy of 2D vortex propagation problem at *t*=2 on regular grids using QF approach.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	10x10x2	1.44e-3	-	5.00e-2	-
	20x20x2	2.53e-4	2.51	5.05e-3	3.31
	40x40x2	4.32e-5	2.55	1.07e-3	2.24
	80x80x2	6.89e-6	2.65	1.98e-4	2.44
	160x160x2	1.05e-6	2.71	3.17e-5	2.64
4	10x10x2	5.49e-4	-	9.16e-3	-
	20x20x2	4.82e-5	3.51	8.90e-4	3.36
	40x40x2	3.04e-6	3.99	1.07e-4	3.06
	80x80x2	1.92e-7	3.99	6.97e-6	3.94
	160x160x2	1.36e-8	3.82	6.12e-7	3.51

Table 8. Accuracy of 2D vortex propagation problem at *t*=2 on irregular grids using PQ approach.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	10x10x2	1.11e-3	-	2.67e-2	-
	20x20x2	1.85e-4	2.58	3.23e-3	3.04
	40x40x2	3.00e-5	2.63	6.91e-4	2.23
	80x80x2	4.76e-6	2.66	1.32e-4	2.39
	160x160x2	7.36e-7	2.69	1.95e-5	2.75
4	10x10x2	3.46e-4	-	8.38e-3	-
	20x20x2	3.07e-5	3.49	4.52e-4	4.21
	40x40x2	2.07e-6	3.89	4.87e-5	3.21
	80x80x2	1.42e-7	3.87	3.89e-6	3.65
	160x160x2	9.49e-9	3.90	3.66e-7	3.41

Table 9. Accuracy of 2D vortex propagation problem at *t*=2 on irregular grids using QF approach.

Order of accuracy	Grid	L$_1$ error	L$_1$ order	L$_\infty$ error	L$_\infty$ order
3	10x10x2	1.11e-3	-	3.21e-2	-
	20x20x2	1.87e-4	2.57	3.36e-3	3.26
	40x40x2	3.01e-5	2.63	7.15e-4	2.23
	80x80x2	4.76e-6	2.66	1.33e-4	2.43
	160x160x2	7.36e-7	2.69	1.96e-5	2.76
4	10x10x2	3.44e-4	-	6.75e-3	-
	20x20x2	2.96e-5	3.54	4.42e-4	3.93
	40x40x2	2.05e-6	3.85	4.60e-5	3.26
	80x80x2	1.41e-7	3.86	3.75e-6	3.61
	160x160x2	9.49e-9	3.90	3.58e-7	3.39

3.4.4. Accuracy study with 3D vortex propagation problem

The unsteady 3D Euler equations in conservative form can be written as

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial H}{\partial z}=0, \qquad (3.38)$$

where Q is the vector of conserved variables, F, G, and H are the inviscid flux vectors given below:

$$Q=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{Bmatrix},\quad F=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ u(E+p)\end{Bmatrix},\quad G=\begin{Bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ v(E+p)\end{Bmatrix},\quad H=\begin{Bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ w(E+p)\end{Bmatrix} \qquad (3.39)$$

Here ρ is the density, *u*, *v,* and *w* are the velocity components in *x*, *y* and *z* directions, *p* is the pressure, and *E* is the total energy. The pressure is related to the total energy by

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right), \tag{3.40}$$

with ratio of specific heats γ=1.4. This is an idealized problem for the Euler equations in 3D. The mean flow is $\{\rho, u, v, w, p\} = \{1,1,1,0,1\}$. An isotropic vortex is then added to the mean flow, i.e. with perturbations in *u*, *v*, and temperature $T = p/\rho$, and no perturbation in entropy $S=p/\rho^\gamma$:

$$(\delta u, \delta v, \delta w) = \frac{\varepsilon}{2\pi} e^{0.5\left(1-r^2\right)}\left(-\bar{y}, \bar{x}, 0\right), \quad \delta T = -\frac{(\gamma - 1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad \delta S = 0,$$

where $r^2 = \bar{x}^2 + \bar{y}^2$, $\bar{x} = x - 5$, $\bar{y} = y - 5$, and the vortex strength ε= 5. If the computational domain extends to infinity, the exact solution of the Euler equations with the above initial conditions is just the passive convection of the isotropic vortex with mean velocity (1, 1, 0). In the following accuracy study, the computational domain is taken to be [0, 10] x [0, 10] x [0, 10] with characteristic inflow and outflow boundary conditions imposed on the boundaries. The numerical solution is carried out until *t* = 2 on regular grids generated from uniform hexahedral grids, whereby each hexahedral cell is split into six tetrahedra. No limiters are employed in this study since the problem is smooth, and the Rusanov flux is used in all simulations. In Table 10, we present the L_1 and L_∞ errors and orders in the CV-averaged density using the QF approach. The formal order of the scheme is essentially verified, with about a half order drop for 3rd order. In order for convergence to the formal order of accuracy, it may be necessary to use finer grids for this case. Contours of Mach number and some streamlines for this case are shown in Figure 14.

Table 10. Accuracy of 3D vortex propagation problem at *t*=2 on regular grids.

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
2	10x10x10x6	6.07e-3	-	1.16e-1	-
	20x20x20x6	1.56e-3	1.96	3.98e-2	1.54
	40x40x40x6	3.92e-4	1.99	9.82e-3	2.02
3	10x10x10x6	1.34e-3	-	6.03e-2	-
	20x20x20x6	2.23e-4	2.59	6.33e-3	3.25
	40x40x40x6	3.94e-5	2.50	1.07e-3	2.56

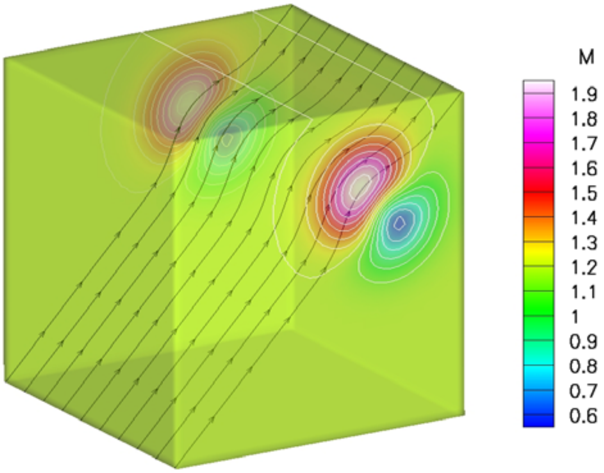


Figure 14. Contours of Mach number and some streamlines for 3D vortex propagation case at *t*=2.

3.4.5. Zalesak's disk rotation problem

The rigid body rotation of Zalesak's disk (notched circle in 2D) under a constant vorticity velocity field [59] is employed to demonstrate the ability of the limiter to preserve discontinuities. The computational domain is taken to be [0, 1] x [0, 1], and the notched circle centers at (0.5, 0.75) with radius 0.15 and the slot has a width of 0.05 and length of 0.2479. The velocity field is given by:

$$\begin{aligned} u &= 0.5 - y \\ v &= x - 0.5 \end{aligned} . \tag{3.41}$$

The conserved variable is initialized to +1 inside the notched circle, and -1 everywhere else. The numerical solution is carried out until *t*=2π, which constitutes one full revolution of the disk. Figure 15 shows results for 2nd and 3rd order QF schemes on regular 80x80x2 and 160x160x2 grids.

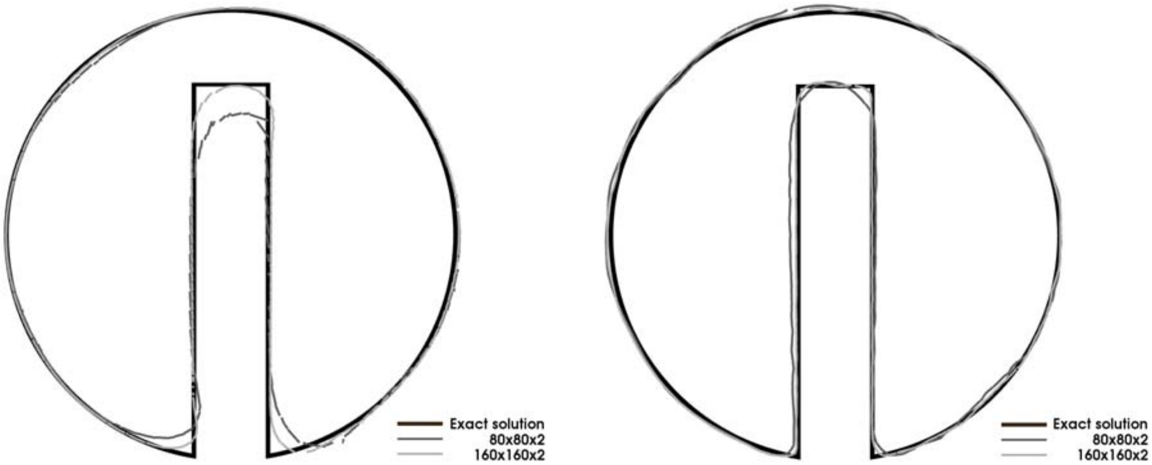


Figure 15. Zero contour for Zalesak's disk rotation problem. (a) 2nd order; (b) 3rd order.

It is evident that both the 2nd and 3rd order schemes capture the sharp features of the disk reasonably well. The 3rd order SV scheme obviously captures the discontinuity with higher resolution than the second order scheme, as expected.

3.4.6. Mach 3 wind tunnel with a step

This problem was studied extensively by Woodward and Colella [56], and has been widely used to assess the performance of shock-capturing methods. The 2D wind tunnel is 3 units long and 1 unit wide, with a step of 0.2 units high located at 0.6 units from the tunnel inlet. The initial condition is a Mach 3 right-going uniform flow. Inviscid wall boundary conditions (reflective) are used for tunnel wall boundaries, while inflow and outflow boundary conditions are used at the inlet and exit of the wind tunnel. It is well known that the corner of the step is a singularity, and often leads to a spurious Mach stem at the downstream bottom wall, and an erroneous entropy layer at the bottom wall. In [56], various numerical treatments were used to remedy these artifacts. In the present study, no special treatments were used for the singularity to see how the singularity affects the numerical solutions.

It was shown in [53] that the 4th order partition in Figure 2c produces large errors for this case, due likely to the nonuniform sub-cell mesh, so this study will focus on the 2nd and 3rd order partitions shown in Figure 2a and Figure 2b, respectively. Two different grids are considered here, consisting of 8,746 and 37,146 triangles, the coarsest of which is shown in Figure 16.

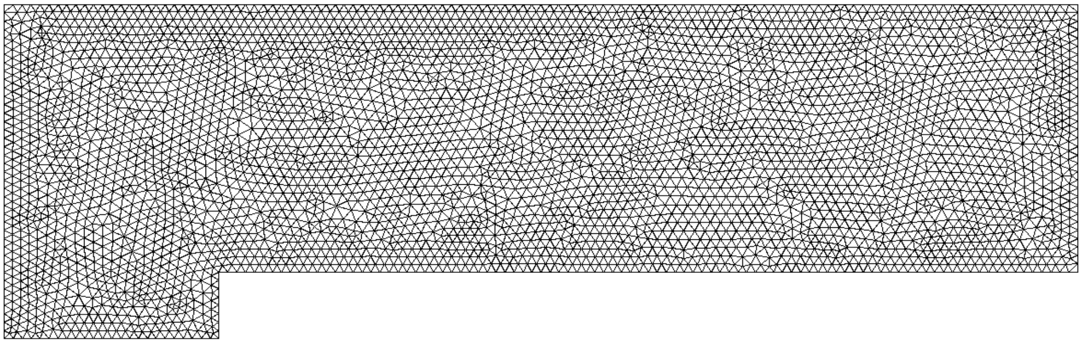


Figure 16. A coarse mesh for the supersonic wind tunnel problem with 8,746 triangles.

Thus, the number of degrees-of-freedom for the 2nd order case is 26,238 and 111,438, respectively, and the number of degrees-of-freedom for the 3rd order case is 52,476 and 222,876, respectively. All simulations are carried out until t=4. The results obtained using the QF approach with Rusanov flux are nearly identical to those obtained using the PQ approach, so only the QF results with Rusanov flux are shown here. The node-neighbor stencil for data limiting is used for all cases here, as the face-neighbor stencil causes too many cells to be limited, essentially reducing the schemes to low order.

The computed density contours for the 2nd and 3rd order simulations on all grids are shown in Figure 17 and Figure 18, respectively. Obviously, the resolution of the 3rd order results is better than that of the 2nd order results, while the 3rd order results are slightly more oscillatory than the 2nd order results.

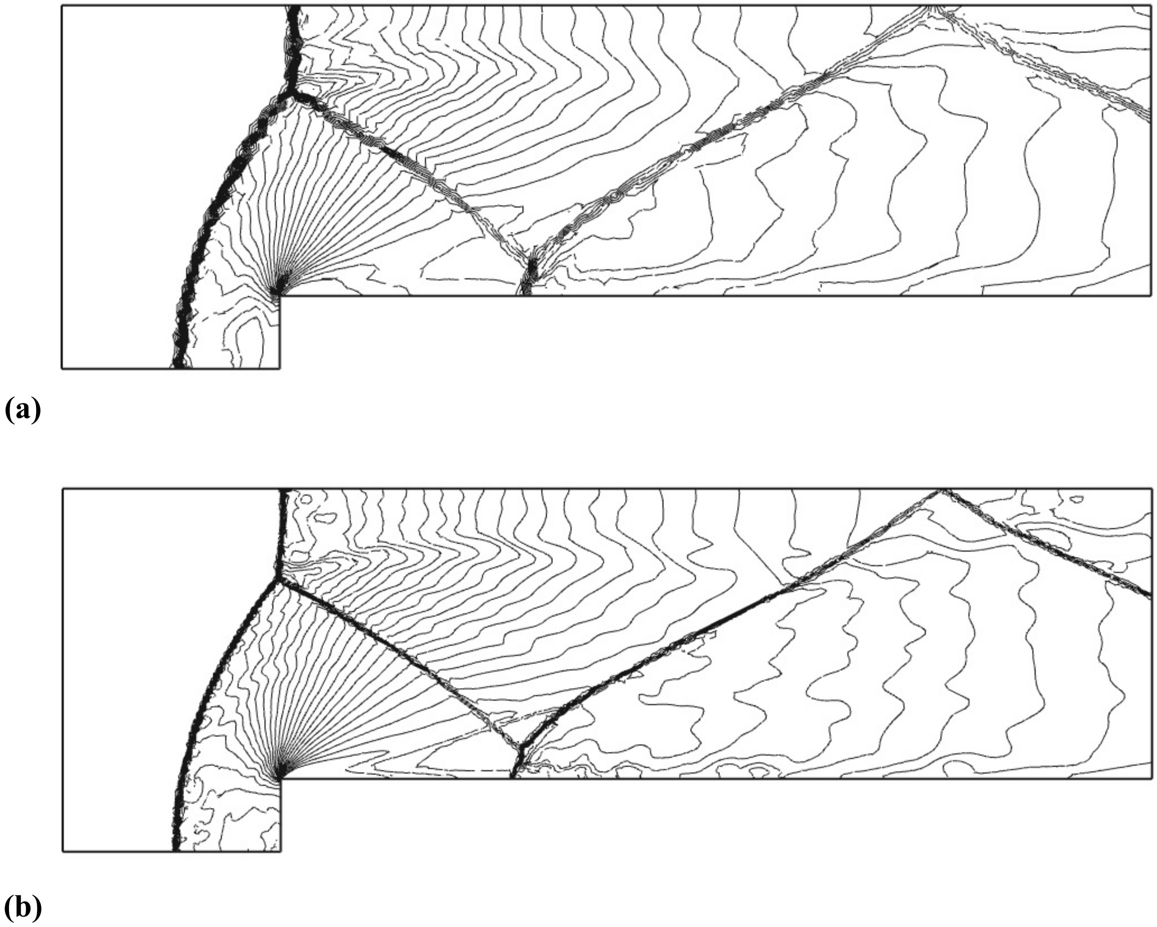


Figure 17. Density contours for the 2nd order scheme. Thirty even contour lines between 0.09 and 4.53; (a) 26,238 DOFs; (b) 111,438 DOFs.

This behavior is consistent with the results found in [53]. For example, the spurious Mach stems in the 3rd order results are considerably weaker than those in the 2nd order results. Also, the instability generated by the slip line begins to become visible in the 3rd order results on the fine mesh. It should be noted that the resolution of the results in [53] is slightly better than those presented here due to the presence of sub-cell resolution in the limiter. Never-the-less, the results presented here show similar trends for the density contours, and the strength of the spurious Mach stem is dramatically reduced under h-refinement.

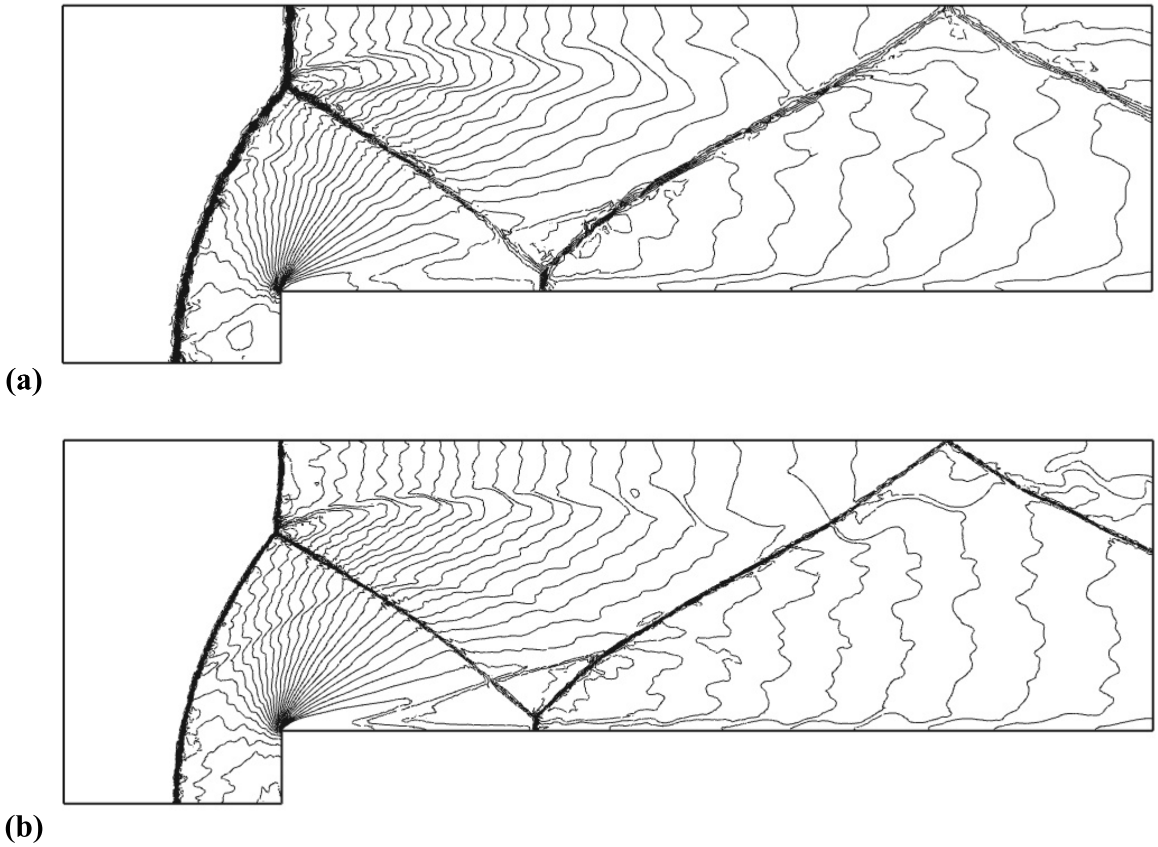


Figure 18. Density contours for the 3rd order scheme. Thirty even contour lines between 0.09 and 4.53; (a) 52,476 DOFs; (b) 222,876 DOFs.

3.4.7. Subsonic flow over a circular cylinder

A subsonic inviscid flow over a circular cylinder at Mach = 0.3 is selected to test the curved boundary representation. Due to flow symmetry, only the top half of the physical domain is selected as the computational domain although the complete physical domain is displayed in all of the figures. The far field boundary is located 20 diameters away from the cylinder center and the characteristic inflow/outflow boundary conditions are used at the far field boundary. In order to perform a grid refinement accuracy study, three different triangular grids were generated from three structured grids with 32x8, 64x16 and 128x32 cells, which are shown in Figure 19.

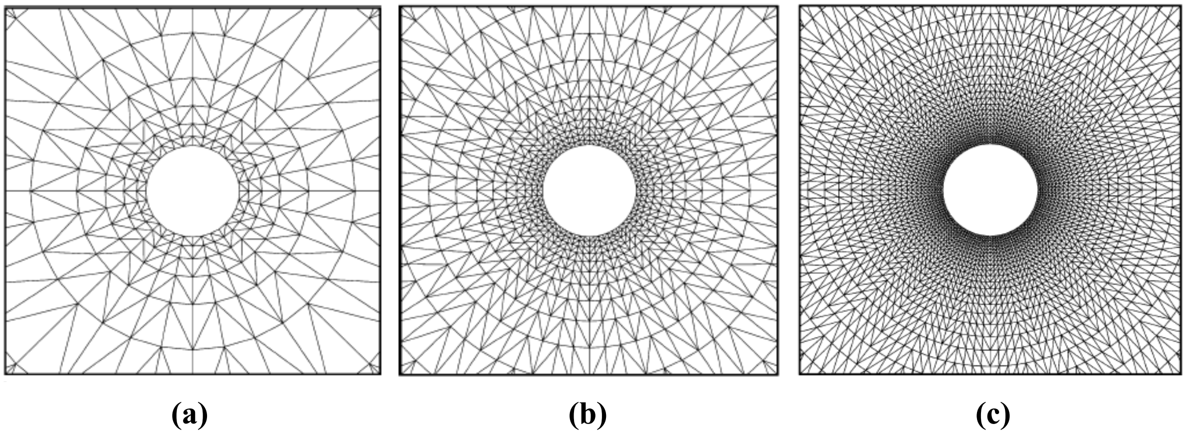


Figure 19. Computational grids for flow over a circular cylinder; (a) 32x8x2 cells; (b) 64x16x2 cells; (c) 128x32x2 cells.

The solution is initialized to a uniform free stream. All simulations employ the simplified curved boundary treatment of Krivodonova and Berger [31] outlined in Section 3.3.3. The simplified approach of Section 3.3.2 gives results that are very similar to the approach of Section 3.3.3, but we advocate the latter approach because of its efficiency and ease of implementation. The simulations were also run using a piecewise-linear boundary representation, but the solution quickly diverged in all cases. Also, even with the curved boundary treatment, the use of the Rusanov flux in the QF approach can lead to instabilities due to spurious entropy production at the trailing edge. This is overcome by the use of either the Roe [35], or HLLC [25,41,42,6] flux.

Figure 20 and Figure 22 show results for the PQ approach and the HLLC flux for both 3rd and 4th order simulations, respectively, and Figure 21 and Figure 23 show results for the QF approach and the Roe flux for both 3rd and 4th order simulations, respectively.

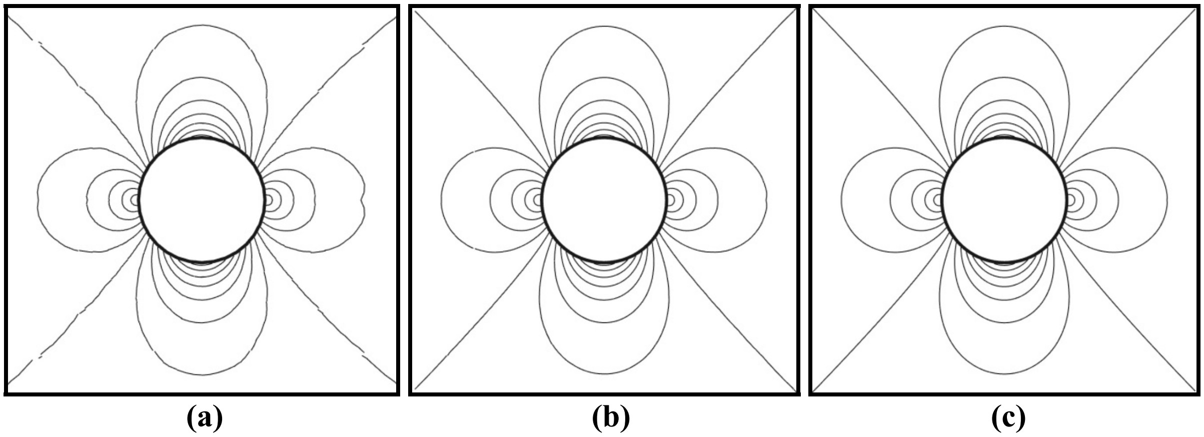


Figure 20. Mach contours computed with 3rd order PQ approach and HLLC flux. (a) 3,072 DOFs. (b) 12,288 DOFs. (c) 49,152 DOFs.

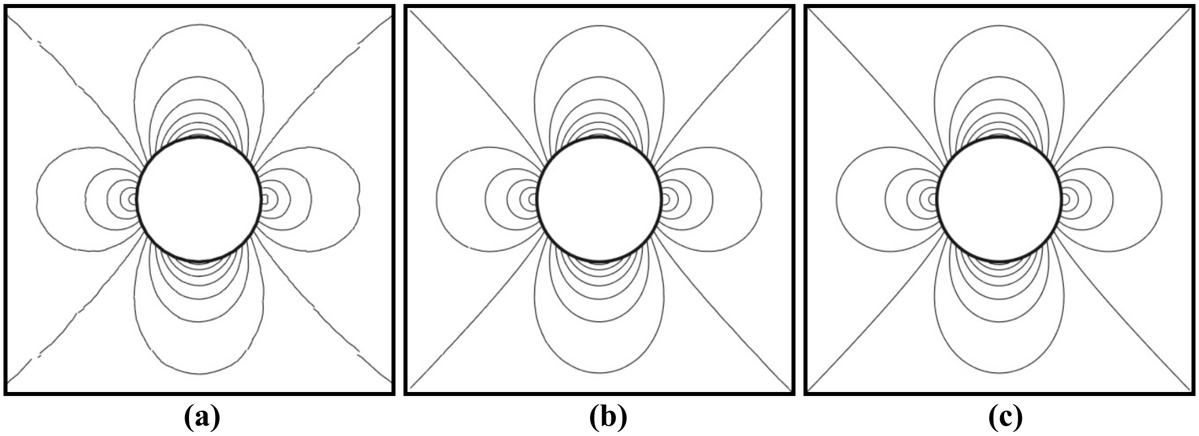
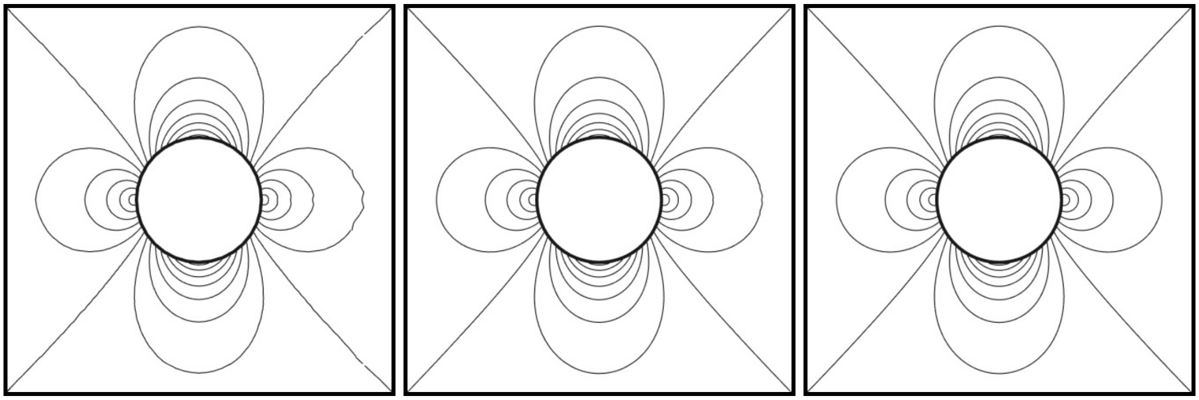


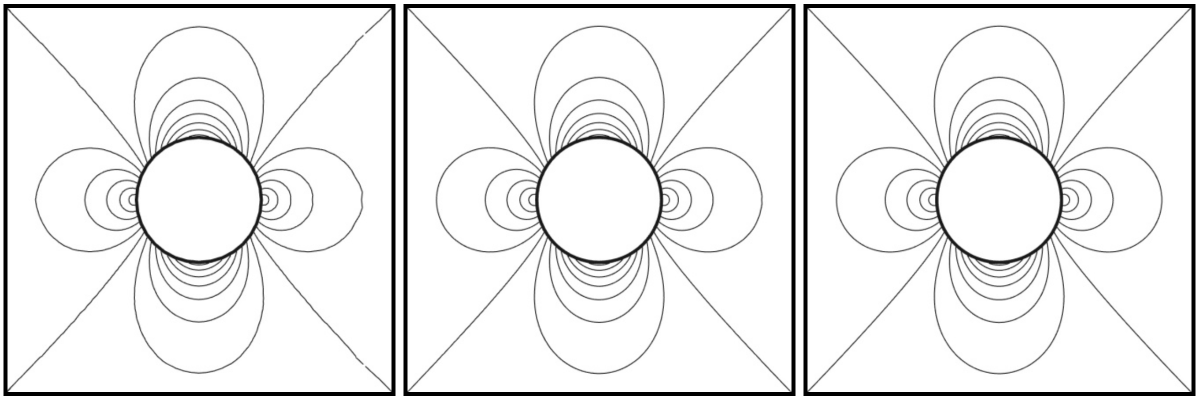
Figure 21. Mach contours computed with 3rd order QF approach and Roe flux. (a) 3,072 DOFs. (b) 12,288 DOFs. (c) 49,152 DOFs.

All plots show fourteen even contours of Mach number between 0 and 0.65. The coarse grids show some discontinuous contours and a slight asymmetry at the trailing edge. As the grid is refined, the contours become continuous and the flow becomes perfectly symmetric with respect to both axes. For the 3rd order case, grid independence is achieved on the medium grid, while for the 4th order case, even the coarse grid results are nearly grid independent. It is interesting to note that these results appear to be slightly better than those obtained using the full isoparametric SV approach on the same grids [54].



(a) **(b)** **(c)**

Figure 22. Mach contours computed with 4th order PQ approach and HLLC flux. (a) 5,120 DOFs. (b) 20,480 DOFs. (c) 81,920 DOFs.



(a) **(b)** **(c)**

Figure 23. Mach contours computed with 4th order QF approach and Roe flux. (a) 5,120 DOFs. (b) 20,480 DOFs. (c) 81,920 DOFs.

3.4.8. Subsonic flow over a NACA 0012 airfoil

As a demonstration using a more realistic geometry, inviscid subsonic flow over a NACA 0012 airfoil at Mach = 0.4, and angle of attack of 5 degrees is simulated. In this simulation, the 3rd and 4th order schemes are tested using a very coarse mesh with 48x16x2 triangles, as shown in Figure 24, to further demonstrate the effectiveness of the Krivodonova and Berger approach [31] for curved boundaries. The outer boundary is 20 chords away from the center of the airfoil.

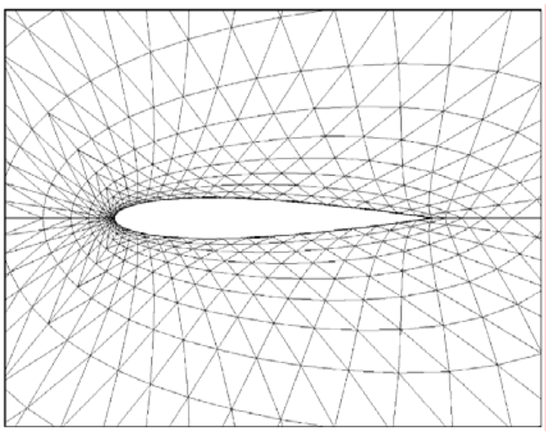
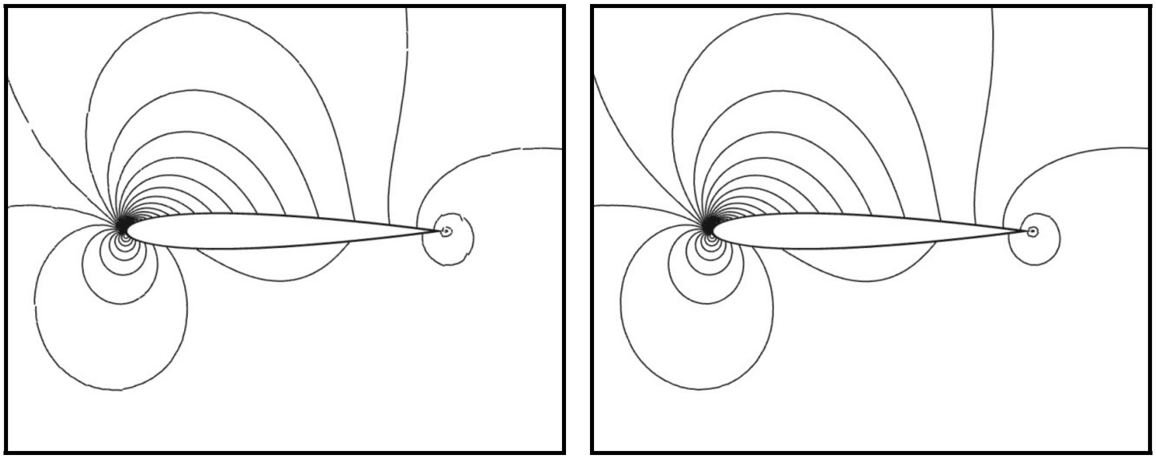


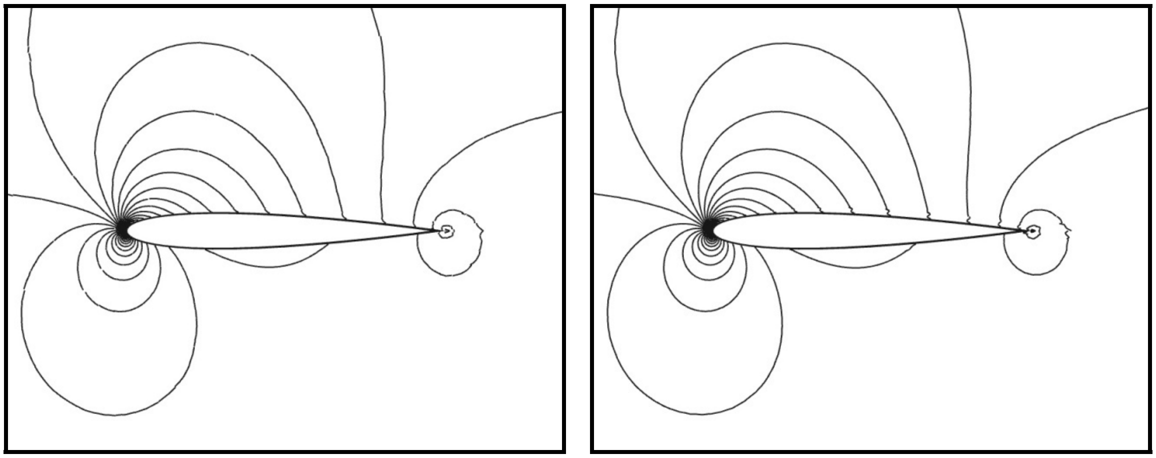
Figure 24. Computational grid for flow over a NACA 0012 airfoil (48x16x2 triangles).

Thirty-three even pressure contours between 0.5 and 0.8 computed using both a linear and quadratic boundary representation are shown in Figure 25a and Figure 25b, respectively. Figure 26a and Figure 26b show thirty-three even contours of Mach number between 0 and 0.8. All results are for the QF approach and Roe flux. As can be seen in Figure 25, the pressure contours become very smooth and continuous under p-refinement. Some spurious entropy production has caused a slight irregularity in the Mach contours at the airfoil surface as shown in Figure 26, which can most likely be attributed to the extreme coarseness of the grid.



(a) (b)

Figure 25. Pressure contours for QF approach and Roe flux: (a) 3rd order; (b) 4th order.

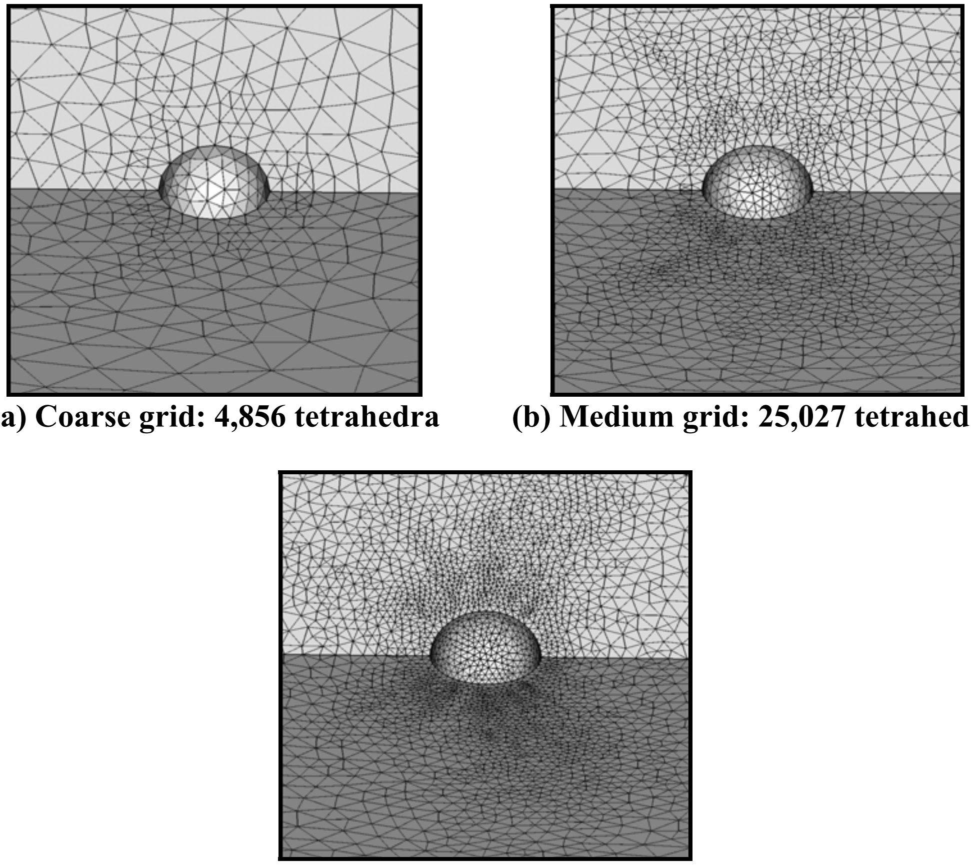


(a) (b)

Figure 26. Mach contours for QF approach and Roe flux: (a) 3rd order; (b) 4th order.

3.4.9. Subsonic flow over a sphere

In this case, we consider an inviscid Mach 0.3 flow over a sphere. We employ a sequence of 3 fully unstructured grids as shown in Figure 27.

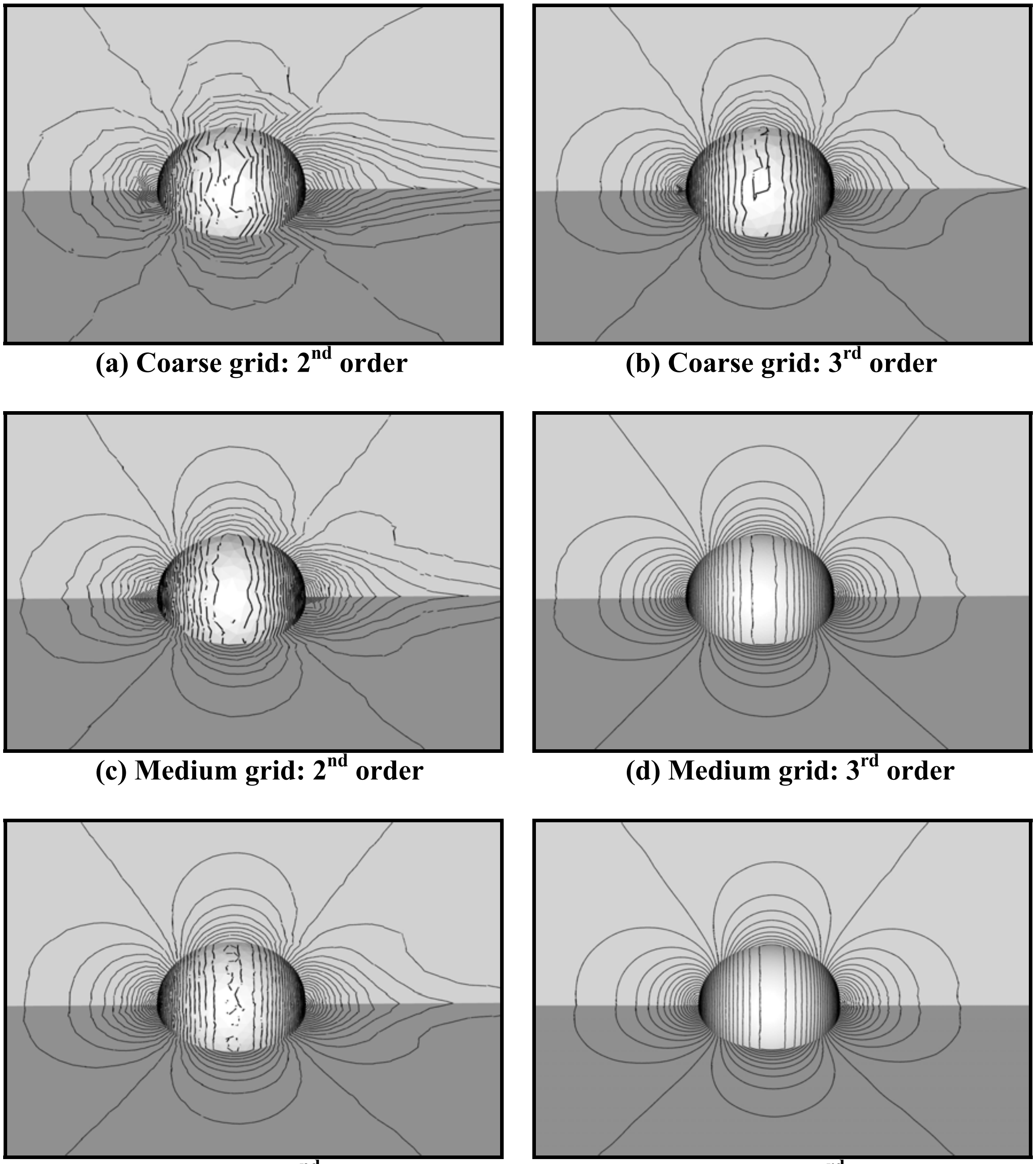


(a) Coarse grid: 4,856 tetrahedra **(b) Medium grid: 25,027 tetrahedra**

(c) Fine grid: 53,520 tetrahedra

Figure 27. Fully unstructured grids for subsonic inviscid flow over a sphere.

The curved boundary is again represented using the simplified approach of Krivodonova and Berger [31]. Figure 28 shows contours of Mach number for 2nd-order and 3rd-order simulations. It is apparent from the coarse grid results that there is a significant artificial wake region behind the sphere, which can be partially attributed to the coarseness of the grid on the sphere surface. This phenomenon is far more profound in the 2nd-order results, as shown in Figure 28a. As the grid is refined, however, the artificial wake completely disappears for the 3rd-order case, while it is still very much present for the 2nd-order case. This demonstrates a need for high-order accurate simulations when solving problems involving boundaries of high curvature. Although this artificial wake problem is not as significant when using low-order methods, such as the k-exact finite volume method [4,14], it is still a problem that does not completely vanish under moderate grid refinement [54,5].



(a) Coarse grid: 2nd order

(b) Coarse grid: 3rd order

(c) Medium grid: 2nd order

(d) Medium grid: 3rd order

(e) Fine grid: 2nd order

(f) Fine grid: 3rd order

Figure 28. Contours of Mach number for subsonic inviscid flow over a sphere.

3.5 Conclusions

An efficient quadrature-free implementation of the spectral volume method has been successfully carried out for scalar and Euler equations in both 2D and 3D. Two different approaches to compute the flux integrals more efficiently, the partial quadrature (PQ) and quadrature free (QF) approaches, have been developed and evaluated for both 3rd and 4th-order SV schemes. For a 3rd order SV scheme in 2D, the PQ and QF approaches require 22% and 72% fewer flux calculations than the traditional approach, respectively, and for a 4th order SV scheme in 2D, the PQ and QF approaches require 28% and 72% fewer flux calculations than the traditional approach, respectively. The savings is obviously significant in 2D, but in 3D the hundreds or thousands of flux calculations per SV required in the traditional approach are reduced to only dozens. For all approaches, it has been found that the nearly optimum order of accuracy can be obtained in both the L_1 and L_∞ norms with respect to density errors. A total variation diminishing (TVD) limiter has been implemented in both the PQ and QF framework, and demonstrated for a 2D unsteady supersonic problem with strong shocks. The simplified curved boundary treatment of Krivodonova and Berger, which significantly reduces complexity of the numerical implementation, has been implemented for the QF framework and successfully employed for both 2D and 3D simulations. The 3rd and 4th order schemes using the QF approach and Roe flux have been implemented for the case of subsonic flow around a NACA 0012 airfoil to further demonstrate the effectiveness of the simplified high-order boundary treatment. The linear boundary representation was also used with 3rd and 4th-order SV schemes for the case of subsonic flow over a cylinder, but we were only able to obtain convergent numerical solutions using the high-order boundary treatment for this case. To validate the Krivodonova and Berger approach for 3D simulations, the problem of subsonic inviscid flow over a sphere was successfully simulated. It was shown that for this case, the 2nd-order simulations were plagued by an artificial wake region behind the sphere, which was much weaker and ultimately disappeared under grid refinement for the 3rd-order simulation.

CHAPTER 4. LOCAL ADAPTIVE HP-REFINENT FOR SV METHOD

The high-order quadrature-free spectral volume (SV) method is now extended to handle local adaptive hp-refinement (grid and order refinement). This methodology is also described in Harris et al. [20,21]. An edge-based adaptation methodology utilizing a binary tree search algorithm is employed for high efficiency. An adaptation criteria is selected which focuses computational effort near discontinuities, and effectively reduces the physical area of the domain necessitating data limiting for stability. This makes the method very well suited for capturing and preserving discontinuities with high resolution. Since the accuracy of the SV method is heavily dependent on both the grid and the degree of polynomial interpolation, both h- and p- refinement are presented in a general framework where it is possible to perform either or both on any grid cell at any time. Several well-known inviscid flow test cases, subjected to various levels of adaptation, are utilized to demonstrate the effectiveness of the method.

4.1 Overview of local adaptive hp-refinement

Adaptive hp-refinement is used to locally refine or coarsen both the grid and solution polynomial as the flow develops. It is often desirable to refine the grid near discontinuities, and to refine the polynomial near fine smooth features to avoid the high computational cost of global refinement. This is the focus of the current work.

4.2 Framework for local adaptive hp-refinement

Local adaptive grid refinement is used to focus computational effort near discontinuities and fine smooth features to reduce the overall computational effort in the entire domain. H-refinement involves modification of cell sizes while p-refinement involves modification of polynomial orders. We wish to utilize either or both on-the-fly as the flow develops. Both h- and p-refinements are carried out using only local operations to maximize the efficiency and accuracy of the procedure.

4.2.1 h-refinement

In this study, since we are only dealing with triangular SVs, the h-refinement can be performed without introducing the so-called hanging nodes. Therefore, it is basically a matter of grid regeneration, with no required modification of the solver itself. An efficient hierarchical edge-based adaptation algorithm is employed, which allows the grid to be adapted any number of levels from the base (coarsest) grid at any time. Let ε_i be an error indicator for edge i, and let ε_{max} be some norm of ε_i taking into account the error for all edges in the domain. If for any edge i, $\varepsilon_i>\alpha\varepsilon_{max}$, then edge i is split into two edges, otherwise edge i is maintained. Here $\alpha>0$ is a user specified constant. This procedure begins with edges in the base grid (root edges), and continues until the maximum number of adaptation levels is reached. The error indicator ε_i is always computed using the solution from the previously adapted (finest) grid. When the above procedure completes, new SVs are added to the grid as a result of the split edges. There are essentially four different situations that can occur when the grid is adapted, as shown in Figure 29.

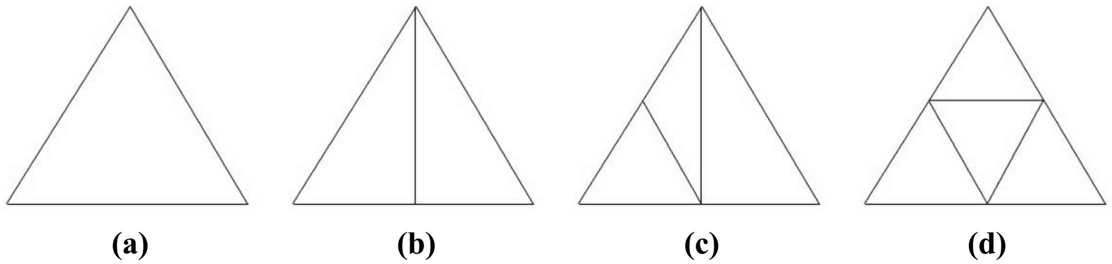


Figure 29. Four situations that can occur when a SV is refined; (a) No edges are split so the SV is unchanged; (b) Two new SVs are generated due to one split edge; (c) Three new SVs are generated due to two split edges; (d) Four new SVs are generated due to three split edges.

For each SV in the grid, the difference in adaptation level for that SV's edges is allowed to be no greater than one. This is done to ensure that all grids are comparable in quality to the base grid. When the creation of new SVs is complete, new cell-averages are then computed using

$$\overline{Q}_{i,j} = \sum_{k=1}^{N_S} \overline{M}_{j,k} Q_{i,k} \ , \tag{4.1}$$

where $\overline{M}_{j,k}$ are the node-based shape functions for node k averaged over CV j, and $Q_{i,k}$ are the conserved variables evaluated at node k of SV i. If node k exists within a SV in the previously adapted grid, then $Q_{i,k}$ are obtained from (2.6) using CV-averaged solutions $\overline{Q}_{i,j}$ from that SV. Otherwise, if node k exists at the junction between two or more SVs in the previously adapted grid, then $Q_{i,k}$ are obtained from an average of (2.6) among all SVs which have the physical location of node k in common. The above interpolation gives rise to an inherent loss of precision associated with coarsening of the solution, which is an unavoidable consequence of the h-refinement procedure. Three different methods for computing the error indicator ε_i are given below. The first, and simplest error indicator is computed using

$$\varepsilon_i = \left|\Delta\psi_i\right| A_i^u , \tag{4.2}$$

where $\Delta\psi_i$ is the difference of ψ (a variable of interest) between the two endpoints of edge i, A_i is the area of edge i, and u>0 is a user specified parameter. An alternative gradient-based error indicator is computed using

$$\varepsilon_i = \left|\Delta\left(\nabla\psi\cdot\vec{l}\right)_i\right| A_i^u , \tag{4.3}$$

where $\Delta\left(\nabla\psi\cdot\vec{l}\right)_i$ is the difference of the gradient of ψ between the two endpoints of edge i projected in the direction tangent to edge i. The final, and most expensive error indicator is computed using

$$\varepsilon_i = \int_{S_i}\left\{\left(\frac{\partial^2\psi}{\partial x^2}\right)^2 + \left(\frac{\partial^2\psi}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2\psi}{\partial y^2}\right)^2\right\} dV . \tag{4.4}$$

This is a cell-based error indicator, whereas (4.2-4.3) are edge-based error indicators. Thus, (4.4) is computed for each cell, and if a given cell is marked for refinement, the edges bounding that cell are marked for refinement. In addition, since information about 2nd derivatives is needed, (4.4) is only valid for 3rd and higher order simulations. A comparison of results from the error indicators given by (4.2-4.4) is given in Section 4.3.4. For all cases considered here, u=1/2 and ε_{max} is taken to be the L_2 norm of ε over all edges, given by

$$\varepsilon_{\max} = L_2\left\|\varepsilon\right\| = \sqrt{\frac{1}{Ne}\sum_{i=1}^{Ne}\varepsilon_i^2} , \tag{4.5}$$

where Ne is the total number of edges in the grid, or cells if (4.4) is used.

4.2.2 p-refinement

P-refinement, or order refinement, allows for a distribution of SVs where the degree of the polynomial reconstruction may vary from one SV to another. Unlike h-refinement, p-refinement does require significant modification of the solver itself. Among other things, the terms N and N_s in (2.6) and (3.1) are no longer constant, but depend on the level of p-refinement of the current cell. In addition, computation of the face-averaged terms in (3.3) and (3.4) is not as straightforward as before. For example, consider the case where a linear SV is adjacent to a quadratic SV, as shown in Figure 30.

Here, the face-averaged shape functions for CV faces on SV boundaries must be computed in parts. For the corner CVs in the quadratic partition (right), the face-averaged shape functions are computed as usual, but the face-averaged shape function for the side CV must be computed in two parts to coincide with the intersection of that CV face with the face of the adjacent CV in the linear SV. With the face-averaged shape functions computed in this manner, the face-averaged terms in (3.3) and (3.4) can be readily computed.

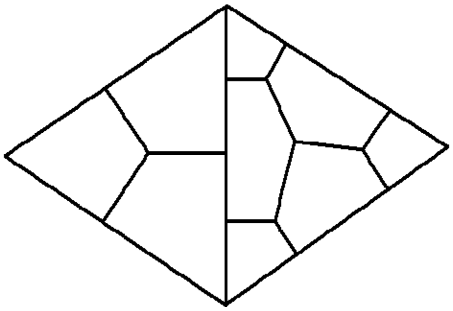


Figure 30. Two adjacent SVs with p-refinement levels differing by one. The left SV contains a linear partition, and the right SV contains a quadratic partition.

Let the error indicator ε_i for edge i be the same as is defined in (4.2), and ε_{max} is again taken to be the L_2 norm of ε over all edges. If for any edge i, $\varepsilon_i>\beta\varepsilon_{max}$, then the degree of polynomial reconstruction, for the cells adjacent to edge i, is increased by 1. Similarly if $\varepsilon_i<\gamma\varepsilon_{max}$, then the degree of polynomial reconstruction for the cells adjacent to edge i is either maintained or decreased by 1, if a lower order is available. Here $\gamma,\beta>0$ are user specified constants.

4.2.3 hp-refinement

For simplicity, simultaneous h- and p-refinements are carried out in a decoupled manner. H-refinement is first performed to generate a new grid, and p-refinement is then performed to increase or decrease the degree of the polynomial reconstruction for each SV in the new grid. As new SVs are created as a consequence of h-refinement, the polynomial degree is set to minimum (1 in this case), and may not be increased as a result of p-refinement. This is a safeguard to ensure that the lowest possible degree polynomial is used near very high gradient regions. Such a measure should minimize oscillations due to extreme flow phenomena such as shock waves. This methodology for hp-refinement should be able to tackle a wide range of problems, resolving both shock waves and fine smooth features simultaneously. This is in contrast to other recent approaches by Biswas et al. [7], Flaherty et al. [16], and Remacle et al. [34], where both h- and p-refinements are carried out in the same regions. While this approach may work well for some situations, it could lead to large oscillations for problems involving strong shock waves. This is due to the fact that there is no mechanism in place to prevent the use of a high-order polynomial near a shock wave, other than a limiting procedure.

4.3 Results

In this section, the SV method with local adaptive hp-refinement is evaluated for the 2D Euler equations. Several well known inviscid flow test cases are utilized to demonstrate the effectiveness of local hp-refinement. In all cases involving shock waves, the TVD limiter presented in Section 3.2 is employed to maintain a stable numerical scheme. In all cases involving curved-wall boundaries, the approach of Krivodonova and Berger [31] outlined in Section 3.3.3 is utilized to maintain low computational cost. All of the following cases employ the Rusanov [36] flux, and for time integration we use the 3rd order Strong Stability-Preserving [18] (SSP) Runge-Kutta scheme, as outlined in (2.11).

4.3.1 Subsonic flow over NACA 0012 airfoil

As a demonstration of the p-refinement technique, subsonic flow at Mach=0.4, and angle of attack of 5° around a NACA 0012 airfoil is considered. The grid used for the NACA 0012 case is again semi-structured, as shown in Figure 24. The outer boundary is 20 chord lengths away from the center of the airfoil. For this case, the SVs near the farfield are orders-of-magnitude larger than the SVs near the airfoil surface. Thus, to prohibit adaptation near the farfield, it is necessary to remove the area weighting in (4.2) for this case. Also, ψ in (4.2) is taken to be the Mach number.

A converged solution from a 2nd order simulation is subjected to 1 level of p-refinement and run until convergence. This case will be subsequently denoted as the 2-3 case. Thus, the resulting solution will contain some 2nd order SVs and some 3rd order SVs. Mach contours for this simulation, as well as uniform 2nd and 3rd order simulations for comparison, are shown in Figure 31.

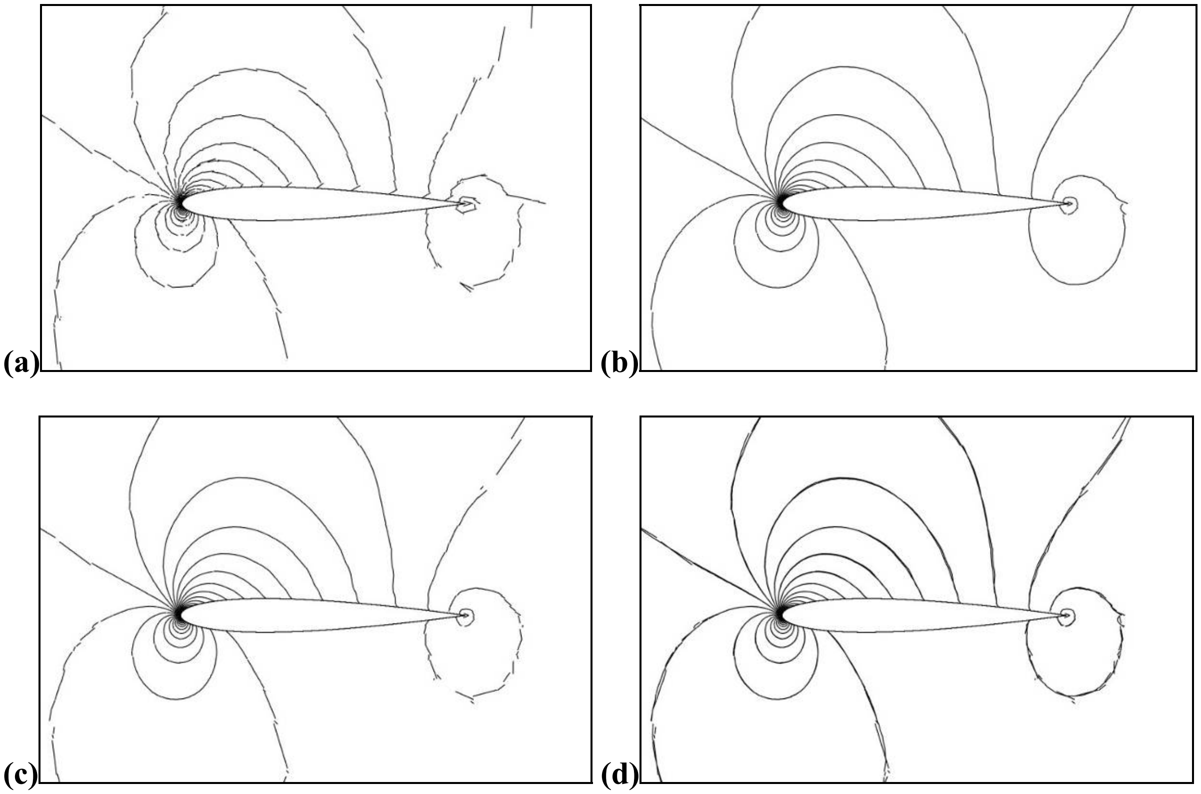
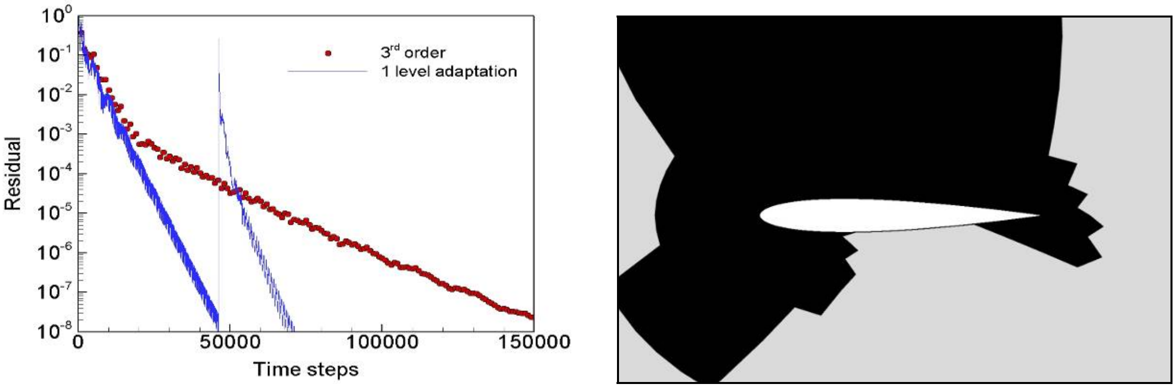


Figure 31. Contours of Mach number for subsonic flow over a NACA 0012 airfoil; (a) 2nd order (4,608 DOFs); (b) 3rd order (9,216 DOFs); (c) 1 level of p-adaptation starting from the converged 2nd order solution shown in (a) (6,519 DOFs); (d) The 1 level case shown with the 3rd-order case to illustrate differences.

Figure 31d shows Mach contours for the 2-3 case and for a uniform 3rd order case for comparison. It is evident that Mach contours for the 2-3 case agree reasonably well with the 3rd order contours, and the large errors present near the airfoil in the 2nd order case are eliminated in the 2-3 case. The convergence history for this case is shown in Figure 32a.



(a) **(b)**

Figure 32. Results for subsonic flow over NACA 0012 airfoil (a) Convergence history (The red circles are for a uniform 3rd order case, while the blue line shows the convergence of the 2nd order solution, the p-adaptation step, and the convergence of the resulting "2-3" case) (b) Schematic showing the status of p-refinement after refining from a converged 2nd order solution. Black regions represent 3rd order SVs, while gray regions represent 2nd order SVs.

It is apparent that the 2-3 case costs slightly more than the 2nd order case in terms of required time steps, but it costs much less than the 3rd order case. This is encouraging, as the 2-3 case agrees with the 3rd order case extremely well at the airfoil surface (which is where a lift/drag calculation would take place), for significantly less computational cost than that required for a full 3rd order simulation. Figure 32b clarifies which SVs are increased to 3rd order for the 2-3 case. It is clear that the majority of SVs in the domain are still 2nd order, and 3rd order SVs are only used in regions of largest change in Mach number.

4.3.2 Subsonic flow over NASA GA(W)-1 airfoil

The study in Section 4.3.1 is now repeated for the NASA GA(W)-1 airfoil shown in Figure 33. Figure 34 shows Mach contours for this case, and Figure 35 shows the convergence history and refinement region. This case uses a finer grid, so the results are not quite as discernible as with the NACA 0012 case, but none-the-less, a smoother and more accurate solution is attained for a slightly higher cost than a 2nd order simulation.

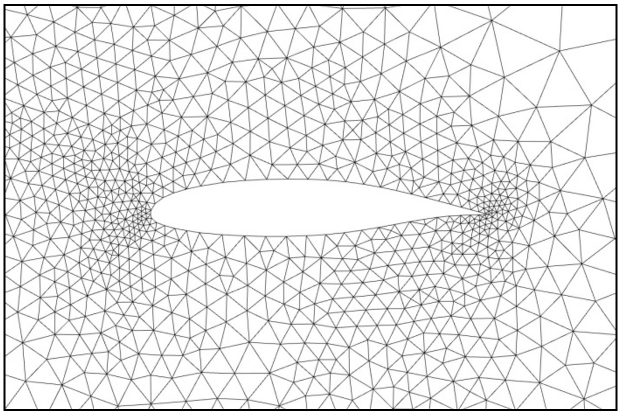
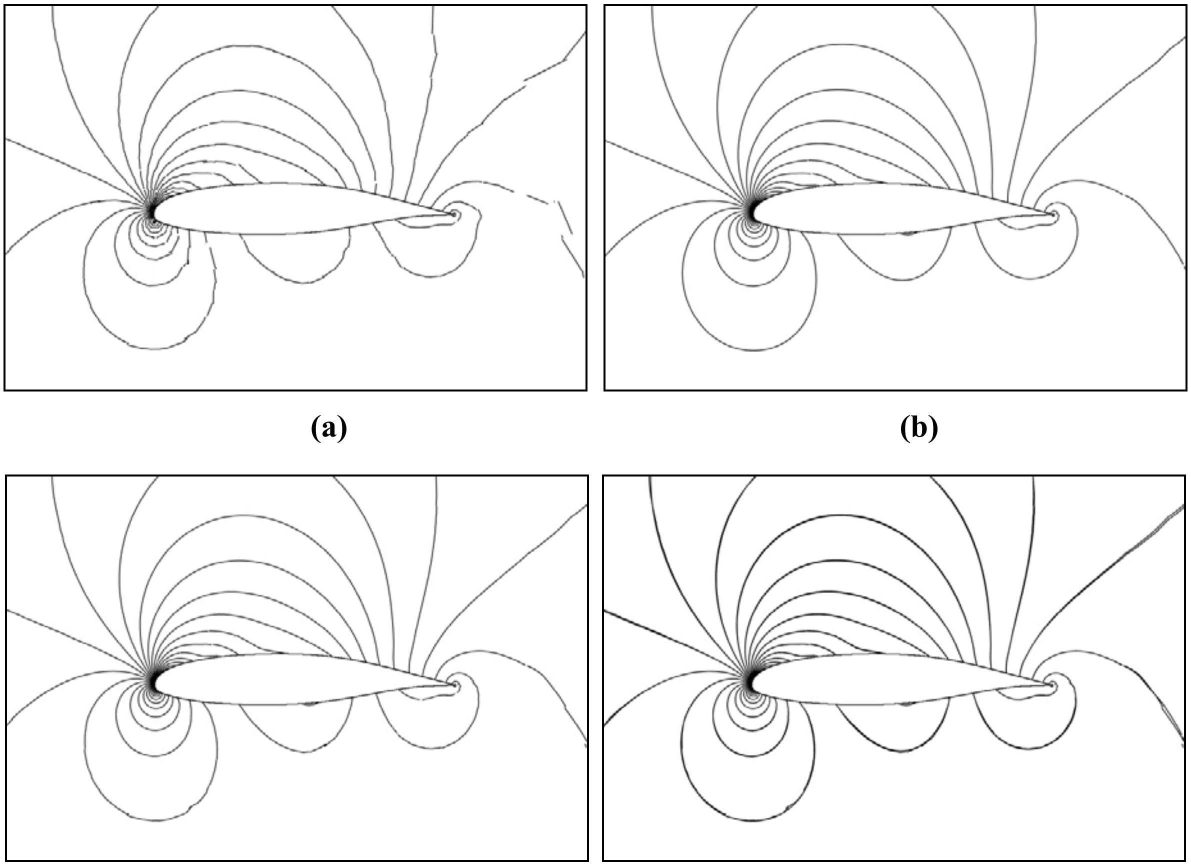


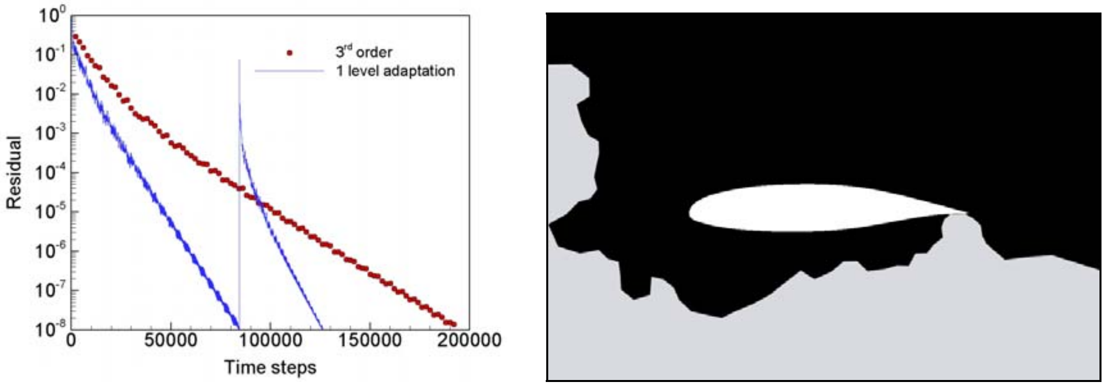
Figure 33. Irregular grid for inviscid subsonic flow over NASA GA(W)-1 airfoil (2,722 triangles).



(a) (b)

(c) (d)

Figure 34. Contours of Mach number for subsonic flow over a NASA GA(W)-1 airfoil; (a) 2nd order (8,166 DOFs); (b) 3rd order (16,332 DOFs); (c) 1 level of p-adaptation starting from the converged 2nd order solution shown in (a) (11,958 DOFs); (d) The 1 level case shown with the 3rd order case to illustrate differences.



(a) (b)

Figure 35. Results for subsonic flow over NASA GA(W)-1 airfoil (a) Convergence history (The red circles are for a uniform 3rd order case, while the blue line shows the convergence of the 2nd order solution, the p-adaptation step, and the convergence of the resulting “2-3” case) (b) Schematic showing the status of p-refinement after refining from a converged 2nd order solution. Black regions represent 3rd order SVs, while gray regions represent 2nd order SVs.

4.3.3 Transonic flow over NACA 0012 airfoil

As a demonstration of the h-refinement technique, transonic flow at Mach=0.9, and angle of attack of 1° over a NACA 0012 airfoil is considered. The base grid used for this simulation is the same as that used in Section 4.3.1. Here the error indicator (4.2) is computed based on density and total energy, and again the area weighting is removed to avoid unnecessary refinement in the farfield. This case involves shock waves on both the upper and lower surface of the airfoil, so the aforementioned TVD limiter is utilized to maintain stability. A converged 2nd order solution is again taken as the initial condition, and grid is then re-adapted 3 times and then frozen for the remainder of the simulation.

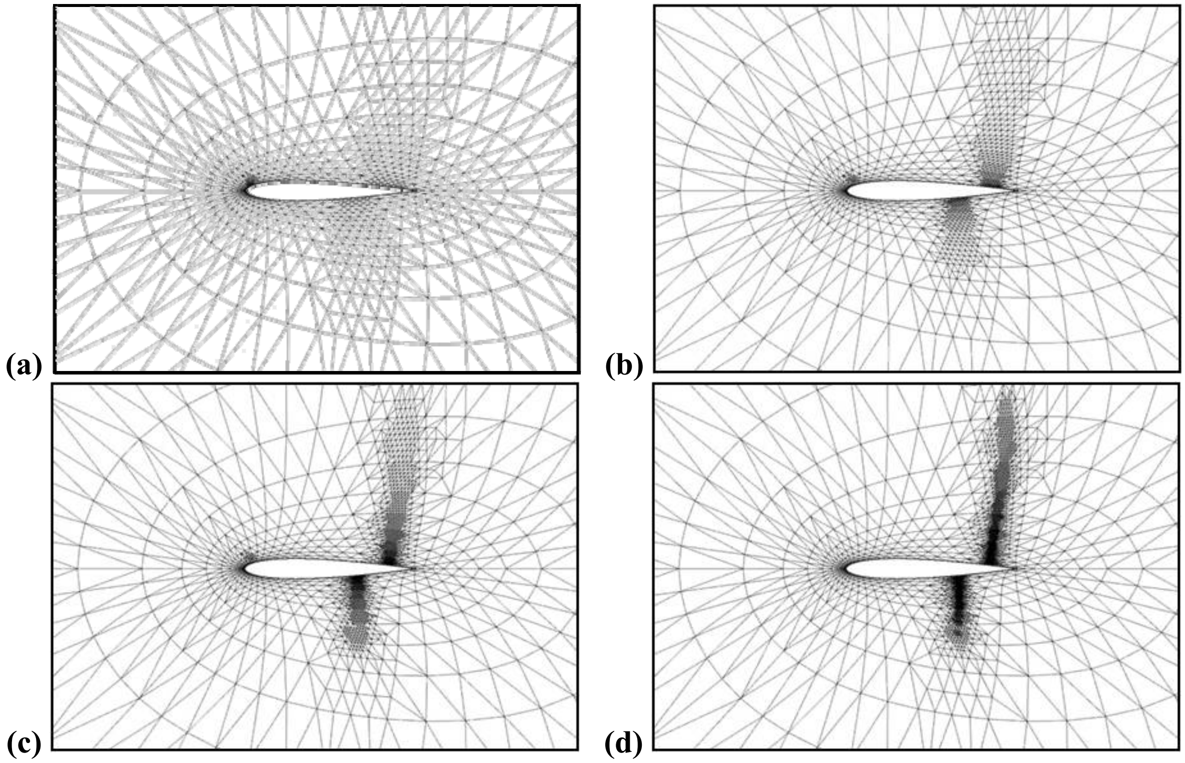


Figure 36. Grids for 2nd order solution of transonic flow over NACA 0012 airfoil with adaptive h-refinement; (a) 1 level (3,349 triangles); (b) 2 levels (9,337 triangles); (c) 3 levels (30,498 triangles); (d) 4 levels (92,551 triangles); A converged 2nd order solution on the base grid is used as the initial condition for all cases. The grid is re-adapted 3 times (once every 100 time steps for the first 300 time steps), and then frozen for the remainder of the simulation.

Figure 36 and Figure 37 show the computational grids and Mach contours, respectively, for 1, 2, 3 and 4 levels of adaptive h-refinement. It is evident that without refinement, the shock waves are smeared over several grid cells and the solution is of low quality. As the adaptation level is increased, the grid density in the vicinity of both shock waves is increased markedly. This produces a much higher quality solution with more precisely captured and finely resolved shock waves.

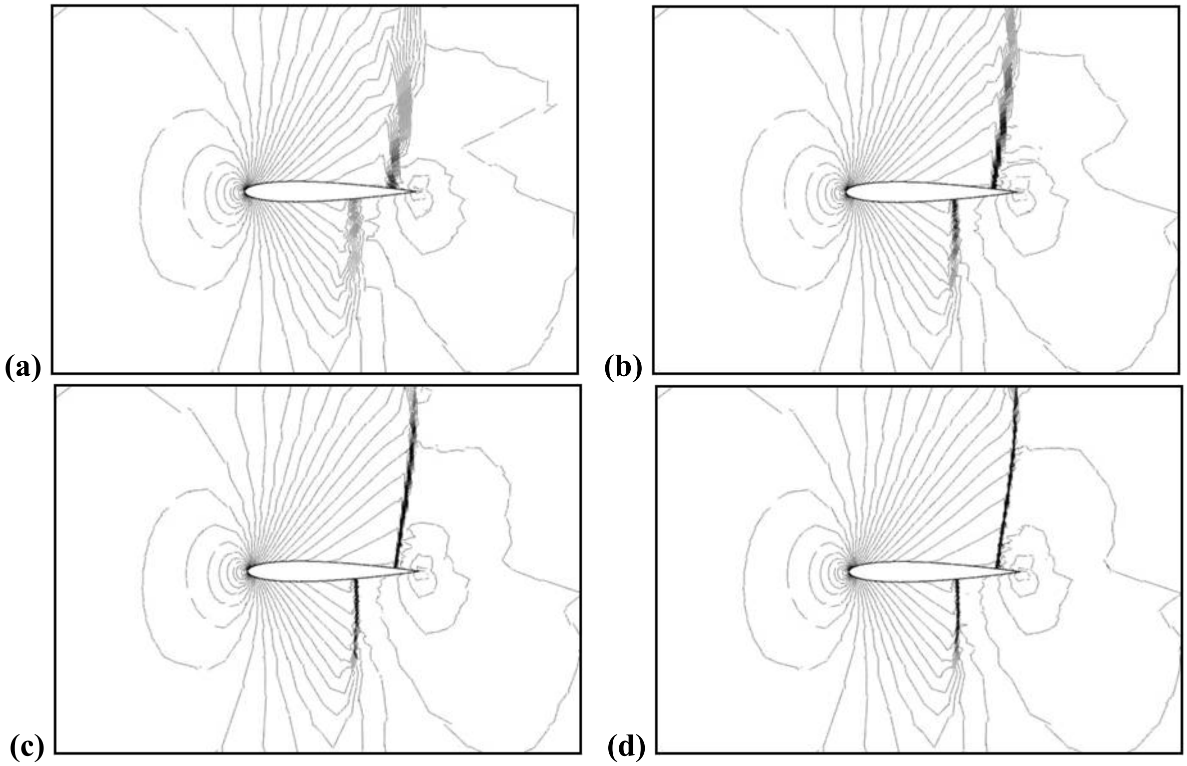
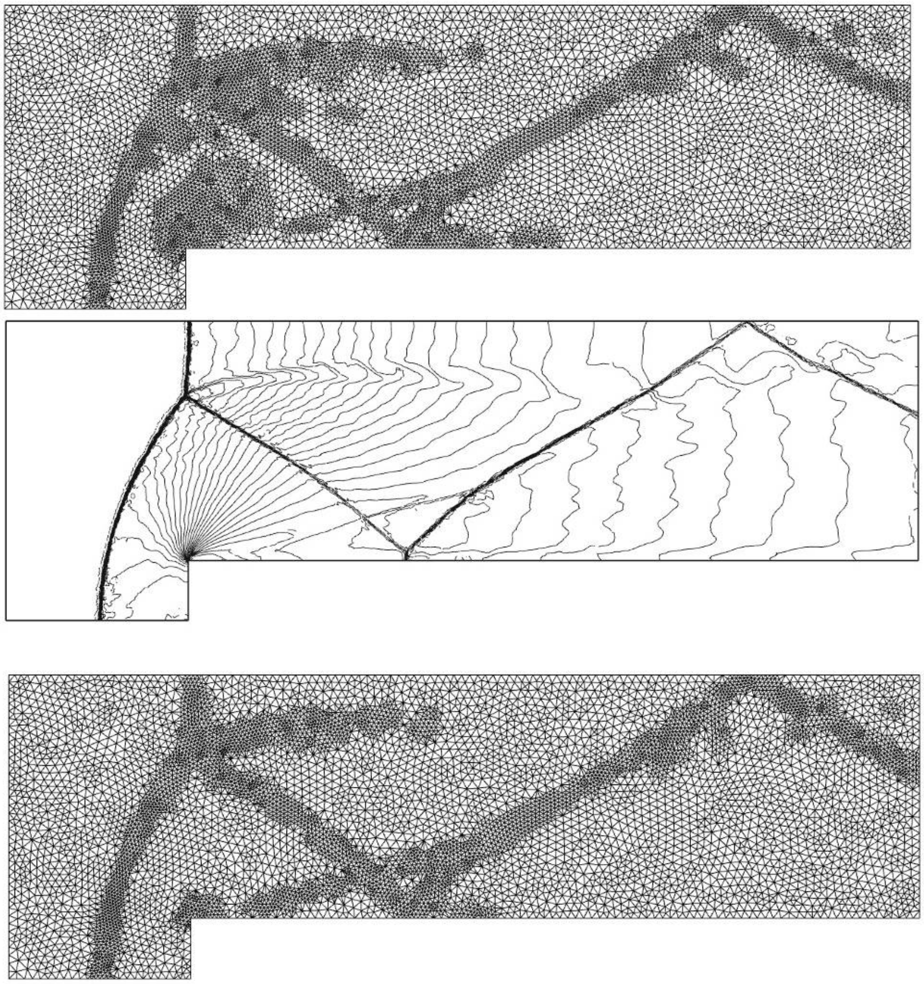


Figure 37. Mach contours for 2nd order solution of transonic flow over NACA 0012 airfoil with adaptive h-refinement; (a) 1 level (10,047 DOFs); (b) 2 levels (28,011 DOFs); (c) 3 levels (91,494 DOFs); (d) 4 levels (277,653 DOFs); A converged 2nd order solution on the base grid is used as the initial condition for all cases.

4.3.4 Mach 3 wind tunnel with a step

The Mach 3 wind tunnel with a step problem outlined in Section 3.4.6 is now repeated using local adaptive h-refinement. The base grid for this simulation is shown in Figure 16. Both 2nd and 3rd order simulations are carried out, using various levels of h-refinement. Figure 38 shows a comparison of results using the error indicators (4.2-4.4) for a 3rd order simulation with 1 level of refinement. It is apparent that the results are all very similar, while (4.3) and (4.4) are more expensive to compute than (4.2). In addition, as the number of adaptation levels is further increased, (4.3) and (4.4) become extremely sensitive, and as a result, further adaptation becomes increasingly non-isotropic. For this reason, and because it is significantly less expensive to compute while producing desirable results, we use the error indicator (4.2) which is computed based on density, and the area weighting exponent u=1/2. Figure 39 and Figure 40 show grids and density contours obtained for a 2nd order simulation with 0-3 levels of h-refinement.



(a)

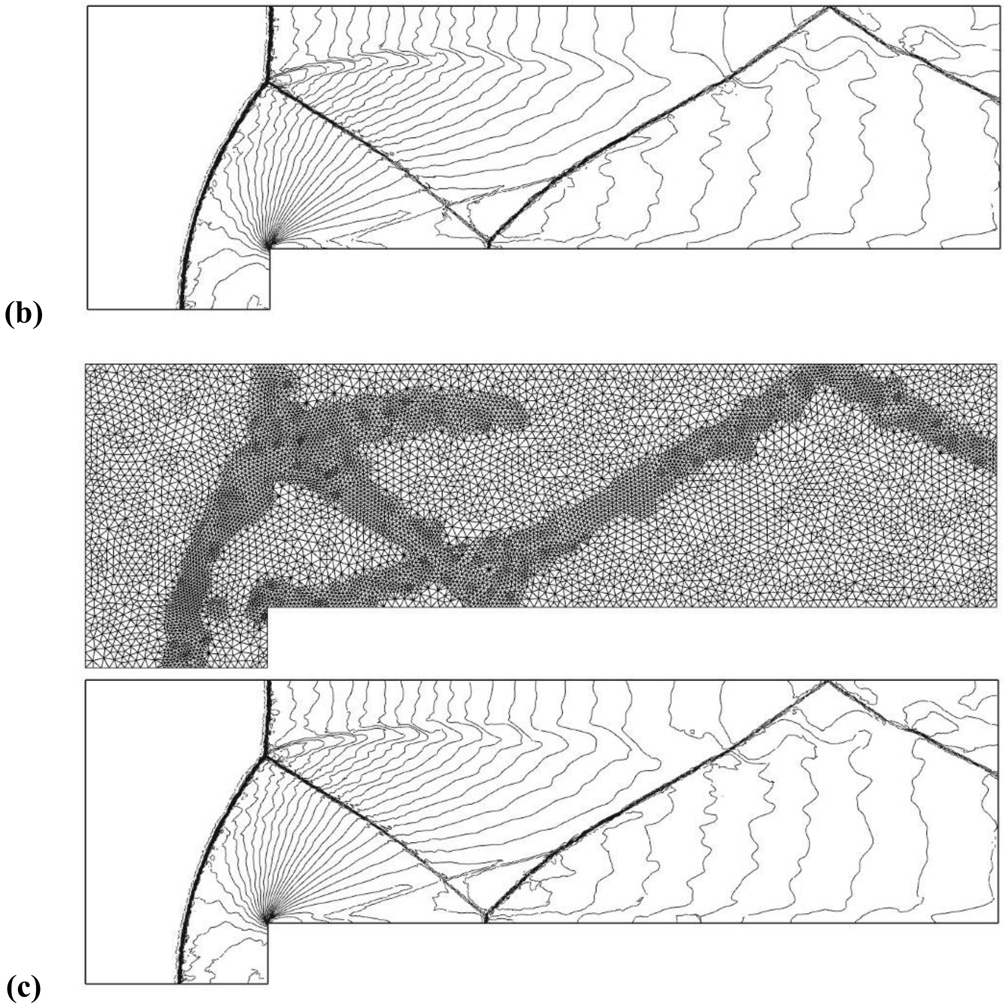


Figure 38. Grid and density contours for 3rd order SV scheme under 1 level of adaptive h-refinement at time=4.0; (a) Using error indicator given by (4.2) (89,928 DOFs); (b) Using error indicator given by (4.3) (84,228 DOFs); (c) Using error indicator given by (4.4) (95,046 DOFs); Refined from base grid every 100 time steps.

All plots show 30 even contours of density between 0.09 and 4.53. It is clear that as the adaptation level is increased, the grid becomes exceedingly dense in the vicinity of the shock wave, near the corner of the step, and downstream of the triple point. Also as the adaptation level is increased, the shock is captured more accurately with less smearing, and the spurious Mach stem downstream of the step is completely removed.

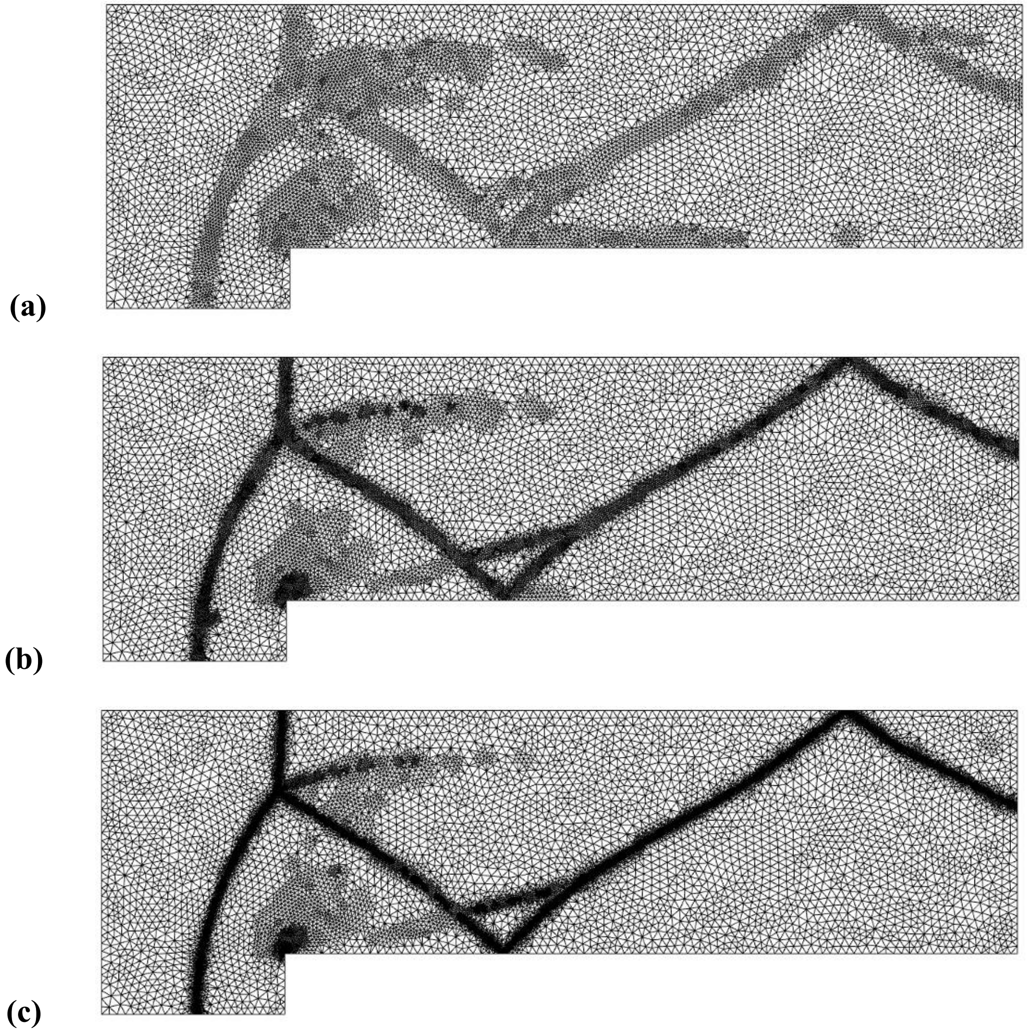


Figure 39. Grids for 2nd order SV scheme under adaptive h-refinement at time=4.0; (a) 1 level (14,765 triangles); (b) 2 levels (22,104 triangles); (c) 3 levels (35,846 triangles); Refined from base grid every 100 time steps.

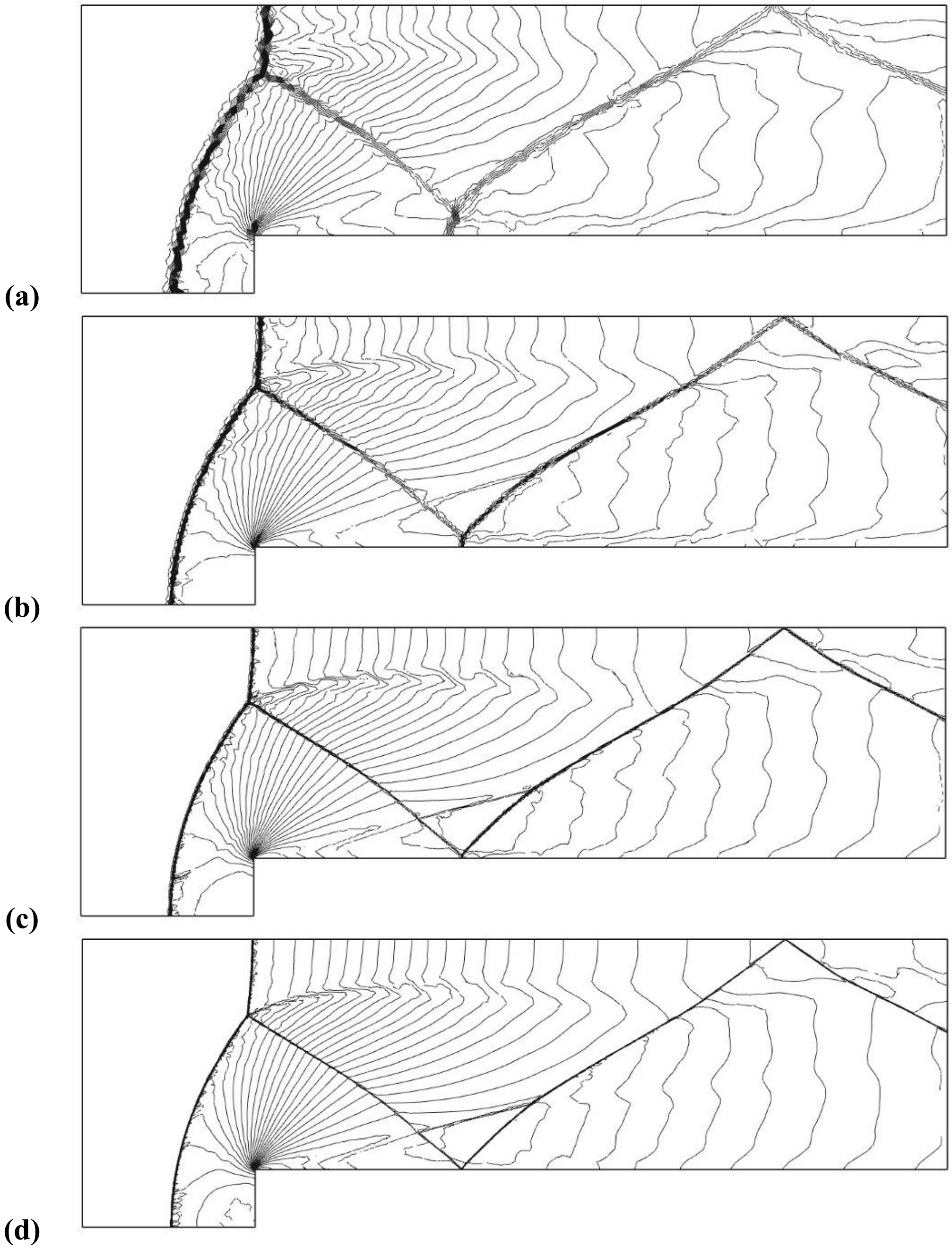


Figure 40. Density contours for 2nd order SV scheme under adaptive h-refinement at time=4.0; (a) Base grid (26,238 DOFs); (b) 1 level (44,295 DOFs); (c) 2 levels (66,312 DOFs); (d) 3 levels (107,538 DOFs); Refined from base grid every 100 time steps.

Figure 41 and Figure 42 show results for a 3rd order simulation. In both 2nd and 3rd order cases, the spurious Mach stem is completely eliminated for 2 or more levels of h-refinement, and when compared to global refinement, even 1 level of h-refinement produces a much better solution with far fewer degrees of freedom than that on a grid that has been globally refined 1 level.

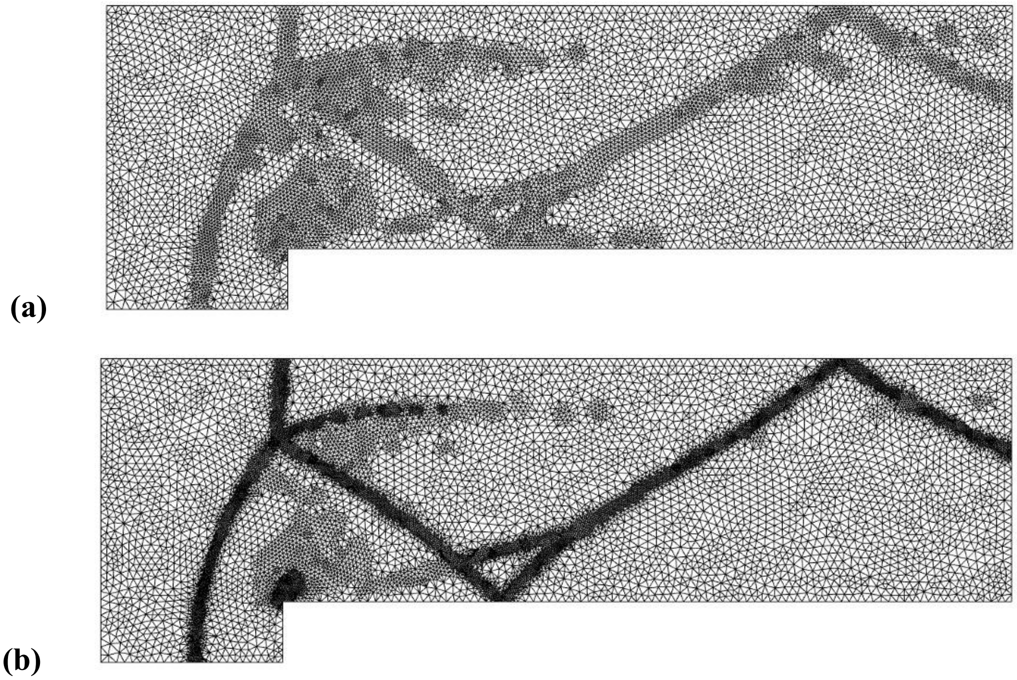


Figure 41. Grids for 3rd order SV scheme under adaptive h-refinement at time=4.0; (a) 1 level (14,988 triangles); (b) 2 levels (22,677 triangles); Refined from base grid every 100 time steps.

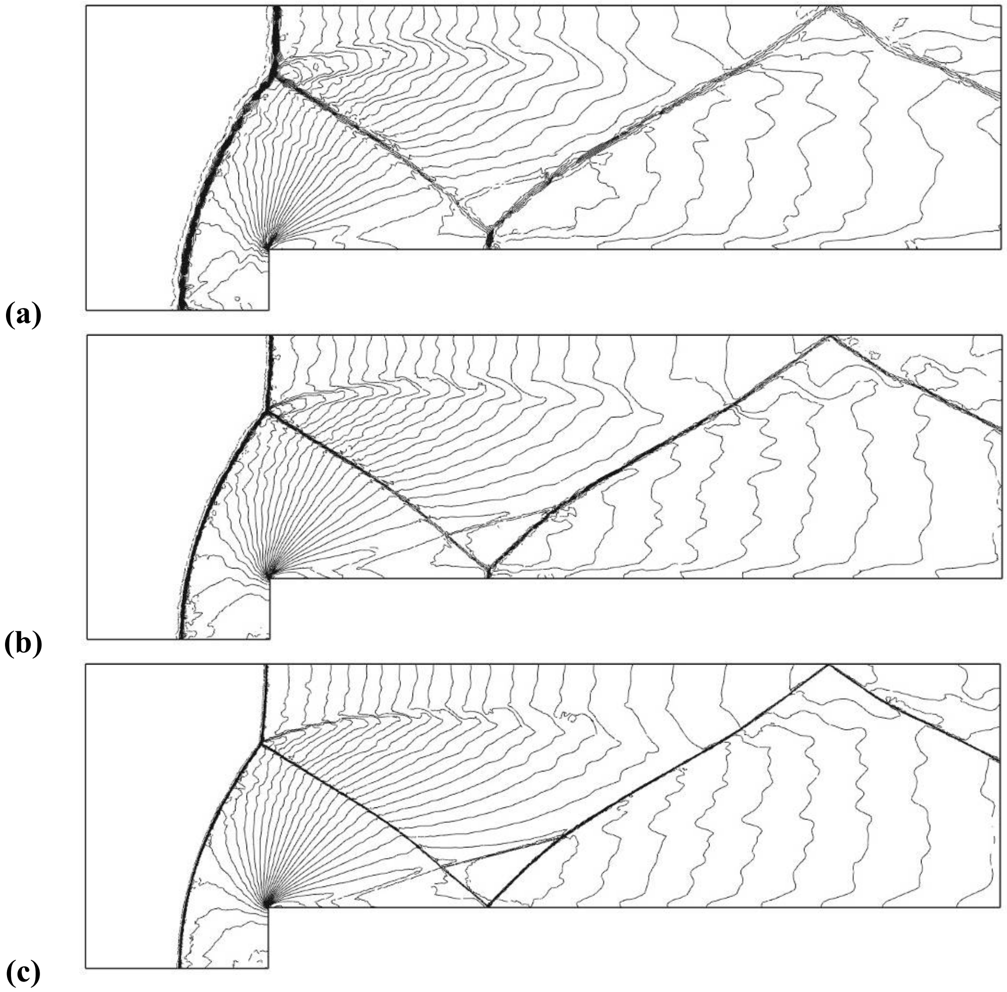


Figure 42. Density contours for 3rd order SV scheme under adaptive h-refinement at time=4.0; (a) Base grid (52,476 DOFs); (b) 1 level (89,928 DOFs);(c) 2 levels (136,062 DOFs); Refined from base grid every 100 time steps.

4.3.5 Rayleigh-Taylor instability problem

The Rayleigh-Taylor instability (RTI) problem involves a cold fluid overlying a warm fluid. Two inviscid fluids are initially taken to be in hydrostatic equilibrium in an isolated chamber, as shown in Figure 43a.

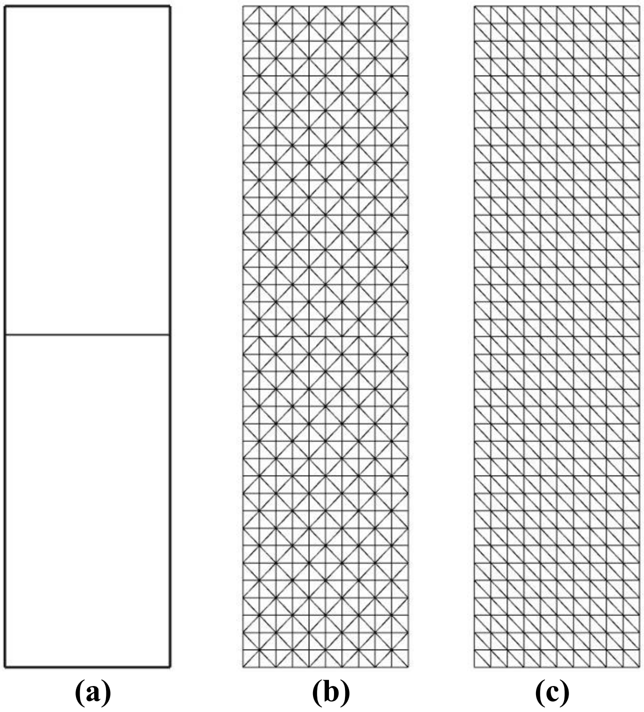


Figure 43. Problem domain and grids for RTI problem; (a) domain; (b) symmetric grid (10x38x2 triangles); (c) asymmetric grid (10x38x2 triangles).

The chamber is 1 unit high, and 0.25 units wide. The upper half of the chamber contains a fluid of density two, while the lower half of the chamber contains a fluid of unit density. The initial pressure field is chosen to ensure hydrostatic equilibrium, and an initial perturbation of the velocity field triggers the instability.

The flow is governed by the Euler equations with the addition of source terms in the y-momentum and energy equations which correspond to unit gravity in the downward direction. The initial data is summarized in Table 11, where M_0=0.1, τ=6, γ=1.4, $\varepsilon_y = M_0\sqrt{\gamma/2}$, and $\varepsilon_x = -\varepsilon_y \tau/16$.

Table 11. Initial conditions for RTI problem. The geometric center of the chamber is taken to be the origin of the coordinate system.

Parameter	upper part	lower part
ρ	2	1
u	$\varepsilon_x\sin(8\pi x)\cos[\pi(y+1/2)]\sin^{\tau-1}[\pi(y+1/2)]$	same as upper part
v	$-\varepsilon_y\cos(8\pi x)\sin^{\tau}[\pi(y+1/2)]$	same as upper part
p	$2-2y$	$2-y$

A perturbation is selected which gives rise to a single mode instability, and inviscid wall boundary conditions are used for the chamber walls. While there are no shock waves in this problem, there is a contact discontinuity between the two fluids.

Figure 44 shows 1st, 2nd, 3rd, and 4th order results for this case on both symmetric and asymmetric grids with no adaptation. In all plots, 30 even contours of density between 0.84 and 2.4 are presented. In addition, Roe flux is used for all RTI cases and the TVD limiter is taken to be based on the conserved variables. These practices dramatically increase the quality of the results for the RTI problem. From Figure 44, it is evident that as the polynomial order is increased on a uniform grid, the solution contours are over-dissipated by the limiter. Thus, if a local min/max-based limiter is used for this problem, adaptive local h-refinement is essential if high-order accuracy is sought. For comparison, Figure 45 shows 2nd and 3rd order results using uniformly refined grids. Results and grids for this case obtained using local adaptive h-refinement are shown in Figure 46 through Figure 48. In all cases considered here, the grid is refined from the base grid every 10 time steps.

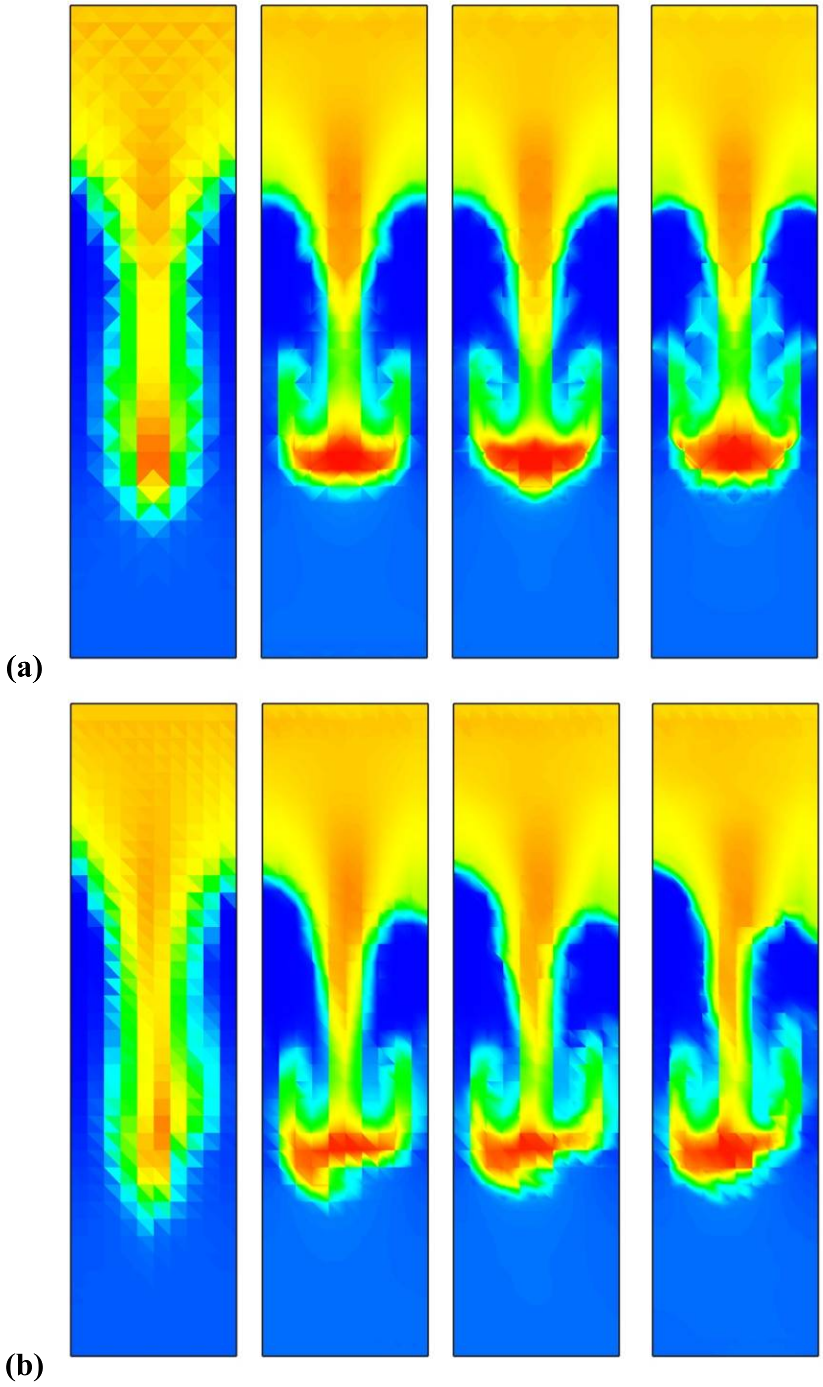


Figure 44. 1st, 2nd, 3rd, and 4th order density contours (left-to-right, 760, 2,280, 4,560, and 7,600 DOFs) for RTI problem at time=1.8 with no adaptation; (a) symmetric grid; (b) asymmetric grid.

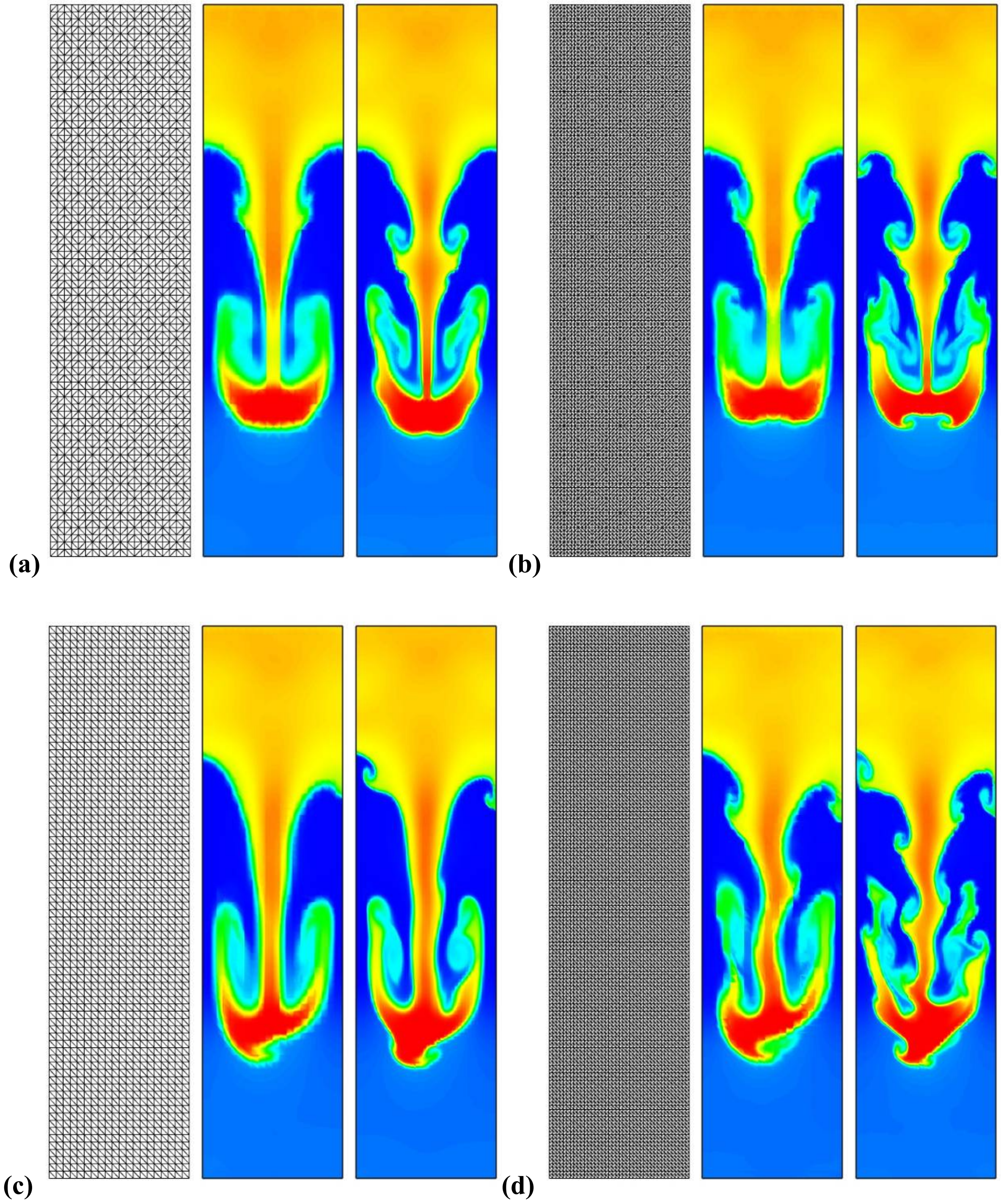


Figure 45. Grid, 2nd and 3rd order density contours (left-to-right) for RTI problem at time=1.9 with no adaptation; (a) symmetric grid (20x76x2 triangles); (b) symmetric grid (40x152x2 triangles); (c) asymmetric grid (20x76x2 triangles); (d) asymmetric grid (40x152x2 triangles).

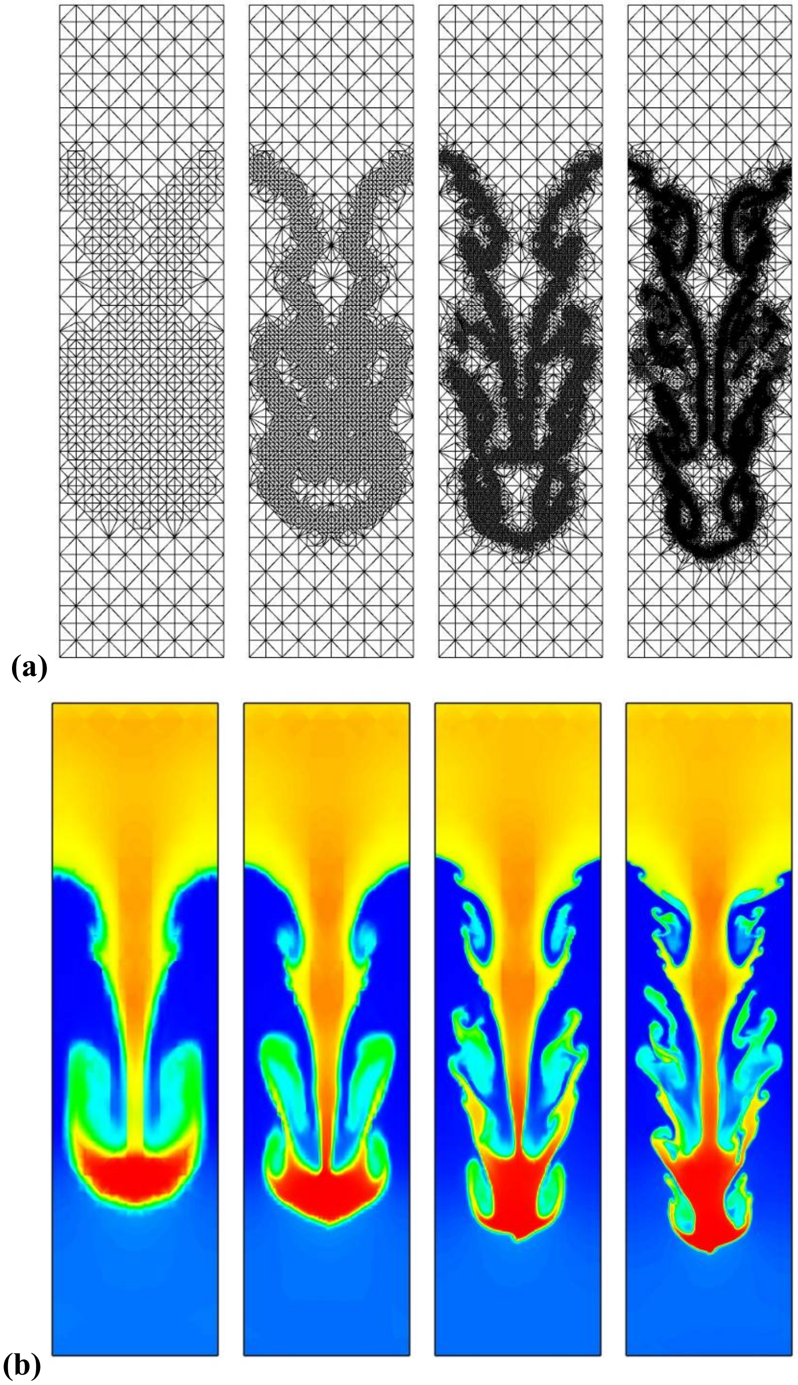


Figure 46. Results for RTI problem at time=1.9 on symmetric grids; (a) grids obtained using 1-4 levels of adaptation (left-to-right, 1,810, 4,960, 14,079, and 38,281 triangles); (b) 2nd order density contours obtained using 1-4 levels of adaptation (left-to-right, 5,430, 14,880, 42,237, and 114,843 DOFs).

Both symmetric and asymmetric base grids are employed for this simulation. It is immediately apparent that the behavior of the RTI problem is heavily dependent on the grid used. Namely, if the initial grid is symmetric, the solution tends to stay symmetric (for the most part), otherwise the solution is completely asymmetric. In all cases, the typical mushroom-cap behavior is observed, with increasingly complicated flow structure downstream as the number of adaptation levels is increased. Comparing Figure 46 through Figure 48, it is clear that local adaptive h-refinement is far more effective than global refinement at resolving the flow features for this problem. In fact, a much more highly resolved solution is obtained using local h-refinement with far fewer degrees-of-freedom than is necessary for a global refinement strategy to produce similar results.

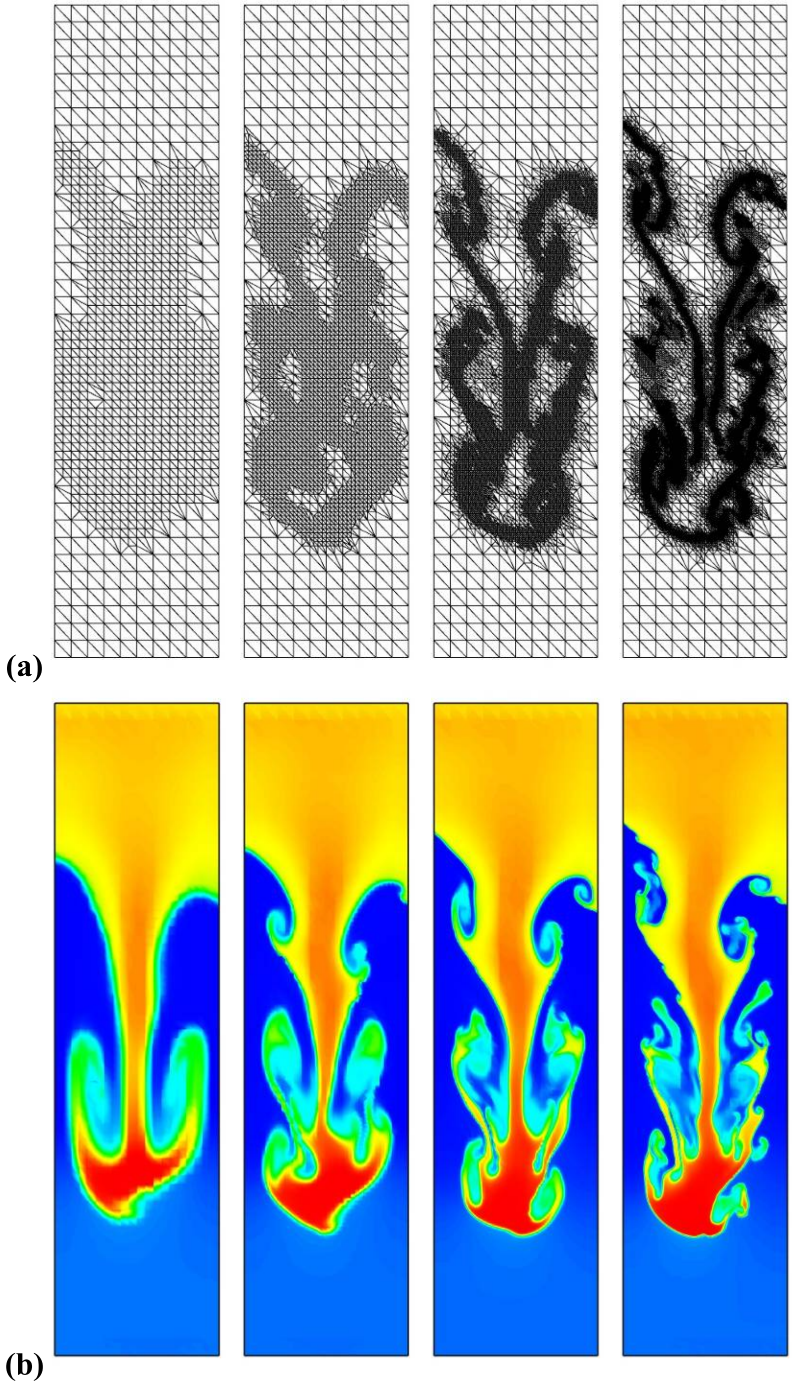


Figure 47. Results for RTI problem at time=1.9 on asymmetric grids; (a) grids obtained using 1-4 levels of adaptation (left-to-right, 1,834, 5,226, 14,391, and 36,185 triangles); (b) 2nd order density contours obtained using 1-4 levels of adaptation (left-to-right, 5,502, 15,678, 43,173, and 108,555 DOFs).

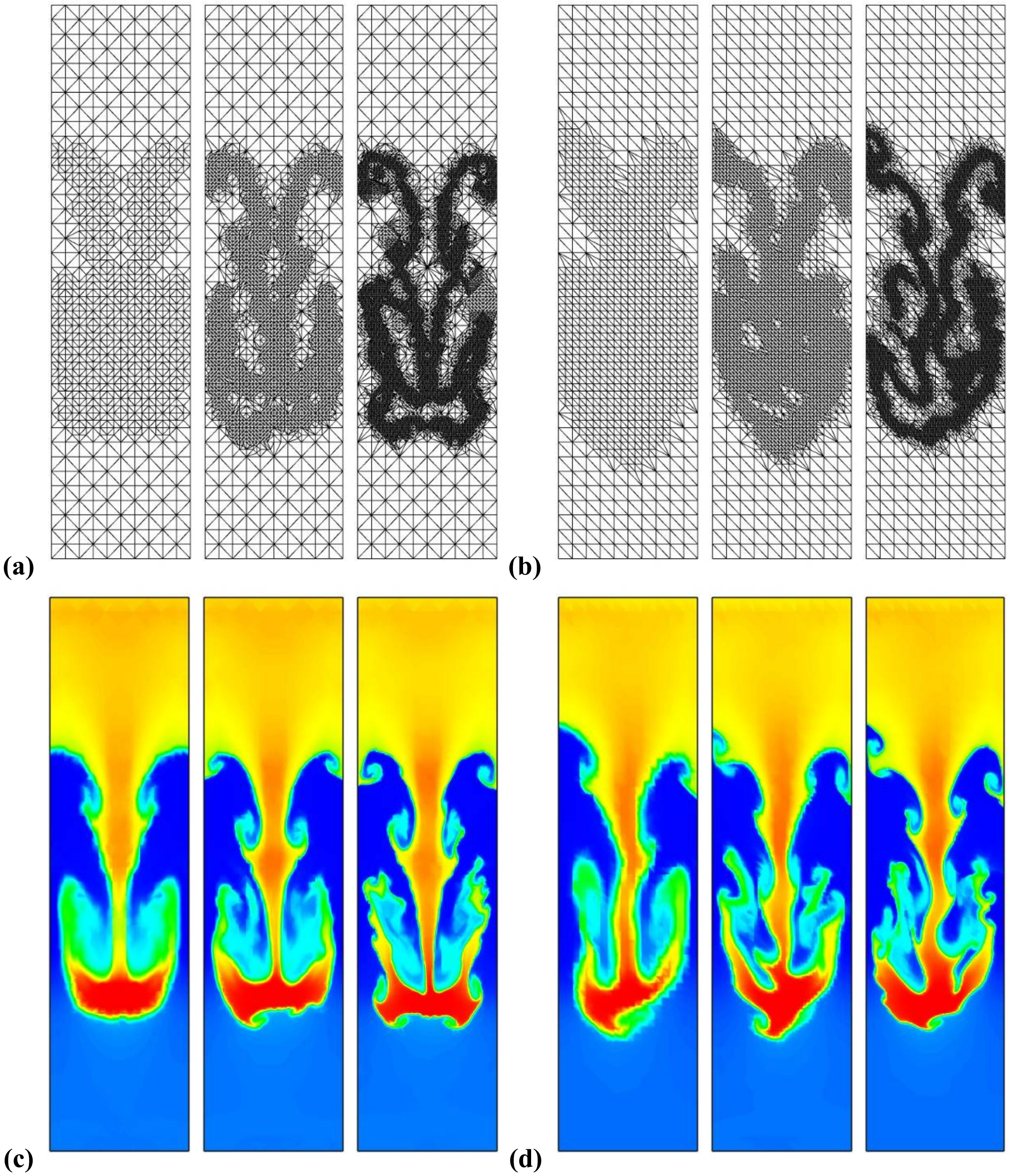


Figure 48. 3rd order results for RTI problem at time=1.9 obtained using 1-3 levels of adaptation; (a) symmetric grids (left-to-right, 1,770, 4,788, and 13,513 triangles); (b) asymmetric grids (left-to-right, 1,891, 5,254, and 13,698 triangles); (c) density contours for symmetric grids (left-to-right, 10,620, 28,728, and 81,078 DOFs); (d) density contours for asymmetric grids (left-to-right, 11,346, 31,524, and 82,188 DOFs).

4.3.6 Reflection of Mach 3 shock wave from 2 offset circular cylinders

This problem involves a right-moving Mach 3 shock wave impacting two offset circular cylinders. The domain is 1 unit high and 1 unit wide, with two 0.15 radius cylinders located at (0.4, 0.25) and (0.5, 0.75), respectively. The shock wave is initially located at *x*=0.2, and the solution is carried out until time=0.16.

The base grid for this case, and results for 2nd, 3rd and 4th order simulations with no adaptation are shown in Figure 49.

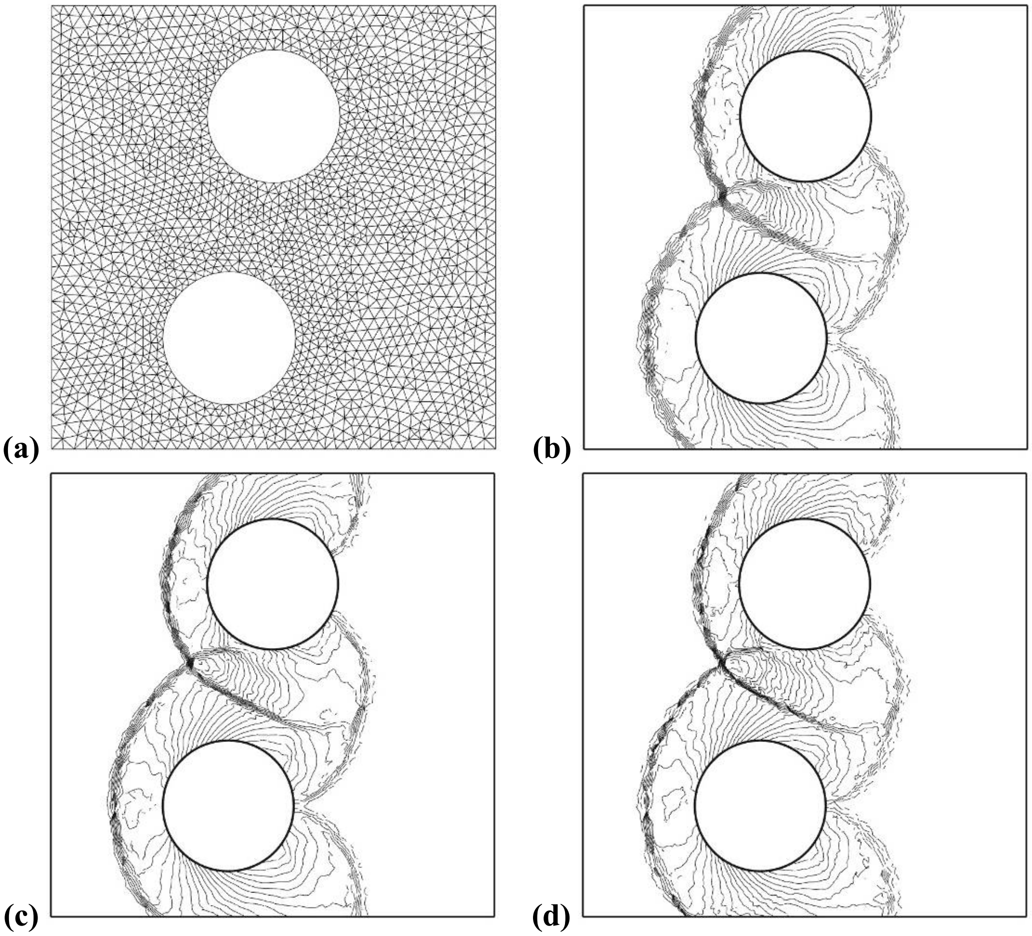
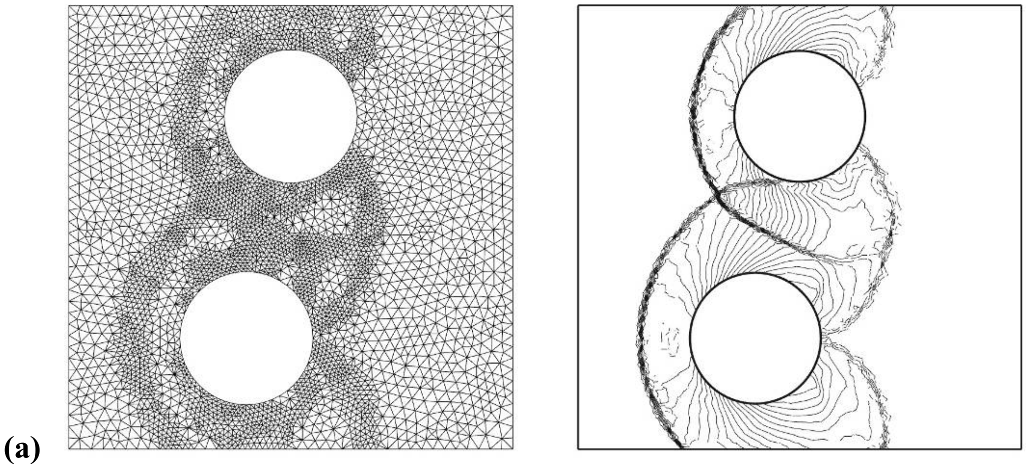


Figure 49. Results for 2 cylinder Mach reflection case at time=0.16 with no adaptation; (a) base grid (3,740 triangles); (b) 2nd order density contours (11,220 DOFs); (c) 3rd order density contours (22,440 DOFs); (d) 4th order density contours (37,400 DOFs).

It is evident that the 3rd order simulation has more effective resolution of the shock waves than the 2nd order simulation, however, the 4th order simulation has resolution comparable to the 3rd order simulation. This is likely because limiters based on a local maximum principle are often over dissipative, and can effectively reduce a high-order simulation to low-order.

Figure 50 and Figure 51 show 2nd and 3rd order results and grids for this simulation subjected to various levels of h-adaptation. In all plots, 30 even contours of density between 0.3 and 18.0 are presented. It is apparent that as the adaptation level is increased, the resolution of the shock waves increases markedly. This is true for the region where the shock reflects off of the cylinder, as well as for the region where the shocks intersect. From Figure 51, it is clear that the 3rd order simulation gives better resolution of the shocks, and more accuracy in smooth regions than the 2nd order case.



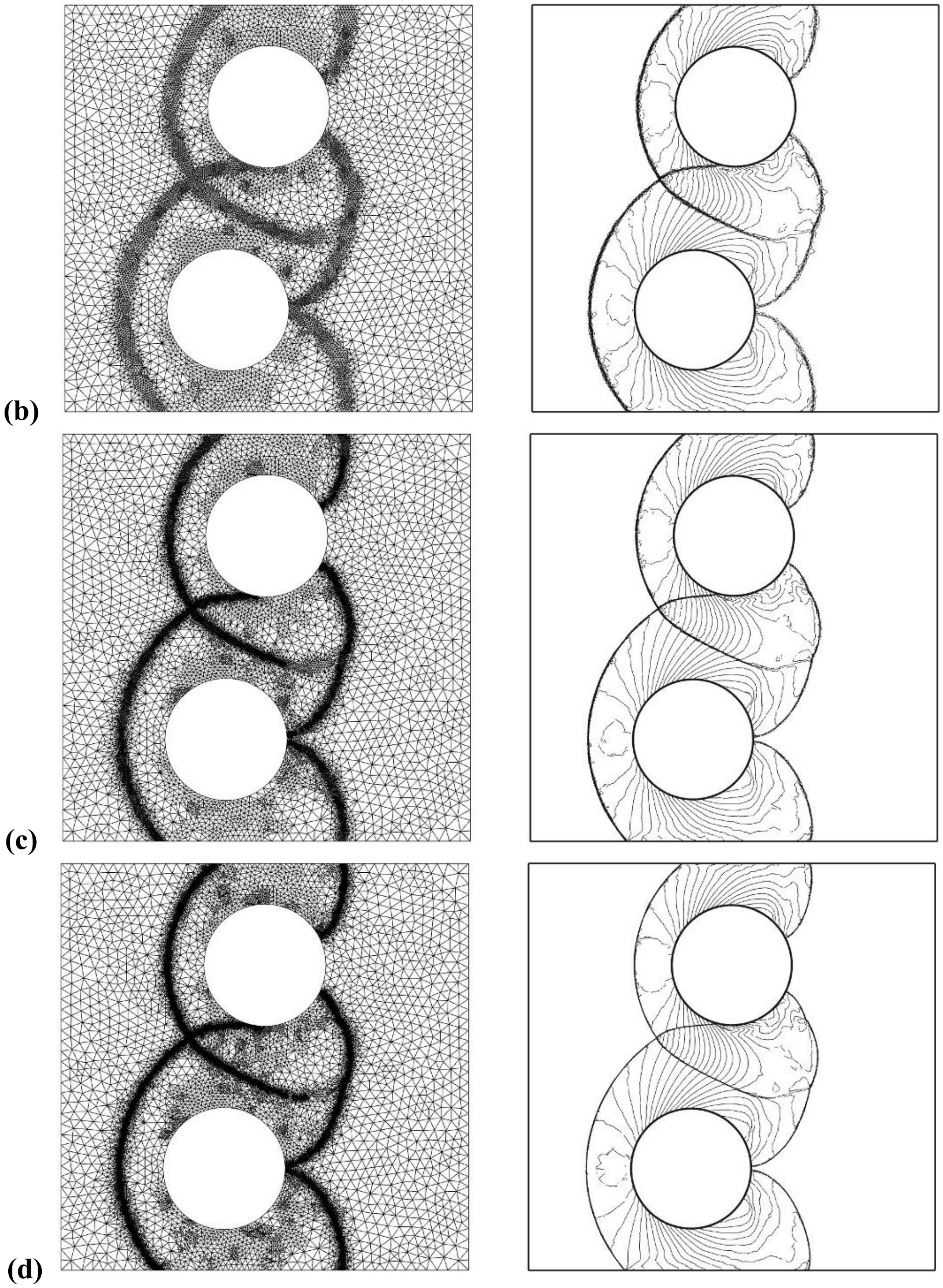


Figure 50. 2nd order grid and density contours for 2 cylinder Mach reflection case at time=0.16; (a) 1 level (22,566 DOFs, 7,522 triangles); (b) 2 levels (39,144 DOFs, 13,048 triangles); (c) 3 levels (71,427 DOFs, 23,809 triangles); (d) 4 levels (134,484 DOFs, 44,828 triangles); Refined from base grid every 100 time steps.

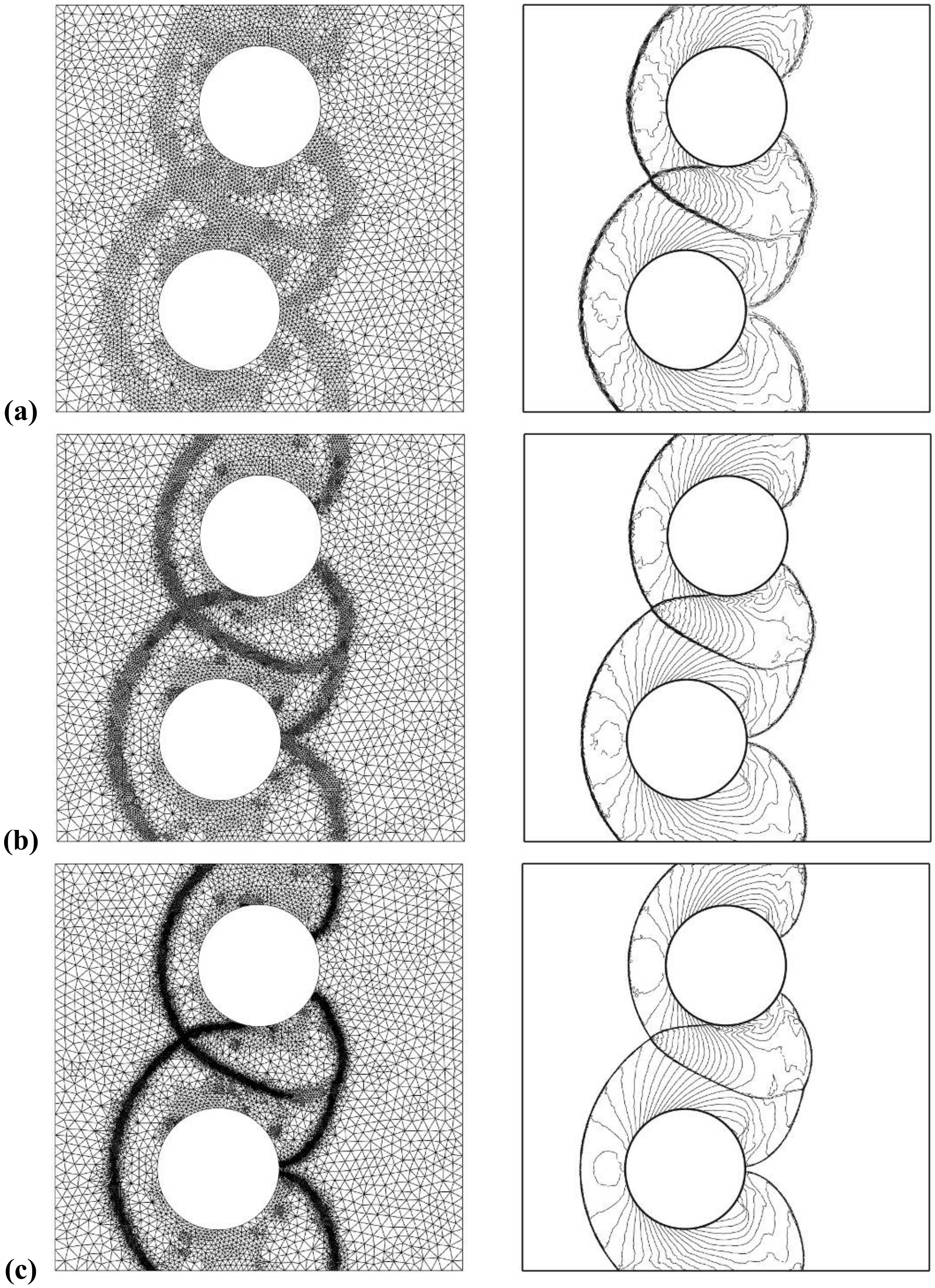


Figure 51. 3rd order grid and density contours for 2 cylinder Mach reflection case at time=0.16; (a) 1 level (45,756 DOFs, 7,626 triangles); (b) 2 levels (78,978 DOFs, 13,163 triangles); (c) 3 levels (143,580 DOFs, 23,930 triangles); Refined from base grid every 100 time steps.

4.4 Conclusions

The high-order quadrature-free spectral volume method has been successfully extended for use with local adaptive hp-refinement. A hierarchical edge-based adaptation algorithm was employed for high efficiency. The p-refinement methodology was effectively utilized for the case of subsonic flow over a NACA 0012 airfoil and a NASA GA(W)-1 airfoil, and the h-refinement technique was also employed with success for transonic flow over a NACA 0012 airfoil. In addition, the h-refinement technique was also demonstrated for supersonic flow in a wind tunnel with a forward-facing step, reflection of a moving shock wave off of 2 offset circular cylinders, and the Rayleigh-Taylor instability problem. It was demonstrated that local adaptive h-refinement is far more effective than global refinement at resolving important flow features, and a much more highly resolved solution can often be obtained using local adaptive h-refinement with far fewer degrees-of-freedom than is necessary for a global refinement strategy to produce similar results.

CHAPTER 5. STABILITY ANALYSIS FOR SV METHOD

An analysis of the accuracy and stability properties of the 2D spectral volume method, with emphasis on extremely high-order accurate simulations, is presented. In general, the partitioning of an SV into CVs is not uniquely defined, and thus it is of great importance to select a partition which yields favorable stability properties, and results in an interpolation polynomial of high quality. In recent work, researchers have largely focused on polynomial quality, abandoning stability considerations, when designing SV partitions. There has been some work advocating the use of stability analyses in the design of SV partitions, however a guiding principle which seeks to simultaneously provide favorable stability properties and the highest quality polynomial possible, is very much missing. Here we seek to address the issue of stability, as well as polynomial quality, in the design of SV partitions. A new approach is presented, which efficiently locates stable partitions by means of constrained minimization. This is motivated by the fact that, at present, an exhaustive search approach to SV partition design would be prohibitively costly and thus not feasible. Once stable partitions are located, a high quality interpolation polynomial is then assured by subsequently minimizing the dissipation and dispersion errors of the stable partitions. Preliminary results are presented for new 4th, and 5th order partitions.

5.1 Review of recent work

The partitioning of an SV into CVs has been one of the greatest challenges in the implementation of the SV method since its inception [50]. This partitioning defines the reconstruction stencil, and thus plays a vital role in determining the accuracy and stability properties of the scheme. Early on, several researchers focused on using the Lebesgue constant as a means to design accurate SV partitions [51,32,8,9]. While this criteria may be used to find partitions with lower error bounds, it does not guarantee that a particular scheme will be more or less accurate, and it offers no information about the stability of the scheme. A positive step towards addressing the issue of stability was given by Van den Abeele et al. [43,44,45]. In this work, some previously used SV partitions were found to be weakly unstable, and several new stable partitions were proposed. It was also shown that the new partitions had lower dissipation and dispersion errors than some previously used partitions despite having larger Lebesgue constants. This showed that although the Lebesgue constant should be small to ensure a lower upper bound on the error, it need not be minimal for a scheme to possess superior accuracy.

5.2 Framework for stability analysis

The wave propagation properties associated with a particular SV partition and Riemann flux carry information about both the accuracy and stability of the numerical scheme. Recent work by Van den Abeele et al. [43-45] has utilized the so-called Fourier footprint to facilitate the design of stable SV partitions with favorable wave propagation properties. A similar analysis is employed here.

Consider the 2D linear advection equation

$$\frac{\partial Q}{\partial t}+\nabla\cdot\left(\vec{a}Q\right)=0\,, \tag{5.1}$$

with periodic boundary conditions, where $\vec{a}=a(\cos\psi,\ \sin\psi)$ is the wave propagation velocity, and ψ is the wave propagation direction. The initial solution is taken to be a harmonic plane wave

$$Q\left(x,y,0\right)=e^{Ik\left(x\cos\theta+y\sin\theta\right)}, \tag{5.2}$$

with wave number k, I is the square root of -1, and θ is the orientation angle. The SV method is then applied to (5.1) on a grid comprised entirely of equilateral triangles. The simplest unit which produces the entire grid when periodically repeated, commonly referred to as the generating pattern (GP), is shown in Figure 52.

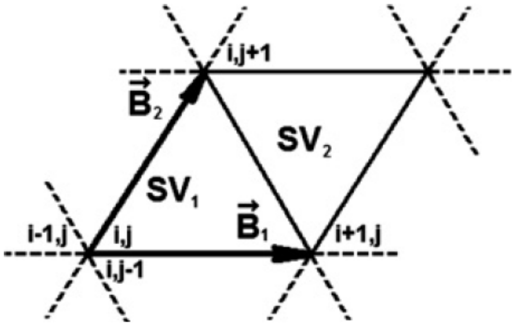


Figure 52. Generating pattern for the grid (taken from [44]).

The GP is completely defined by the vectors $\vec{B}_1$ and $\vec{B}_2$. On the boundary between two SVs, the following Riemann flux is employed

$$\hat{F}\left(Q_L,Q_R\right)=\frac{1}{2}\left(\vec{a}\cdot\vec{n}+\varphi\left|\vec{a}\cdot\vec{n}\right|\right)Q_L+\frac{1}{2}\left(\vec{a}\cdot\vec{n}-\varphi\left|\vec{a}\cdot\vec{n}\right|\right)Q_R\,, \tag{5.3}$$

where Q_L is the solution due to the SV to the left of the face, Q_R is the solution due to the SV to the right of the face, and φ is an upwinding parameter, where $\varphi = 0$ gives rise to a central flux, and $\varphi = 1$ results in a simple upwind flux. After applying the SV method to (5.1), we obtain

$$\sum_{n=1}^{2N}\left[\frac{V'\Delta B}{a}U_{m,n}\frac{d\overline{Q}_{i,j;n}}{dt}+M_{m,n}^{0}\overline{Q}_{i,j;n}+M_{m,n}^{-1}\overline{Q}_{i-1,j;n}+M_{m,n}^{+1}\overline{Q}_{i+1,j;n}\right.$$

$$\left.+N_{m,n}^{-1}\overline{Q}_{i,j-1;n}+N_{m,n}^{+1}\overline{Q}_{i,j+1;n}\right]=0, \tag{5.4}$$

where *ΔB* is the magnitude of $\vec{B}_1$, a is the magnitude of $\vec{a}$, *V'* is the volume of an SV nondimensionalized by ΔB^2, the index m varies from 1 to $2N$, and the indices i and j denote a particular GP. The variables $\overline{Q}_{i,j;n}$ for n = 1 to N are the CV-averages corresponding to the first SV in the GP (SV$_1$), while the variables for n = N + 1 to $2N$ are the CV-averages corresponding to the second SV in the GP (SV$_2$), and the matrices $U_{m,n}, M_{m,n}^{0}, M_{m,n}^{-1}, M_{m,n}^{+1}, N_{m,n}^{-1}$, and $N_{m,n}^{+1}$ are functions of the wave propagation direction and are given in the Appendix. Substitution of the harmonic plane wave $\overline{Q}_{i,j;m}(t)=\widetilde{Q}_m e^{I\left[k\left((iB_{1x}+jB_{2x})\cos\theta+(iB_{1y}+jB_{2y})\sin\theta\right)-\omega t\right]}$ into (5.4) yields

$$\sum_{n=1}^{2N}\left[-I\widetilde{\Omega}V'U_{m,n}+M_{m,n}^{0}+M_{m,n}^{-1}e^{-IK\left(B_{1x}^{'}\cos\theta+B_{1y}^{'}\sin\theta\right)}+M_{m,n}^{+1}e^{+IK\left(B_{1x}^{'}\cos\theta+B_{1y}^{'}\sin\theta\right)}\right.$$

$$\left.+N_{m,n}^{-1}e^{-IK\left(B_{2x}^{'}\cos\theta+B_{2y}^{'}\sin\theta\right)}+N_{m,n}^{+1}e^{+IK\left(B_{2x}^{'}\cos\theta+B_{2y}^{'}\sin\theta\right)}\right]\widetilde{Q}_n=0, \tag{5.5}$$

where K is the nondimensional wave number, and $\widetilde{\Omega}$ is the nondimensional numerical frequency. The numerical dispersion relation can then be written as

$$\Big|-I\widetilde{\Omega}V'\mathbf{U}+\mathbf{M}^{o}+\mathbf{M}^{-1}e^{-IK\left(B'_{1x}\cos\theta+B'_{1y}\sin\theta\right)}+\mathbf{M}^{+1}e^{+IK\left(B'_{1x}\cos\theta+B'_{1y}\sin\theta\right)}$$

$$+\mathbf{N}^{-1}e^{-IK\left(B'_{2x}\cos\theta+B'_{2y}\sin\theta\right)}+\mathbf{N}^{+1}e^{+IK\left(B'_{2x}\cos\theta+B'_{2y}\sin\theta\right)}\Big|=0\,, \qquad (16)$$

from which $\widetilde{\Omega}$ can be readily computed. The quantity $-I\widetilde{\Omega}$ is the so-called Fourier footprint $\Re=\Re^{\text{Re}}+I\Re^{\text{Im}}$ of the discretization, $\Re^{\text{Im}}$ being a measure of the dispersive properties of the scheme, and $\Re^{\text{Re}}$ being a measure of the dissipative properties. To ensure stability, $\Re^{\text{Re}}$ should be nonpositive for all *K, θ*, and *ψ*.

5.3 Partition generation and optimization

Families of SV partitions which are considered here, as with those considered in previous work [8,44], contain quadrilateral, pentagonal, and hexagonal CVs. It turns out that for an interpolation polynomial of degree *m*, there must be 3 quadrilateral corner CVs, 3(*m*-1) pentagonal side CVs, and (m^2-3*m*+2)/2 hexagonal interior CVs. Initially a uniform partition of this family is generated, as shown in Figure 53 for polynomials of degree 3, 4, and 5.

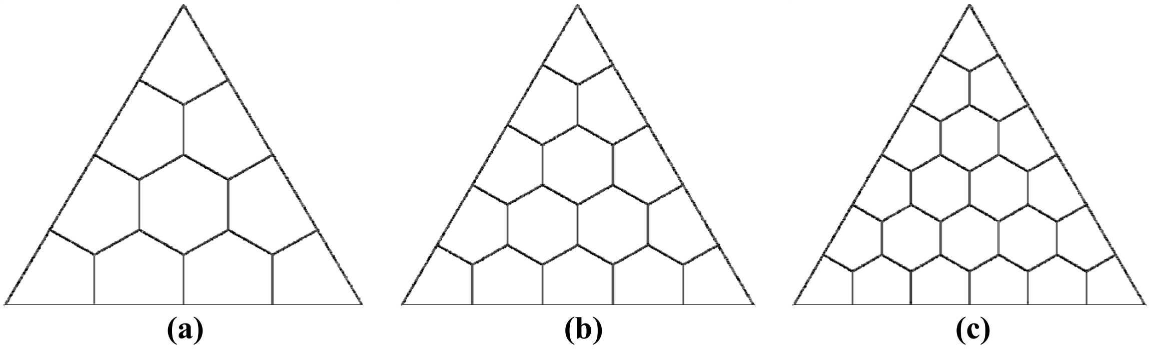

(a) **(b)** **(c)**

Figure 53. Uniform partitions for polynomials of degree 3 (a), 4 (b), and 5 (c).

The parameters that uniquely define the SV partition, which will subsequently be referred to as the control vector, can then be determined. The control vector essentially contains the locations of the nodes which physically define the shapes of the CVs within the SV. If any linear manipulation of the SV, while keeping the SV center fixed, causes a given node to coincide with any other node, then any movement of that given node is tied to the movement of the "coincident" nodes. These "coincident" nodes will hereafter be referred to as partner nodes, as shown in Figure 54 for partitions of degree 3 and 4. Denote the components of the control vector as (α_4, β_4, γ_4, δ_4) for a 4th order partition. These components refer to positions of the square, circle, diamond, and delta symbols from Figure 54, respectively.

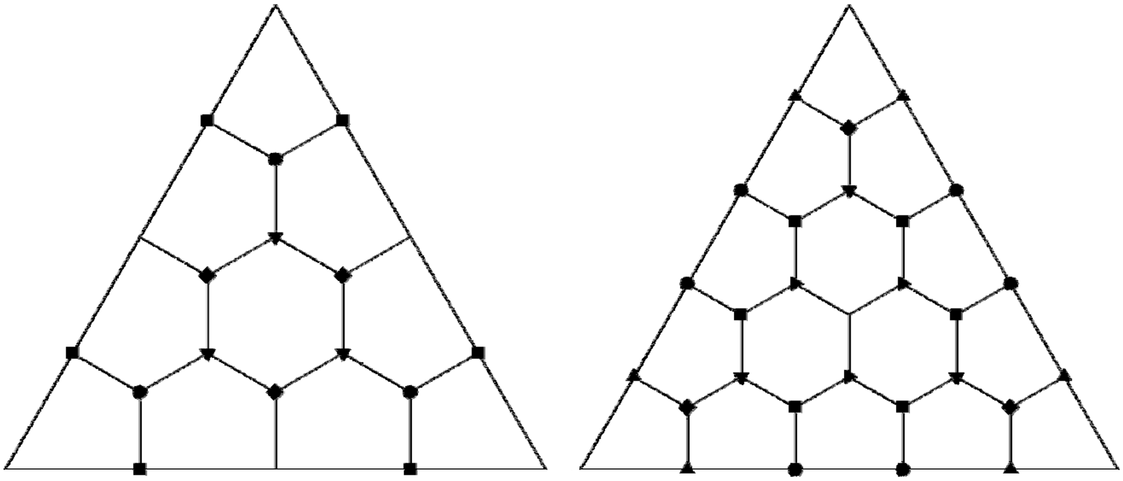


Figure 54. Identification of partner nodes (denoted by like shapes) for partitions of degree 3 (left) and 4 (right). The degree 3 partition is completely defined by 4 parameters, whereas the degree 4 partition requires 7 parameters.

Thus if a given node is found to have partner nodes, for simplicity, the control vector need only contain the position of the given node. Furthermore, the three corner nodes, as well as the SV center node (if it exists) and the SV edge center nodes (if they exist) are omitted from the control vector, since they are immovable. Finally, if a node exists on an SV edge or on an SV line of symmetry, that node is constrained to move along a line and is thus represented by a single coordinate in the control vector, while all other nodes are unconstrained and are represented by two coordinates in the control vector.

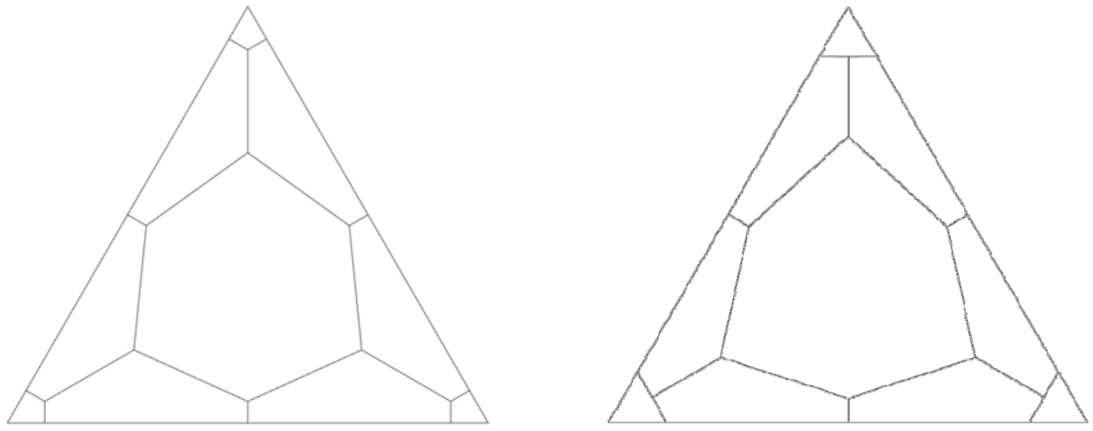
The constrained minimization program called CONMIN [48] is employed to optimize the SV partitions. CONMIN is a gradient-based optimizer which utilizes the method of Feasible Directions [61] to find the Feasible Direction, and then move in that direction to update the control vector. The objective or cost function for CONMIN is taken to be the maximum real part of the Fourier footprint of the scheme $\Re_{\max}^{\mathrm{Re}}$. Since CONMIN is used for minimization, it will attempt to drive $\Re_{\max}^{\mathrm{Re}}$ to as low a value as possible, and if it reaches a nonpositive value, a stable partition has been discovered. Then, upon discovery of many stable partitions, those with the lowest dissipation and dispersion errors are deemed likely to be suitable for simulation.

5.4 Results

The constrained minimization approach outlined above has been applied to a 4th-order SV partition. In Figure 55, we present a new partition denoted as "SV4H", and compare with a partition previously proposed by Van den Abeele et al. denoted as "SV4P" [44], which is known to have favorable stability properties and high accuracy. In Figure 56 we present the dispersion and dissipation errors for the both partitions as a function of wave number, leaving the wave angle fixed at π/6. It is evident that the new partition proposed here has better agreement with the exact solution for a wider range of wave numbers than partition "SV4P". In addition, plots of the corresponding Fourier footprints in Figure 57 show that the "SV4H" partition also has a smaller Fourier footprint than the "SV4P" partition, and thus will allow for larger time steps to be taken. To validate this analysis numerically, both partitions are used to solve the problem of linear advection of a sine wave on a 10x10x2 grid with periodic boundary conditions. The solution is carried out until time=400. Figure 58 shows the time history of the residual for this simulation, and it is clear that the "SV4H" partition has considerably less damping than the "SV4P" partition. It is thus apparent that the "SV4H" partition is capable of preserving a wave for a longer period of time than the "SV4P" partition on the same grid.

Table 12. Control vector and Lebesgue constant for "SV4P" and "SV4H" partitions.

Partition	α_4	β_4	γ_4	δ_4	$\|\|\Gamma_{\Pi}\|\|$
SV4P	0.07800000	0.07800000	0.03900000	0.26325000	4.2446
SV4H	0.12061033	0.09097092	0.05000000	0.23419571	4.0529



(a) **(b)**

Figure 55. Partitions for 4th-order SV schemes; (a) “SV4P” proposed in Van den Abeele et al. [44]; (b) “SV4H" proposed here.

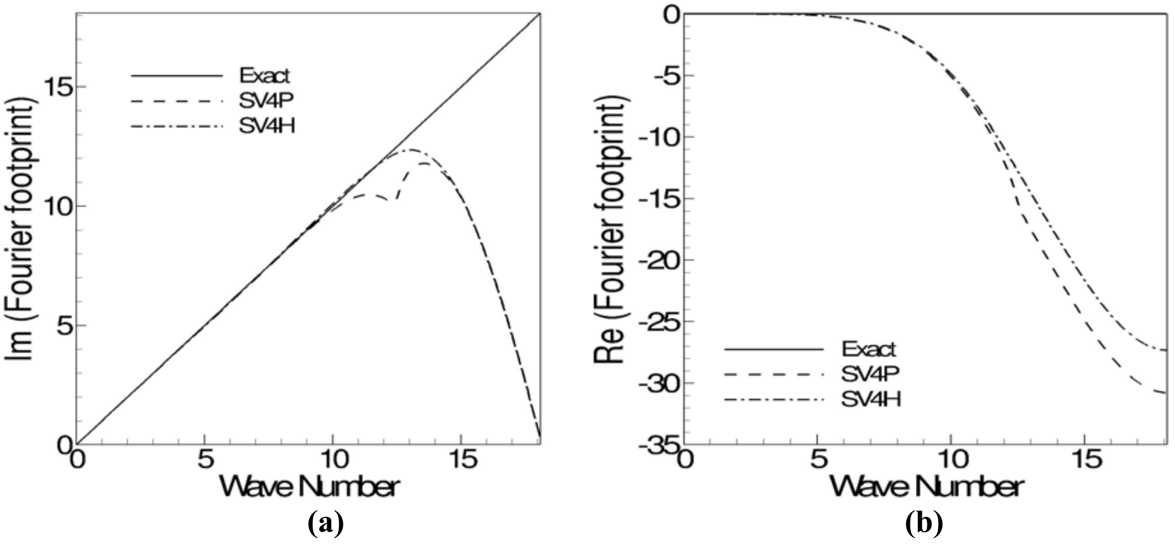


Figure 56. Dissipation and dispersion errors as a function of wave number for various 4th-order partitions. The wave angle considered here is π/6. (a) dispersion error vs. wave number; (b) dissipation error vs. wave number.

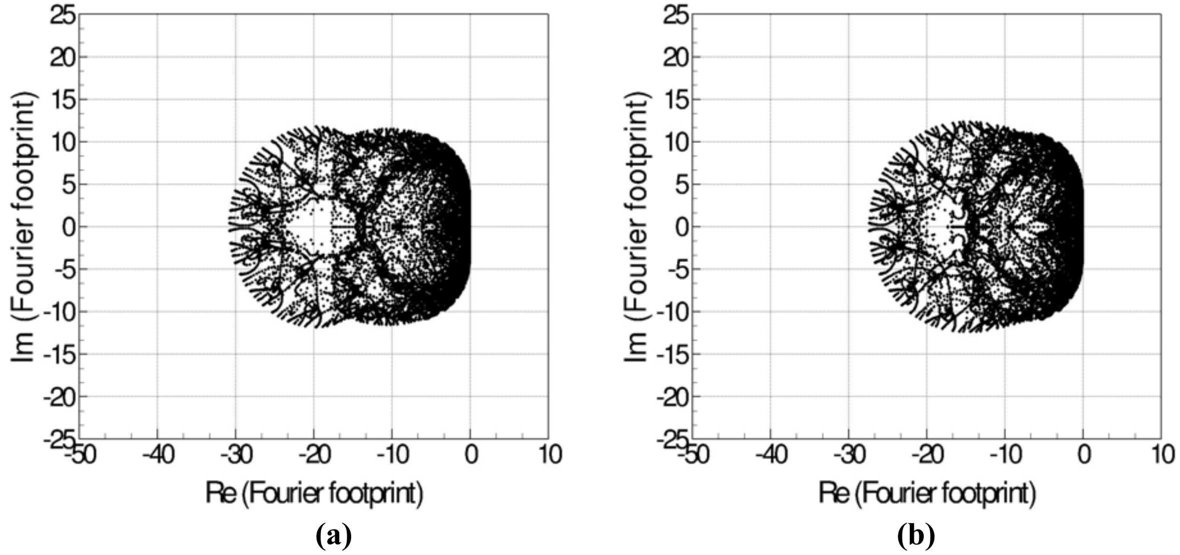


Figure 57. Fourier footprint for 4th-order partitions; (a) "SV4P"; (b) "SV4H".

The new approach has also been applied to a 5th order partition, and preliminary studies have produced some stable 5th order partitions. An example of a stable 5th order partition and its corresponding Fourier footprint are shown in Figure 59. Although more work still needs to be done to minimize the dissipation and dispersion errors of the partition, it is now very promising that this procedure can be used to design stable SV partitions of arbitrary order.

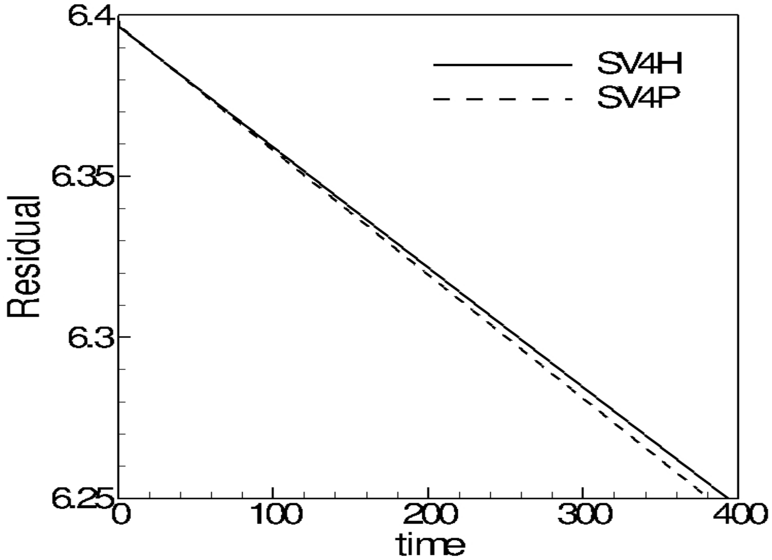


Figure 58. Residual vs. time history for 4th order simulation of the linear advection of a sine wave in a 10x10x2 domain with periodic boundary conditions.

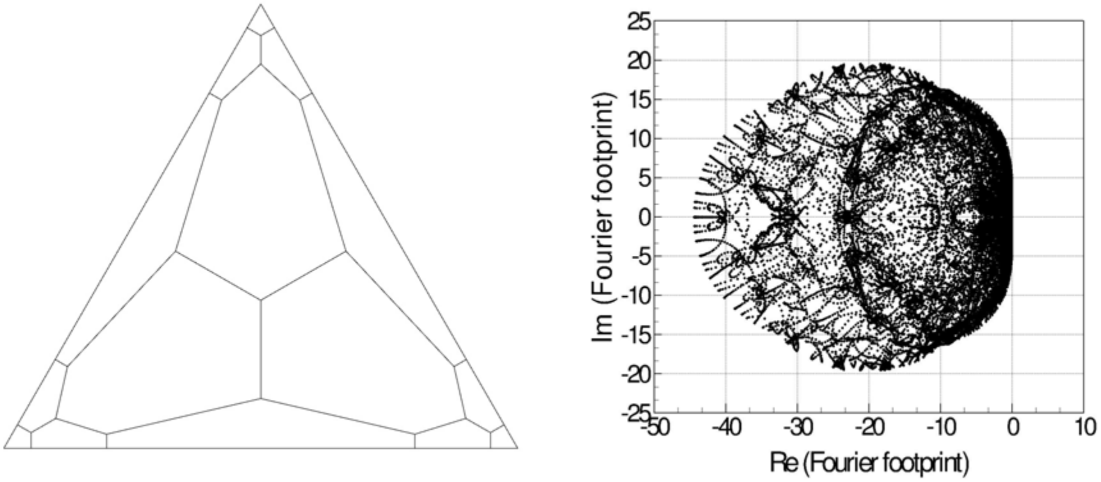


Figure 59. Stable 5th order partition (left) and corresponding Fourier footprint (right).

5.5 Conclusions

A new method for obtaining stable SV partitions with low dissipation and dispersion errors via constrained minimization has been presented. The methodology for automatically generating a uniform partition and extracting the parameters that define the design space is outlined. Preliminary results are given for both 4th and 5th order partitions which demonstrate the potential of this method for producing stable and highly accurate partitions of arbitrary order.

CHAPTER 6. SUMMARY AND FUTURE WORK

An efficient new quadrature-free implementation of the spectral volume method has been successfully carried out for scalar and Euler equations in both 2D and 3D. Two different approaches to compute the flux integrals more efficiently, the partial quadrature (PQ) and quadrature free (QF) approaches, have been developed and evaluated for both 3rd- and 4th-order SV schemes in 2D, and for 2nd- and 3rd-order schemes in 3D. The savings is significant in 2D, but in 3D the hundreds or thousands of flux calculations per SV required in the traditional approach are reduced to only dozens. For all approaches, it has been found that the nearly optimum order of accuracy can be obtained in both the L_1 and L_∞ norms with respect to density errors, and results for several well known inviscid flow test cases have further demonstrated the capabilities of the new method.

The new quadrature-free implementation has also been successfully extended for use with local adaptive hp-refinement. A hierarchical edge-based adaptation algorithm was employed for high efficiency. The p-refinement methodology was effectively utilized for the case of subsonic flows over airfoils, and the h-refinement technique was also employed with success for several transonic and supersonic flow cases, both in the presence of strong shock waves and curved boundaries. It was demonstrated that local adaptive h-refinement is far more effective than global refinement at resolving important flow features, and a much more highly resolved solution can often be obtained using local adaptive h-refinement with far fewer degrees-of-freedom than is necessary for a global refinement strategy to produce similar results.

A new method for obtaining stable SV partitions with low dissipation and dispersion error via constrained minimization has been presented. The methodology for automatically generating a uniform partition and extracting the parameters that define the design space is outlined. Preliminary results are given which indicate this to be an effective method for use in the design of stable and highly accurate partitions of arbitrary order.

Further investigation is needed for the design of very high-order stable SV partitions, as the design space becomes extremely large with increasingly high-order partitions. Moreover, the problem becomes even more expensive in 3D where the number of cell faces, even for slightly high-order accurate schemes, is overwhelming and is heavily CPU- and memory-intensive. For these reasons, and to allow for the simulation of practical problems of interest in 3D, there is a great need for a parallel implementation of the new QF approach. In addition, the algorithm for local adaptive h-refinement could be extended to 3D to promote a more efficient use of computational resources. These are some areas which could benefit from further study.

APPENDIX. MATRIX DEFINITIONS FOR STABILITY ANALYSIS

Here the expressions for the matrices $\boldsymbol{Q}$, $\boldsymbol{M}^0$, $\boldsymbol{M}^{-1}$, $\boldsymbol{M}^{+1}$, $\boldsymbol{N}^{-1}$, $\boldsymbol{N}^{+1}$ from Section 5.2 as outlined in Van den Abeele [44] are given. The indices *m* and *n* are CV indices that take on the values m, $n = 1,\ldots, N_p$, and the indices *i* and *j* run over the two SVs in the generating pattern, taking on the values 1 and 2. V'_m is the dimensionless volume of the CV with index *m* (same for all SVs). $S'^{,\text{int}}_{i,m}$ is the part of the boundary surrounding CV *m* in SV *i,* that does not belong to the boundary of SV *i*. $S'^{,lef}_{i,m}$, $S'^{,rig}_{i,m}$, $S'^{,low}_{i,m}$, and $S'^{,upp}_{i,m}$ are the parts of the boundary surrounding CV *m* in SV *i,* that belong to the left, right, lower, and upper boundary of the GP, respectively. $S'^{,con}_{i,m}$ is the part of the boundary surrounding CV *m* within SV *i,* that belongs to the boundary between SV 1 and SV 2. The normals to these boundaries always point out of the CV. Also, let us define local coordinates $\vec{\xi} = (\xi, \eta)$ such that:

$$\begin{aligned} x &= \Delta B \xi \\ y &= \Delta B \eta \end{aligned}. \tag{A.1}$$

Then the matrices are given as:

$$Q_{N(i-1)+m,N(j-1)+n} = \delta_{mn}\delta_{ij}\frac{V'_m}{V'}, \tag{A.2}$$

$$\begin{aligned} M^{0}_{N(i-1)+m,N(j-1)+n} &= \delta_{ij} \int_{S'^{,\text{int}}_{i,m}} L_{j,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n}\right)ds + \frac{1}{2}\delta_{i1}\delta_{j1} \int_{S'^{,con}_{1,m}} L_{1,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} + \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds \\ &+ \frac{1}{2}\delta_{i1}\delta_{j2} \int_{S'^{,con}_{1,m}} L_{2,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} - \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds + \frac{1}{2}\delta_{i2}\delta_{j2} \int_{S'^{,con}_{2,m}} L_{2,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} + \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds \\ &+ \frac{1}{2}\delta_{i2}\delta_{j1} \int_{S'^{,con}_{2,m}} L_{1,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} - \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds + \frac{1}{2}\delta_{i1}\delta_{j1} \int_{S'^{,lef}_{1,m}} L_{1,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} + \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds \\ &+ \frac{1}{2}\delta_{i2}\delta_{j2} \int_{S'^{,rig}_{2,m}} L_{2,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} + \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds + \frac{1}{2}\delta_{i1}\delta_{j1} \int_{S'^{,low}_{1,m}} L_{1,n}\left(\vec{\xi}\right)\left(\vec{l}_a \cdot \vec{n} + \varphi\left|\vec{l}_a \cdot \vec{n}\right|\right)ds \end{aligned}$$

$$+\frac{1}{2}\delta_{i2}\delta_{j2}\int\limits_{S_{2,m}^{',upp}} L_{2,n}\left(\vec{\xi}\right)\left(\vec{l}_a\cdot\vec{n}+\varphi\left|\vec{l}_a\cdot\vec{n}\right|\right)ds\,, \tag{A.3}$$

$$M_{N(i-1)+m,N(j-1)+n}^{-1}=\frac{1}{2}\delta_{i1}\delta_{j2}\int\limits_{S_{1,m}^{lef}} L_{2,n}\left(\vec{\xi}+\vec{B}_1\right)\left(\vec{l}_a\cdot\vec{n}-\varphi\left|\vec{l}_a\cdot\vec{n}\right|\right)ds\,, \tag{A.4}$$

$$M_{N(i-1)+m,N(j-1)+n}^{+1}=\frac{1}{2}\delta_{i2}\delta_{j1}\int\limits_{S_{2,m}^{rig}} L_{1,n}\left(\vec{\xi}-\vec{B}_1\right)\left(\vec{l}_a\cdot\vec{n}-\varphi\left|\vec{l}_a\cdot\vec{n}\right|\right)ds\,, \tag{A.5}$$

$$N_{N(i-1)+m,N(j-1)+n}^{-1}=\frac{1}{2}\delta_{i1}\delta_{j2}\int\limits_{S_{1,m}^{low}} L_{2,n}\left(\vec{\xi}+\vec{B}_2\right)\left(\vec{l}_a\cdot\vec{n}-\varphi\left|\vec{l}_a\cdot\vec{n}\right|\right)ds\,, \tag{A.6}$$

$$N_{N(i-1)+m,N(j-1)+n}^{+1}=\frac{1}{2}\delta_{i2}\delta_{j1}\int\limits_{S_{2,m}^{upp}} L_{1,n}\left(\vec{\xi}-\vec{B}_2\right)\left(\vec{l}_a\cdot\vec{n}-\varphi\left|\vec{l}_a\cdot\vec{n}\right|\right)ds\,. \tag{A.7}$$

BIBLIOGRAPHY

1. R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45-58.
2. J. D. Anderson, Computational fluid dynamics: the basics with applications, (McGraw-Hill Inc., New York, 1995).
3. H. L. Atkins, Chi-Wang Shu, Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations, AIAA J. 96 (1996) 1683.
4. T. J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.
5. F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138, 251-285 (1997).
6. P. Batten, N. Clarke, C. Lambert, D. M. Causon, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. Vol. 18 No. 6 (1997) 1553-1570.
7. R. Biswas, K. D. Devine, J. E. Flaherty, Parallel, adaptive finite element methods for conservation laws, Applied Num. Math. 14 (1994) 255-283.
8. Q.-Y. Chen, Partitions of a simplex leading to accurate spectral (finite) volume reconstruction. SIAM J. Sci. Comput. Vol. 27, No. 4 (2006) 1458-1470.
9. Q.-Y. Chen, Partitions for spectral (finite) volume reconstruction in the tetrahedron, SIAM J. Sci. Comput. (2005).
10. B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411-435.

11. B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90-113.

12. B. Cockburn, S. Hou, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545-581.

13. B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199-224.

14. M. Delanaye, Yen Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.

15. J. Ekaterinaris, High-order accurate, low numerical diffusion methods for aerodynamics, Progress in Aerospace Sciences 41 (2005) 192-300.

16. J. Flaherty, L. Krivodonova, J.-F. Remacle and M. Shephard, Aspects of discontinuous Galerkin methods for hyperbolic conservation laws. Finite Elements Anal. Design 38 (2002) 889–908.

17. S. K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271.

18. S. Gottlieb, C-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Review, v.43 n.1, p. 89-112, 2001.

19. T. Haga, N. Ohnishi, K. Sawada, and A. Masunaga, Spectral volume computation of flowfield in aerospace application using Earth simulator, AIAA Paper No.06-2823, 2006.

20. R. Harris and Z.J. Wang, High-Order Adaptive Quadrature-Free Spectral Volume Method on Unstructured Grids (under preparation).

21. R. Harris and Z.J. Wang, High-Order Adaptive Quadrature-Free Spectral Volume Method on Unstructured Grids, AIAA Paper No. 2008-779.

22. R. Harris, Z. J. Wang, Y. Liu, Efficient implementation of high-order spectral volume method for multidimensional conservation laws on unstructured grids, AIAA Paper No. 2007-912.

23. R. Harris, Z. J. Wang, Y. Liu, Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation, J. Comput. Phys. 227 (2008) 1620-1642.

24. A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231.

25. A. Harten, P. D. Lax, B. Van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35-61.

26. J. S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, SIAM J. Numer. Anal. Vol. 35 No. 2 (1998) 655-676.

27. J. S. Hesthaven, C. H. Teng, Stable spectral methods on tetrahedral elements, SIAM J. Sci. Comput. Vol. 21 No. 6 (2000) 2352-2380.

28. C. Hirsch, Numerical computation of internal and external flows, Volume 1: Fundamentals of numerical discretization, (Wiley, New York, 1988).

29. C. Hirsch, Numerical computation of internal and external flows, Volume 2: Computational methods for inviscid and viscous flows, (Wiley, New York, 1990).

30. C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97-127.

31. L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, J. Comput. Phys. 211 (2006) 492-512.

32. Y. Liu, M. Vinokur, Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems, J. Comput. Phys. 212 (2006) 454-472.

33. H. Luo, J. Baum, R. Löhner, On the computation of steady-state compressible flows using a discontinuous Galerkin method, presented at the Fourth International Conference on Computational Fluid Dynamics, July 10-14, 2006, Ghent, Belgium.

34. J-F Remacle, J.E. Flaherty, M.S. Shephard, An Adaptive Discontinuous Galerkin Technique with an Orthogonal Basis Applied to Compressible Flow Problems, SIAM Review, Vol. 45 No. 1 (2003) 55-73.

35. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357-372.

36. V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267-279.

37. W. F. Spotz and G. F. Carey, High-Order Compact Finite Difference Methods with Applications to Viscous Flows, Report 94-03 of the Texas Institute for Computational and Applied Mathematics, (1994) 1-63.

38. Y. Sun, Z. J. Wang, Y. Liu, High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, Commun. Comput. Phys., 2 (2007) 310-333.

39. Y. Sun, Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow, J. Comput. Phys. 215 (2006) 41-58.

40. J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, Computational fluid mechanics and heat transfer, (Taylor and Francis, Washington DC, 1997), 2nd Ed.

41. E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL Riemann solver, Shock Waves 4 (1994) 25-34.

42. S. Tu, S. Aliabadi, A slope limiting procedure in discontinuous Galerkin finite element method for gasdynamics applications, Int. J. Numer. Anal. & Mod. Vol. 2 No. 2 (2005) 163-178.

43. K. Van den Abeele, T. Broeckhoven, and C. Lacor, Dispersion and Dissipation properties of the 1D spectral volume method and application to a p-multigrid algorithm, J. Comput. Phys. 224 (2) (2007) 616-636.

44. K. Van den Abeele, and C. Lacor, An accuracy and stability study of the 2D spectral volume method, J. Comput. Phys. 226 (1) (2007) 1007-1026.

45. K. Van den Abeele, G. Ghorbaniasl, M. Parsani, and C. Lacor, A stability analysis for the spectral volume method on tetrahedral grids, (submitted to Elsevier Science, 3-17-2008).

46. B. van Leer, Towards the ultimate conservative difference scheme V. a second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101-136.

47. B. van Leer, Upwind and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes, Commun. Comput. Phys., 1 (2006) 192-206.

48. N. G. Vanderplaats, CONMIN USER'S MANUAL, Ames Research Center and U.S. Army Air Mobility, R&D Laboratory, Moffet Field, Calif. 94035, 1978.

49. Z. J. Wang, High-Order Methods for the Euler and Navier-Stokes Equations on Unstructured Grids, Journal of Progress in Aerospace Sciences, Vol. 43 No. 1-3 (2007).

50. Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210.

51. Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665.

52. Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: extension to one-dimensional systems, J. Sci. Comput. 20 (2004) 137.

53. Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2004) 716.

54. Z. J. Wang, Y. Liu, Extension of the spectral volume method to high-order boundary representation, J. Comput. Phys. 211 (2006) 154-178.

55. S. Wolfram, *Mathematica* Book (Wolfram Media and Cambridge Univ. Press, New York, 1999), 4th Ed.

56. P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115-173.

57. Y. Xing, C. W. Shu, A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms, Commun. Comput. Phys., 1 (2006), 100-134.

58. H. Yang, R. Harris, Z. Wang, and Y. Liu, Efficient quadrature-free 3D high-order spectral volume method on unstructured grids, AIAA Paper No. 2007-4325.

59. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 32 (1979) 335-362.

60. Zhang M, Shu CW. An analysis and a comparison between the discontinuous Galerkin method and the spectral finite volume methods. Comput Fluids 2005;34(4–5):581–92.

61. G. Zoutendijk, Methods of Feasible Directions, Elsevier Publishing Co., Amsterdam, 1960.