2009

Progress on unstructured-grid based high-order CFD method

Hong Yang
Iowa State University

Recommended Citation

Yang, Hong, "Progress on unstructured-grid based high-order CFD method" (2009). *Graduate Theses and Dissertations*. 10982.
https://lib.dr.iastate.edu/etd/10982

Progress on unstructured-grid based

high-order CFD methods

by

Michael Hong Yang

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:
Zhi Jian Wang, Major Professor
Tom Shih
Paul Durbin
Richard H. Pletcher
Richard G. Hindman

Iowa State University

Ames, Iowa

2009

DEDICATION

I would like to dedicate this thesis to my dear family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my adviser, Dr.Zhi-Jian Wang for the enormous research opportunities and financial support offered to me throughout my thesis. His inspirations, patience, knowledge, and enthusiasm have been essential for the completion of this thesis work and beyond as a lifetime educational experience to me. I owe my deep gratitude to Dr.Tom I-P Shih, Dr.Paul Durbin, Dr.Richard Pletcher, and Dr.Richard Hindman for serving on my doctoral committee and for their insightful advices on my thesis research. I feel lucky and humbled to have these respectful big names in CFD area on my thesis faculty member list. I would like to thank all my friends and fellow colleagues in the CFD Lab for many of their kind and friendly helps in my daily research needs. I am grateful to Dr.Joseph Schaefer for his willingness of training me on teaching skills and his thoughtful arrangements in great details on all the lab TA work or scheduling. From him I have learned many valuable things regarding teaching. Many thanks go to Ms.Delora Pfeiffer, who always provides to me with prompt information and handy reminders and many other aids. Also thanks go to the Air Force and the Department of Energy. This study has been supported by the Air Force Office of Scientific Research grant AF9550-06-1-0146, and the Department of Energy grant DE-FG02-05ER25677.

I owe thanks to my close friends around and in distance for lots of enjoyable time that are much needed to make even the stressful time bearable. Special thanks go to Dr.Rao Lindsay and Mrs.Marilyn Lindsay for their long-term friendship and love. Last, definitely not least, I would like to thank my family for their love and supports to give me strength to get this far.

ABSTRACT

Several new methods have been developed to meet the critical and diversified challenges in the state-of-art unstructured-grids based high-order methods for 3D real-world applications, including 1) parameter-free high-order generalized moment limiter for arbitrary mesh; 2) efficient line implicit method; 3) efficient quadrature-free SV method; 4) novel high-order mesh generation method for 3D hexahedral mesh. The parameter-free high-order generalized moment limiter does not need any user-specified free parameter to detect the discontinuities and exclude the smooth extrema. The present limiter has been designed to be naturally generic, compact, and efficient, which can be applied for arbitrary mesh and general unstructured-grids based high-order methods. The present low-storage line implicit BLU-SGS method significantly overcomes the anisotropy stiffness due to highly stretched wall grids in high Reynolds number flows. Up to 3 times of saving on CPU time and improved robustness have been demonstrated compared with the cell BLU-SGS solver. This line implicit method preserves the favorable feature of high compactness from the cell BLU-SGS method, and can be programmed as a black box so as to be easily applied in general high-order methods. The quadrature-free SV method has improved the original SV method by replacing the large number of quadrature for face integrals in 3D case with many less nodal operations based on analytical shape functions. Finally for high-order unstructured mesh generation, the present novel and fully automatic algorithm guarantee to resolve the self-intersection problem for non-linear quadrilateral or hexahedral mesh with strong robustness. The algorithm also offers the advantage of correcting grid self-intersection without changing the basic aspect ratio of the original grids or degrading the original grid quality.

CHAPTER 1 INTRODUCTION

Over the past three decades, we have witnessed that the frontier of Computational Fluid Dynamics (CFD) as a scientific and engineering discipline has been tremendously expanded along with its explosive market growth in industries such as aerospace, mechanical, chemical, and pharmaceutical engineering, etc. This growth was facilitated by advances of many aspects, of which the major ones are numerical algorithms, grid generation and adaptation, turbulence modeling, flow visualization, as well as the dramatic increase in computer CPU and network speed. Still, reveal of the full potential for CFD to solve 3D real-world problems largely depends on progress in these areas. Numerical algorithm is served as a role of hardcore in CFD. There was a complete link between the development of numerical methods for PDEs and CFD simulations, which makes CFD methods attractive beyond fluid dynamics as well, for example, in computational electromagnetics. The unstructured-grids based high-order methodology is the focus of this dissertation.

1.1 Background

Numerical methods with better accuracy have been extensively explored as a central task since the birth of CFD, particularly for the past three decades. The error order of a numerical method is measured by local truncation errors when the solution is smooth. The spatial error norm e for a method of $(k+1\ )$ order decreases with mesh size h according to

$$e \propto h^{k+1}. \tag{1.1}$$

In this thesis we refer to high-order methods by those with an order of accuracy of at least three $(k \geq 2)$. Without considering computer cost, even a second-order or a first-order method could produce highly accurate results if very fine mesh is used. However, the advantage of a high-order method is that it can achieve the same high accuracy more efficiently, as shown by Wang [127]. A similar analysis as to [127] yet with subtle difference in (1.3) is given as follows. Mesh size h can be estimated from the total number of solution unknowns or degree of freedoms ($NDOFs$) in domain of interest,

$$h \propto \left(\frac{m}{NDOFs}\right)^{1/d}. \tag{1.2}$$

Here $m = 1$ for FD method or FV method with one DOF for each grid point or cell; $m = \frac{\prod_{i=1}^{d}(k+i)}{d!}$ for those methods with multiple DOFs in each cell depending on polynomial degree k or accuracy order $k + 1$, for example, Discontinuous Galerkin method (DG) or Spectral Volume (SV) method or Spectral Difference (SD) method. d is the physical dimension number ($d = 1, 2, 3$). Therefore by combining (1.1) and (1.2) the error can be expressed as

$$e \propto \left(\frac{m}{NDOFs}\right)^{(k+1)/d}. \tag{1.3}$$

Notice that $1 \leq m \ll NDOFs$. Assuming that computer cost is roughly proportional to $NDOFs$, higher order method gives better accuracy than lower order method with the same $NDOFs$ or computer cost.

Choosing low-order or high-order method depends on the balance among computational speed, simplicity of coding, and resolution required. Adequate second-order numerical methods are often good choices for many engineering application problems, of which the solutions are piecewise simple (almost linear) with several isolated discontinuities in between, for example, the solution of most Riemann problems. That explains why most flow solvers in commercial CFD software packages are based on second-order numerical methods, either Finite Volume, Finite Difference, or Finite Element method. But high order methods are necessary for those complex problems that require high resolution for both discontinuities and rich small structures (mainly featured with unsteady vortex motion), for example, the Rayleigh-Taylor instability simulation [145,146], shock interaction with vortices [143,144], direct simulation of turbulence [79,115]. Computational aeroacoustics (CAA) is another example area where low-order methods are too dissipative, thus high-order methods are indispensable.

1.2 Current state-of-the-art

In terms of spatial discretization, the major CFD frameworks were historically cataloged as Finite Difference Method (FD), Finite Volume Method (FV), Finite Element Method (FE), and Spectral Methods, but recently there has been hybrid trend to build new algorithms by combining features from different methods. In order to make CFD a useful tool for the real-world problems, the criteria for a high-order algorithm to meet should include 1) accurate; 2) conservative; 3) geometrically flexible; 4) computationally efficient; 5) easy to implement. Among these criteria preservation of accuracy and local conservation are essential requirements. Obviously unstructured grids provide the best geometrical flexibility compare to structured grids. Regarding computational efficiency and implementation, it is preferred that a method is naturally parallelizable with the property of intrinsically high compactness, which means that data exchange is only needed between immediate neighboring cells. Also related to efficiency, a good scheme is expected to have good flexibility with unstructured mesh so as to allow easy *hp*-adaptation. Abundance of CFD methods have been generated towards the above goals during the short history of CFD, but here only several mainstream methods are to be discussed to illustrate the current state-of-the-art research on unstructured-grids based high-order CFD methods. A good review on this research area is given by Wang [127]. While the spatial discretization is the focus of this section, we briefly introduce the temporal discretization problem caused by high-order spatial operator in Section 1.3.2, and more details about temporal dsicretization will be given in Chapter 4.

Although FE method can achieve high-order accuracy for unstructured grids by using high-order solution and test function polynomial spaces, it is well known that the continuous Galerkin FE method gives rise to central-difference type approximation of the differential operator, so it is unstable for wave equation. Therefore for convection-dominated problems in fluid dynamics, stabilizing technique is needed for the standard continuous Galerkin method, for example, adding artificial dissipation as in the Streamline Upwind Petrov-Galerkin (SUPG) method. On the other hand, FV methods adequately reflect the physics of wave propagation in convection by using Riemann solver, a critical part of Godunov-type scheme for compressible flows. FV method preserves mean values, thus it is local conservative. The second-order FV methods now widely used in CFD industry are suitable for unstructured grids. Unfortunately, FV method loses its compactness in high-order case, because there is only one degree of freedom (DOF), which is the cell-averaged state value, on each cell; therefore multiple cells are needed to construct high-order flux at interface. The *k*-exact FV method developed by Barth and Frederickson [6] was a significant contribution for high-order FV method, but it still requires extended stencil which involves more than immediate neighboring cells. Large stencil destroys compactness and degrades efficiency of parallel computing.

For the past several years the discontinuous Galerkin (DG) finite element method, which combines the favorable features of FE and FV methods, probably has the most impact in high-order method community. Just like FV method and continuous FE method, the DG method approximates the integral version of the conservation law. First the PDE

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0 \tag{1.4}$$

is multiplied by a test function, then it is integrated by parts over each cell. This step is similar to common continuous FE method, but for a DG scheme, both the solution function (u) and the test function are represented by a piecewise polynomial function of degree k, respectively; therefore they are discontinuous at the cell interfaces. Next the idea of finite volume method is borrowed here to find monotone numerical fluxes at the interfaces by using Riemann solver to reflect the physics of wave propagation in convection, a critical part of Godunov-type scheme for compressible flows. The DOFs in DG are the expansion coefficients of the solution polynomial. The DG method is highly compact thus parallelizable because only data from neighboring cells is needed to update the DOFs in the current cell. The DG method also has some other favorable properties, such as provable L^2 stability, flexible for *hp*-adaptation and hanging grids [24].

The DG method was first introduced in 1973 by Reed and Hill [94], in the framework of neutron transport, i.e. a time independent linear hyperbolic equation, and then it was used for unsteady advection laws by Van Leer [120] in 1978. The pioneer development of the DG method for non-linear hyperbolic conservation laws was made by Cockburn, Shu and their collaborators in a series of papers on the Runge-Kutta DG (RKDG) method [27,30,28,26]. Bassi and Rebay made the breakthrough of developing the DG method for the compressible Euler and Navier-Stokes equations [10,9,11]. Many approaches have been developed to deal with the diffusion term in N-S equations, including the local DG (LDG) approach by Cockburn and Shu [25], the compact approach (BR2) by Bassi et al. [13], the interior penalty

(IP) method by Douglas and Dupont [35], and recovery scheme by van Leer [123], etc. To avoid solving Riemann problem for face flux there are active research on high-order central scheme and central DG methd [84,71] , which followed the fashion of the famous Lax-Friedrich scheme, yet reduced dissipative errors to high-order accuracy for convection-related problems. A quadrature-free approach by using node-based shape functions to replace the quadratures for the volume and face integrals was proposed by Atkins and Shu [4] to improve efficiency of the traditional DG methods. A comprehensive review on DG history and literature was given in [29]. Recently the differential version of the DG method was developed based on flux reconstruction for Cartesian mesh by Huynh [53,54], and this type of formulation for high-order scheme is capable of unifying the DG method with several other unstructured-grids based high-order methods such as staggered multi-domain method, spectral difference method, and spectral volume method. The flux reconstruction idea was generalized to "Lifting Collocation Penalty" approach by Wang, et al. [130] to handle triangular, tetrahedral, and prismatic cells, or cells of mixing types. The implementation for the differential version of the DG method is simpler than the original integral version.

The Spectral Volume (SV) method and Spectral Difference (SD) are recently developed high-order methods for unstructured grids, and employ the same solution space as the DG method, i.e., cell-wise discontinuous polynomials. They differ from DG on how the DOFs are defined and updated. The SV method is similar to a FV method, while the SD method is close to a FD method. Although the term "Spectral" is used here, the SV and SD methods use local continuous polynomials to do "finite" spectral reconstruction on multi-domains, compared to the original spectral method [43,17] that uses "global continuous" and orthogonal polynomials for spectral reconstruction on single domain. Just as the Spectral method, when applicable, the SV and SD methods have demonstrated the property of so called "exponential convergence" being the fastest possible.

In SV method, each cell (spectral volume) in the domain is partitioned into sub-cells (control volumes). The solution averages within these control volumes are defined as DOFs and used to build a higher-order reconstruction within the spectral volume. Next the solution average on each control volume evolves in the same way as FV method. At the spectral volume level, only immediate neighboring cells are involved to find high-order flux reconstruction at interface, therefore SV method is compact for high-order implementation. The SV method was developed by Wang, Liu and their collaborators [75,128,129,132,136] for hyperbolic conservation laws. The SV method has been successfully extended to Navier-Stokes equations [110,45], and 3D Maxwell equations [75]. Chen [21,22] developed many high-order SV partitions for simplexes in 2D and 3D with relatively small Lebesgue constants. Comparisons between the SV and DG methods were given in [106,142]. More recently a weak instability in several SV partitions has been identified from Fourier analysis by Van den Abeele et al [117,118], and new partitions were suggested thereafter. The quadrature-free implementation for the SV method has been developed for 2D flows by Harris et al [46] and 3D flows by the present author, et al [138]. The quadrature-free version of SV method is much more efficient than the standard quadrature based SV method, especially in 3D. Chapter 2 of this dissertation gives more details on Quadrature-Free SV (QFSV) method.

In SD method, two sets of points, i.e., the solution points and flux points are defined in each cell. The solution points are the locations where the nodal values of the state variable are specified. Flux points are the locations where the nodal values of fluxes are computed.

The DOFs are the conservative variables at the solution points. Actually the SD solution is independent from the locations of the solution points; only the distribution of flux points matters, as shown in [118]. Next the high-order flux reconstructions obtained from the solution points by "collocation method" are used to evolve the DOFs at the solution points. The SD method in 1D and for 2D quadrilateral mesh is similar to the staggered-grid multi-domain spectral method by Kopriva et al [60,61]. But the difference is that the SD method has the capability to handle simplexes such as triangles or tetrahedra. The SD method shares the same properties of both high compactness and high-order accuracy as the DG and SV method, but is easier to implement than DG and SV because it does not involve surface or volume integrals, especially for high-order curved boundaries. In 1D the SD method is equivalent to the SV method [118]. The SD method was first developed by Liu et al [73,74], then was extended to the Euler and Navier-Stokes equations [133,135, 82,83,52]. Huynh [53] proposed a set of 1D SV and SD schemes based on Legendre-Gauss quadrature points, which are stable for arbitrary orders of accuracy. The present author and Wang [137] developed a parameter-free high-order limiter which has been verified on the SD method to capture sharp discontinuity while preserving high-order accuracy in the smooth extrema region. Chapter 3 of this dissertation gives more details on high-order limiter. An efficient implicit line solver, which is presented in Chapter 4 of this dissertation, has been developed for the SD method by the present author.

The Essentially Non-Oscillatory (ENO) method [49] and Weighted ENO (WENO) method [70, 58] are two important milestones in constructing non-oscillatory schemes, which is also of significance for high-order methods. The basic idea of ENO/WENO schemes is to use "multiple moving" stencils to find a smooth solution reconstruction. Then this reconstruction polynomial is used to compute the state variables at the Gauss quadrature points, which are used to compute the Riemann fluxes. In ENO scheme, the “smoothest” reconstruction is selected from a set of candidate reconstructions built from several different local stencils according to certain smoothness criteria; while in WENO, the smooth high-order reconstruction is obtained from weighted (non-linear) average of the local low-order polynomials. The ENO/WENO schemes were originally developed for structured grids and now have been extended to unstructured grids [1,36,86,104,40,51].

The Residual Distribution (RD) methods have aroused significant interests in high-order method community. The basic idea of RD is to find the reconstruction or distribution of the residual on each node forming a cell from the cell residual, compared to other high-order methods such as DG, SV, SD, and WENO where the basic reconstruction is to find flux polynomials. Here the trick is how to build a conservative distribution function. The sum of the contributions from all the neighboring cells sharing the node is the nodal residual, which should vanish in steady state case, for example. Next an iterative procedure is used to find the nodal solution based on the nodal residual. Obviously this kind of schemes is compact and can be high-order accurate. The upwind RD methods were initiated by Roe [97] and then further developed in collaboration with Deconinck and collaborators [33,87,88,5,85]. Some other significant contributions to RD have been made by many researchers [18,3,95,87]. A comprehensive review of the RD methods is given by Abgrall in [2].

1.3 Remaining challenges

Although many progresses have been made for unstructured-grids based high-order methods, currently there still exist some critical issues for all the high-order methods. The common key component in the high-order methods (as discussed in Section 1.2) is cell-wise high-order polynomial reconstruction, or in general, high-order solution space on each cell, from which we obtain high order of spatial accuracy in smooth region. However, the high-order solution space is not like a free lunch; it causes the following problems wanting good answers.

1.3.1 Solution Discontinuity

How to sharply capture discontinuity while preserving high-order accuracy elsewhere in the solution field, particularly at smooth extrema? This is an old and new problem, which becomes much more difficult in high-order case because spurious oscillations caused by the Gibbs phenomenon at discontinuity are much larger and more out of control when using higher-order interpolation. The artificial viscosity methods suppress spurious oscillations but also degrade order of accuracy globally. The approach more often used is applying a robust locally first–order methods of Godunov-type at discontinuity and high-order method elsewhere for smooth region. However, currently the methods for discontinuity detection cannot guarantee that all the detected "discontinuities" are real ones; some could be smooth extrema. Therefore the first-order scheme originally intended to apply only at "discontinuity" could be misused and pollute smooth solution field. More background of this problem and research overview is given in Chapter 3 of this dissertation.

1.3.2 Efficient temporal discretization for steady problems

The high-order spatial operators are much stiffer than low-order ones. Therefore it is much more difficult to make a high-order simulation converged to steady state. The situation becomes even worse when this stiff spatial operator combines with the anisotropy induced stiffness, which is usually caused by highly clustered grids in the boundary layer for high Reynolds number viscous flow problems. In that case the explicit time integrator is too slow due to the CFL condition. The implicit methods allow much larger stable CFL number. However, currently the basic ideas of the implicit methods or multi-grids methods were extended from the low-order ones; none of them are specifically designed for high-order methods. Now that there are multiple DOFs in one high-order cell, the computer core memory occupied by the cell Jacobian matrices is much larger than that in low-order case. For example, the memory requirement for polynomial construction of degree higher than three might be prohibitive for a 3D engineering problem. Therefore the main challenge will be to develop effective and low storage implicit methods for high-order operators. More background of this problem and research overview is given in Chapter 4 of this dissertation.

1.3.3 High-order mesh generation

The curved boundaries should be accordingly represented by high-order boundary mesh in high-order method; the error generated from the linear element representation for curved boundary must eventually affect not only the boundary region, but also transport elsewhere in the flow field. One can always use very fine linear cells on curved boundary to reduce this error, but cannot eliminate it. Moreover, coarser mesh is actually expected in high-order method; otherwise it loses its advantage compared with low-order method. Another problem with linear cell using straight gridline is that the curved boundary gridlines intersect with interior gridlines if highly clustered grids are used near a curved boundary (for example, wall) in high-Reynolds boundary layer number flows. The self-intersected mesh is not allowed. Therefore high-order mesh generation is necessary for high-order methods.

Currently the ability to generate suitable high-order meshes (at least quadratic for curved boundary) for 3D complex geometries is a significant limiting factor for applying high-order methods in industry, because almost all the available grid generation packages can only generate linear cells. More background of this problem and research overview is given in Chapter 5 of this dissertation.

The high-order methods can only be used in CFD industry as widely as low-order methods when the above issues are solved adequately.

1.4 Objectives and accomplishments

Motivated by solving the issues as discussed in Section 1.3 and building efficient high-order CFD methods for 3D real-world applications, the objectives sought by the present author are as follows:

1) High-order limiting technique for discontinuity;
2) Low-storage efficient solver for high Reynolds number flows;
3) Efficient quadrature-free SV methods for 3D applications.
4) High-order mesh generation for 3D hexahedral mesh;

The accomplishments of the present dissertation include the following diversified aspects:

1) Parameter-free high-order generalized moment limiter for arbitrary mesh. Firstly the discontinuity marker created in this method does not need any user-specified free parameter to detect the discontinuities and exclude the smooth extrema. Secondly the limiter has been designed to be naturally compact and efficient.

Finally it is generic, which can be applied to arbitrary mesh and all the high-order methods. The publication for this work is [137,138].

2) Efficient line implicit solver with several new features including i) a scheme of a line BLU-SGS solver for the lined-up cells within the anisotropic thin boundary layer coupled with a cell BLU-SGS solver for other regions of less anisotropy stiffness, which significantly improves both robustness and convergence rate for highly stretched wall grids. Up to 3 times of saving on CPU time has been demonstrated compared with the cell BLU-SGS solver; ii) low memory storage requirement due to the partial line solver/partial cell solver scheme and an efficient low-storage strategy for LU decomposition of the cell Jacobians; iii) robust and accurate viscous fluxes for anisotropic grids based on the second approach of Bassi and Rebay (BR2); iv) generic and compact formulation and coding as a black box so as to be easily applied in general high-order methods.

3) Efficient quadrature-free SV methods for 3D application. This approach has improved the original SV method by replacing the large number of quadrature for face integrals in 3D case with many less nodal operations based on analytical shape functions. The analytical shape functions on the nodal points, which are to be used for flux reconstruction in the flow solver, have been pre-computed by using symbolic software such as Mathematica. The major contributions from the present author focus on the core parts of the 3D quadrature-free SV method, which include 1) found the complicated connectivity in 3D partition (linear, quadratic, and cubic) of a SV cell, which includes sub-faces, nodes, sub-cells (CVs), flux directions, and orientations relative to the neighboring cells; 2)

successful computed those complicated shape functions for each node. Also an efficient method [62] to deal with curved boundary has been coupled into this quadrature-free SV method. The publication for this work is [139].

4) High-order mesh generation for 3D hexahedral mesh. This novel fully automatic algorithm guarantee to resolve the self-intersection problem for high-order quadrilateral or hexahedral mesh with strong robustness. The algorithm also offers the advantage of correcting grid self-intersection without changing the basic aspect ratio of the original grids or degrading the original grid quality.

1.5 Outlines of dissertation

The dissertation is organized as follows. In Chapter 2 we present the efficient quadrature-free SV method in 3D. This chapter also includes the formulations of the original SV method and 3D partitions of a spectral volume. In Chapter 3 we focus on the parameter-free high-order limiting technique including the parameter-free and accuracy-preserving marker and the high-order generalized moment limiter. Also in this chapter a review is given to the SD method, which plays a role as a test carrier of the limiter in this chapter and the implicit line solver in next chapter. After that, we present the efficient low-storage line solver for high Reynolds number flows in Chapter 4. In Chapter 5 the new high-order mesh generation method for curved boundary and highly clustered boundary grids is given. Finally we conclude this dissertation and discuss some possible future research work in Chapter 6.

CHAPTER 2 QUADRATURE-FREE SPECTRAL VOLUME METHOD

In this chapter we discuss the high-order spectral volume (SV) methods. Firstly we have extended 2D SV method to 3D SV with high-order sub-cell partitions. Secondly an efficient quadrature-free approach to implement the SV methods has been developed to achieve high efficiency while maintaining accuracy. Also an efficient method to deal with curved boundary has been coupled into this quadrature-free SV method.

The focus of this chapter is the quadrature-free approach for the SV methods. In the SV method, in order to perform a high-order polynomial reconstruction, each simplex cell – called a spectral volume (SV) – is partitioned into a "structured" set of sub-cells called control volumes (CVs) in a geometrically similar manner, thus a universal reconstruction formula can be obtained for all SVs from the cell-averaged solutions on the CVs. The SV method avoids the volume integral required in the DG method, but it does introduce more cell faces where face integrals are needed. Therefore in order to improve the efficiency of SV method, the quadrature-free approach was developed by replacing the large number of quadrature for face integrals (in 3D case) in the original SV methods with many less nodal operations based on pre-computed analytical shape functions.

2.1 The general 3D spectral volume method

Consider the 3D conservation law in the following form,

$$\frac{\partial Q}{\partial t} + \nabla \cdot F = 0, \tag{2.1}$$

on domain $\Omega \times [0, T]$ and $\Omega \subset \mathrm{R}^3$ with the initial conditions within Ω and appropriate boundary conditions on $\partial\Omega$. The conservative solution variable Q can be a scalar or a vector, and the generalized flux F can be a scalar, vector, or even tensor. Domain Ω is discretized into I nonoverlapping tetrahedral cells (or elements), which are called “spectral volumes (SVs)”. The SV cells are further partitioned into CVs in a geometrically similar manner, as shown in Figure 2.1. For a complete 3D polynomial basis, a reconstruction of degree of precision p requires at least N CVs, where

$$N(p) = (p+1)(p+2)(p+3)/6. \qquad (2.2)$$

From the point of view of best interpolation polynomial, the optimal partition should make the Lebesgue constant minimum [129]. Therefore we use the following partitions which have been optimized for a minimal Lebesque constant. Figure 1 shows the linear, quadratic, and cubic partitions of a tetrahedral SV given by Chen [21,22], where each CV is enclosed by planar polygonal faces for ease of computation. The Lebesque constants are 5.08 and 6.87 for quadratic and cubic partitions, respectively.

Integrating (2.1) over each CV, we obtain

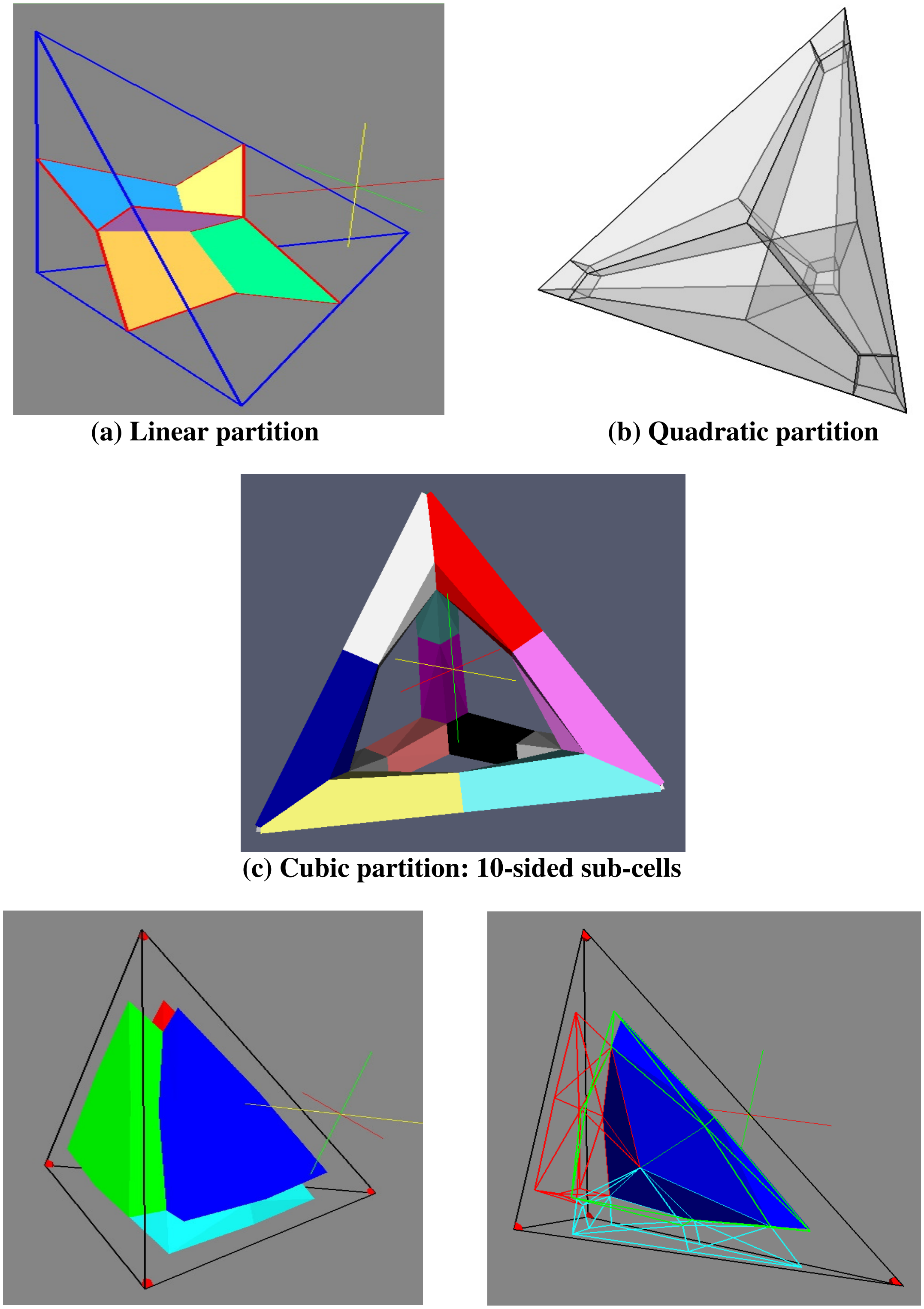
$$\frac{\partial \bar{Q}_{i,j}}{\partial t} + \frac{1}{V_{i,j}} \sum_{r=1}^{K} \int_{A_r} F \cdot \vec{n}_{A_r} dA = 0, \qquad (2.3)$$

where $C_{i,j}, i = 1, \dots, I; j = 1, \dots, N(p)$. $\bar{Q}_{i,j}$ is the cell-averaged solution on $C_{i,j}$; A_r represents the faces (with normal $\vec{n}_{A_r}$) that enclose $C_{i,j}$.

A high-order polynomial is then reconstructed within each SV such that

$$\frac{1}{V_{i,j}} \int_{C_{i,j}} P_i^p(x,y,z) dxdydz = \bar{Q}_{i,j}, \qquad (2.4)$$

where P_i^p is a polynomial (or vector polynomial) of degree p for the i-th SV.



(a) Linear partition

(b) Quadratic partition

(c) Cubic partition: 10-sided sub-cells

(e) Cubic partition: 6-sided sub-cells and 19-sided sub-cells

Figure 2.1 Partition of a tetrahedron

To actually solve the reconstruction problem, we introduce the complete polynomial basis, $e_l(x, y, z) \in D_p$, where $D_p = Span\{e_l(x, y, z)\}_{l=1}^{N(p)}$, and D_p denote the space of degree p polynomials in three dimensions. Therefore P_i^p can be expressed as

$$P_i^p = \sum_{l=1}^{N(p)} a_i^l e_l, \tag{2.5}$$

or in the matrix form

$$P_i^p = ea, \tag{2.6}$$

where e is the basis function vector $[e_1, \dots, e_N]$ and a is the reconstruction coefficient vector $[a_1, \dots, a_N]^T$. Substituting (2.5) into (2.4), we then obtain

$$\frac{1}{V_{i,j}} \sum_{l=1}^{N(p)} a_i^l \int_{C_{i,j}} e_l(x, y, z) dxdydz = \bar{Q}_{i,j}, \tag{2.7}$$

Let $\bar{Q}$ denote the column vector $[Q_{i,1}, \dots, Q_{i,N}]$, Equation (2.7) can be rewritten in the matrix form

$$Ra = \bar{Q}, \tag{2.8}$$

where the reconstruction matrix

$$R = \begin{bmatrix} \frac{1}{V_{i,1}} \int_{C_{i,1}} e_1(x, y, z) dV & \dots & \frac{1}{V_{i,1}} \int_{C_{i,1}} e_N(x, y, z) dV \\ \dots & \dots & \dots . \\ \frac{1}{V_{i,N}} \int_{C_{i,N}} e_1(x, y, z) dV & \dots & \frac{1}{V_{i,N}} \int_{C_{i,N}} e_N(x, y, z) dV \end{bmatrix}. \tag{2.9}$$

The reconstruction coefficients a can be solved as

$$a = R^{-1} \bar{Q}, \tag{2.10}$$

provided that the reconstruction matrix R is nonsingular. Substituting (2.10) into (2.5) or (2.6), P_i^p is then expressed in terms of cardinal basis functions or shape functions $L = [L_1, \dots, L_N]$,

$$P_i^p = \sum_{j=1}^{N} L_j(x,y,z)\bar{Q}_{i,j} = L\bar{Q}. \tag{2.11}$$

Here L is defined as

$$L \equiv eR^{-1}, \tag{2.12}$$

which satisfies

$$\frac{1}{V_{i,j}}\int_{C_{i,j}} L_l dV = \delta_{jl}. \tag{2.13}$$

Equation (2.11) gives the functional representation of the state variable Q within the SV. Therefore the function value of Q at a quadrature point or any point (x_{rq}, y_{rq}, z_{rq}) within the i-th SV is thus simply as follows,

$$P_i^p(x_{rq}, y_{rq}, z_{rq}) = \sum_{j=1}^{N(p)} L_j(x_{rq}, y_{rq}, z_{rq})\bar{Q}_{i,j}. \tag{2.14}$$

Note that once the polynomial basis functions e_l are chosen, the shape functions L_j are solely determined by the partition of a SV cell. The shape and the partition of a SV cell, in general, can be arbitrary as long as the reconstruction matrix R is nonsingular. However, different shapes of SV cells can result in the same expression of the shape functions (in terms of a few geometric parameters) if a geometrically similar partition can be applied to them. Since the volume integral of polynomial basis in (2.9) can be carried out easily over a transformed standard tetrahedron, the shape functions L, which are universal for all SVs, can be calculated analytically and stored as a preprocessing step.

The flux integration over a face is performed using the Gauss quadrature formula

$$\int_{A_r} F \cdot \vec{n}_{A_r} dA = \sum_{q=1}^{J} w_{rq} F\left(P(x_{rq}, y_{rq}, z_{rq})\right) \cdot \vec{n}_{A_r} dA_r + O(h^p), \tag{2.15}$$

where J is the number of quadrature points on the r-th face, and w_{rq} are the Gauss quadrature weights, and (x_{rq}, y_{rq}, z_{rq}) stands for the Gauss quadrature points.

Since a discontinuous solution can exist between SVs, an approximate Riemann solver is used to find fluxes for faces on SV boundaries. Both the explicit 3rd-order TVD Runge-Kutta scheme and the implicit LU-SGS have been developed for time integration.

2.2 Idea and formulation of quadrature-free SV method

2.2.1 Motivation

In the above general SV method, the so called "residual" (spatial discretized part of Equation (2.3)) is the summation of all the CV face flux integrals for a SV cell and each face integral is given by (2.15), including SV-bounding CV faces and internal CV faces (both are referred as "sub-faces" of a SV cell). Computing residual is the major part of computer cost for high-order methods. In 3D high-order case, the partition of a SV cell (tetrahedron) can be complicated with large number of sub-faces, as shown in Figure 2.1. Therefore quite a number of Gauss quadrature points might be needed in one SV cell to compute the face integrals to the desired precision, making the 3D SV method expensive. For example, there are about 130 sub-faces for a cubic partition, and about 1100 quadrature points are needed just for one cell.

To handle the face integrals more efficiently, it is necessary to develop a quadrature-free approach for 3D application. Since volume integrals and face integrals are also involved in DG methods, a quadrature-free method was proposed by Atkins and Shu [4] for DG methods, and was tested in 1D scalar advection and 2D scalar advection and linear Euler equations. Here we extend the basic idea from [4] yet for a quadrature-free 2D SV method [46] to 3D SV. This approach replaces the large number of quadrature for face integrals (in 3D case) with many less nodal operations based on analytical shape functions. The analytical shape functions on the nodal points, which are later used for flux reconstruction in the flow solver, have been pre-computed from some symbolic software such as Mathematica. However, the implementation of the basic quadrature-free idea for 3D situation involves significant efforts including: 1) finding the complicated connectivity in 3D high-order partition of a SV cell, which includes sub-faces, nodes, sub-cells (CVs), flux directions, and orientations relative to the neighboring cells; 2) computing those complicated shape functions for each node in a SV.

2.2.2 Formulation

In the new approach, a near optimal nodal set is selected following Hesthaven and Teng [50]. Figure 2.2 shows the nodal sets used for linear, quadratic, and cubic partitions. This nodal set is then used to reconstruct a degree $p+1$ polynomial approximation for the flux vector, and then the flux integrals are computed analytically, without the need for Gauss quadrature formulas. The flux vector F is approximated in terms of the basis set $\{b_l\}$ (constructed from simple monomials),

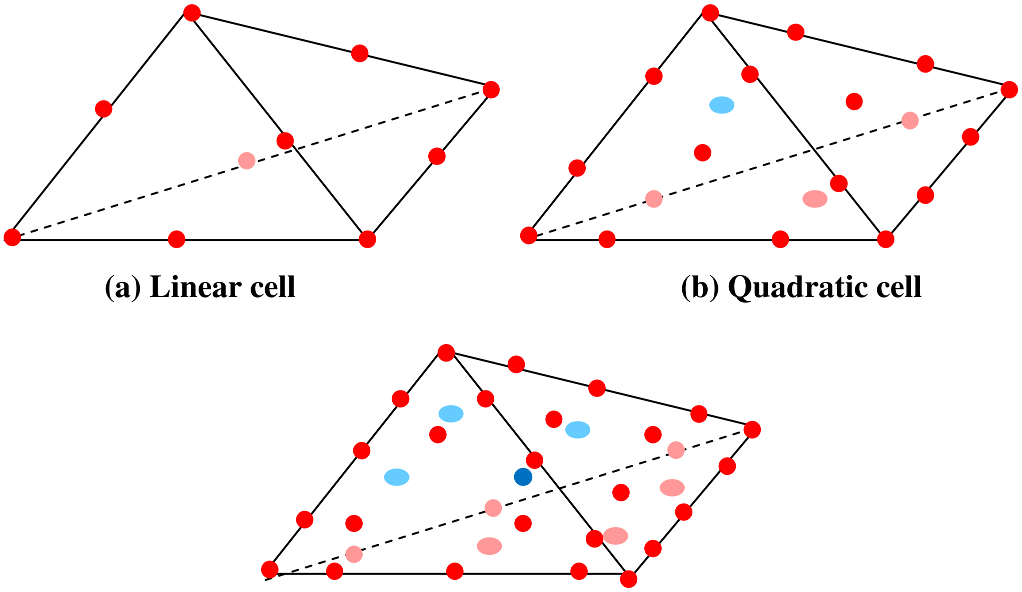
$$F = \sum_{l=1}^{M} b_l e_l + O(h^p), \tag{2.16}$$

If $F(Q)$ is linear, then $M = N(p)$; however, when $F(Q)$ is nonlinear, M must be at least $N(p+1)$ to obtain the design accuracy of order $p+1$. We prefer to use $M = N(p+1)$ for all the cases, thus $e_l(x,y,z) \in D_{p+1}$ with $D_{p+1} = Span\{e_l(x,y,z)\}_{l=1}^{N(p+1)}$. The reconstruction problem reads as follows: Given the nodal values $F_{i,n}$ on a set of nodes within the SV cell i, find $P_i^{p+1} \in D_{p+1}$ such that

$$P_i^{p+1}(x_n, y_n, z_n) = F_{i,n}, \tag{2.17}$$

where

$$P_i^{p+1}(x, y, z) = \sum_{l=1}^{M} b_l e_l(x, y, z). \tag{2.18}$$



(a) Linear cell

(b) Quadratic cell

(c) Cubic cell

Figure 2.2 Node set for a tetrahedron

Therefore a Lagrange shape functions defined by the nodal set can be found from (2.17) and (2.18). Substituting (2.17) into (2.18) yields

$$F_{i,n} = \sum_{l=1}^{M} b_l e_l(x_n, y_n, z_n), \tag{2.19}$$

with $n = 1,2, \dots, M$.

Let $\bar{F}$ denote the column vector $\left[F_{i,1}, \dots, F_{i,M}\right]^T$. Equation (2.19) can be rewritten in the matrix form

$$Sb = \bar{F}, \tag{2.20}$$

where the reconstruction matrix

$$S = \begin{bmatrix} e_1(x_1, y_1, z_1) & e_2(x_1, y_1, z_1) & \dots & e_M(x_1, y_1, z_1) \\ \dots & \dots & \dots & \dots \\ e_1(x_M, y_M, z_M) & e_2(x_M, y_M, z_M) & \dots & e_M(x_M, y_M, z_M) \end{bmatrix}. \tag{2.21}$$

The reconstruction coefficients b can be solved as

$$b = S^{-1}\bar{F}, \tag{2.22}$$

provided that the reconstruction matrix S is nonsingular. Substituting (2.22) into (2.18), P_i^{p+1} is then expressed in terms of shape functions $Z = [Z_1, \dots, Z_M]$:

$$P_i^{p+1} = \sum_{n=1}^{M} Z_n(x, y, z) F_{i,n} = Z\bar{F}, \tag{2.23}$$

here Z is defined as

$$Z \equiv eS^{-1}, \tag{2.24}$$

which satisfies

$$Z_n(x_m, y_m, z_m) = \delta_{mn}, \tag{2.25}$$

Again, this reconstruction is universal for all SVs if a nodal set is distributed in a geometrically similar manner for all SVs. The flux vector F can be computed at any point using

$$F_i(x, y, z) = \sum_{n=1}^{M} Z_n(x, y, z) F_{i,n}, \tag{2.26}$$

For the flux on each internal face, the flux integral can be computed as a weighted average of the flux evaluated at the nodal set, i.e.,

$$\begin{aligned} \int_{A_r} F \cdot \vec{n}_{A_r} dA &= \textstyle\sum_{n=1}^{M} \left(F_{i,n} \cdot \vec{n}_{A_r}\right) \int_{A_r} Z_n(x, y, z) dA + O \\ &= \textstyle\sum_{n=1}^{M} \left(F_{i,n} \cdot \vec{n}_{A_r}\right) A_r \bar{Z}_{n,A_r} + O(h^p). \end{aligned} \tag{2.27}$$

where $F_{i,n}$ is the flux vector evaluated at node n on the i-th SV cell; $\bar{Z}_{n,A_r}$ is the face-averaged value of shape function for face A_r, which is universal for all SVs if a nodal set is distributed in a geometrically similar manner, and thus can be computed during preprocessing for a standard element and then the physical face area A_r is multiplied.

Based on the fact that M is much less than J for 3D high-order partition, it is much more efficient to evaluate flux integration from (2.27) than (2.15). For a cubic partition example $(p = 3)$, only $M = 35$ nodal points are needed in quadrature-free approach (2.27), compared with the traditional SV method (2.15), where there are about 130 sub-faces and about 1100 quadrature points.

For the SV-bounding faces with normal $\vec{n}$, the Riemann flux integral can also be computed without the use of a Gauss quadrature. For example, Rusanov flux gives

$$\hat{F} \cdot \vec{n} = \frac{1}{2}[F^R \cdot \vec{n} + F^L \cdot \vec{n} - \lambda(Q^R - Q^L)], \tag{2.28}$$

where λ is the local maximum eigenvalue based the right and left cells. Integrating (2.28) on face A_r yields

$$\int_{A_r} \hat{F} \cdot \vec{n} dA = \frac{1}{2}\left[\int_{A_r} F^R \cdot \vec{n} dA + \int_{A_r} F^L \cdot \vec{n} dA - \int_{A_r} \lambda(Q^R - Q^L) dA\right]. \tag{2.29}$$

The fact that $\lambda \sim O(1)$ and $(Q^R - Q^L) \sim O(h^{p+1})$ tells us $\lambda(Q^R - Q^L) \sim O(h^{p+1})$, which is actually a high-order small term comparing with $F^R \sim P_R^{p+1} + O(h^{p+1})$ and $F^L \sim P_L^{p+1} + O(h^{p+1})$. Therefore without loss of accuracy we can use a face-centered value based on an average state, λ_{c,A_r}, to replace the local λ in (2.29). Then (2.29) becomes

$$\int_{A_r} \hat{F} \cdot \vec{n} dA = \frac{A_r}{2}\left[\bar{F}^R + \bar{F}^L - \lambda_{c,A_r}(\bar{Q}^R - \bar{Q}^L)\right], \tag{2.30}$$

where

$$\bar{F}^R = \frac{1}{A_r}\int_{A_r} F^R \cdot \vec{n} dA, \quad \bar{F}^L = \frac{1}{A_r}\int_{A_r} F^L \cdot \vec{n} dA, \tag{2.31}$$

$$\bar{Q}^R = \frac{1}{A_r}\int_{A_r} Q^R dA, \quad \bar{Q}^L = \frac{1}{A_r}\int_{A_r} Q^L dA.$$

Our numerical tests in the next section have verified the above accuracy analysis, and have shown that this quadrature-free approach preserves the accuracy of the SV method.

2.3 Local and Global time stepping

For local time stepping, the time step Δt is determined from the stability limit (CFL number) specifically for each control volume or spectral volume (cell). For control volume j on cell i,

$$\Delta t_{i,j} = \frac{CFL \cdot V_{i,j}}{\sum_r (|u_n| + c) \cdot A_r}, \tag{2.32}$$

where the control volume j is enclosed by faces A_r, and u_n is the normal velocity on the face, and c is the speed of sound on the face. Similarly local time step can be found on a SV cell as Δt_i. Local time stepping is usually used for steady flow to speed up convergence regardless to time accuracy.

For global time stepping, the time step Δt is a global constant, which is the same for all the cells in the domain. Global time stepping is needed for unsteady computations where time accuracy matters. Considering stability limit, the global time step Δt is set to be the minimum value of all the local time step sizes $\Delta t_{i,j}$,

$$\Delta t = \min_{i,j} \Delta t_{i,j}. \tag{2.33}$$

2.4 Simplified curved boundary treatment

High-order boundary cells are needed for high-order method to solve curved boundary problems, because low-order linear boundary cells with boundary line segments in 2D or planar facets in 3D induce geometry error which will eventually pollute the solution field and destroy solution accuracy, particularly for Euler equation applied on slip curved wall. In practice even second-order accuracy is not obtained and computations often diverge without any special treatment for curved boundary. One approach is to use isoparametric cells [131], which are commonly used in finite element community, to represent the curved boundary cells in consistent accuracy with solution. However, the drawback of such isoparametric cell treatment is that it necessitates separate computation and storage of reconstruction for each curved boundary cell, which brings complication of implementation and extra computer costs.

Some other treatments [62,77] choose to still use straight faces for boundary cells so that those cells on curved boundaries have the same cell reconstruction with internal cells, but the treatments manage to preserve conservation when computing the inviscid flux through curved boundary. Among them a simple and favorable approach proposed by Krivodonova and Berger [62] has been implemented here in coupled with the present quadrature-free method. This approach does not ensure the formal order of accuracy at boundary, but it does capture the major physical characteristics on slip wall boundary, that is, zero-flow in the normal direction of curved wall. The basic idea of this approach is to use "true" face normal vectors to enforce slip-wall boundary condition with respect to the physical curved boundary, as shown in Figure 2.3. A true normal might not be perpendicular to the straight mesh face on boundary.

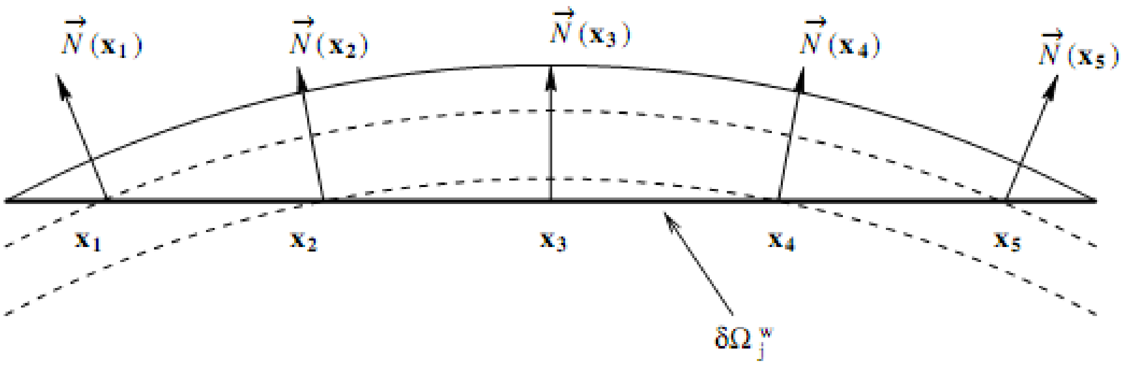


Figure 2.3 "True" face normal for curved boundary

It is not practical to find true face normal vectors in real-world applications. So in Figure 2.3 a set of concentric spheres that intersect on the straight cell boundary face are applied to find the normal of the circle for a specific point on the straight mesh face to use. For example, between Point#2 and #4 the flow field is generated by the curved circular wall as if the flow goes out the straight boundary from Point#2 and goes into it through Point#4, thus maintains zero-flow conservation on the straight mesh boundary. Once the "true" normal is found, the remained procedures of the solver almost keep unchanged, without extra complication or memory storage for curved boundary implementation.

2.5 Numerical tests

The 3D linear advection equation was used to analyze the accuracy of the present method by comparing with the exact solution. The test results for 2D Euler equations with a circle and NACA0012 airfoil were also presented here. The tests for 3D Euler equations can be found in [139]. For all cases the Rusanov numerical flux is applied. Both the explicit 3rd-order TVD Runge-Kutta scheme and implicit LU-SGS scheme are used for time integration in all cases, and the time step Δt used is small enough so that the numerical errors are dominated by the spatial discretization, independent of the time step. Given the initial values at the nodal set, the CV-averaged solution values were initialized by using the CV-averaged node-based shape functions, without the need of Gauss quadrature for preparing the CV-averaged initial conditions. The partition of the SV cells has been taken from [75,21].

2.5.1 Accuracy Study with 3D Steady Linear Advection

The governing equation for this problem is

$$\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}+\frac{\partial u}{\partial z}=0, \tag{2.34}$$

with the boundary conditions as $u = \sin\,[\pi(2x - y - z)]$ for inflow; extrapolation of u for outflow. The above equations describe a steady sinusoidal wave with unit wave speed in all three Cartesian directions. It is obvious that the exact solution for this problem is

$$u = \sin[\pi(2x - y - z)]. \tag{2.35}$$

We solved (2.34) numerically by using pseudo-time integration as follows to find its steady-state solution,

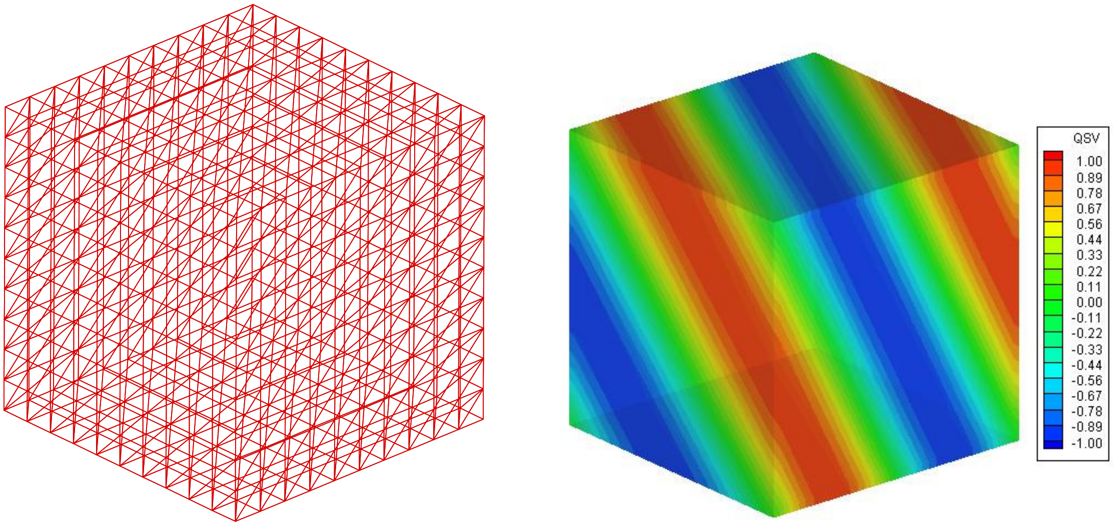
$$\frac{\partial u}{\partial \tau}+\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}+\frac{\partial u}{\partial z}=0. \tag{2.36}$$

We generated a sequence of regular unstructured grids (Figure 2.4) in a cubic domain. The domain size used here is $[0,1] \times [0,1] \times [0,1]$. First the cubic domain is represented by $N \times N \times N$ cubic cells, and then each cell is cut into 6 tetrahedra. Taking $N = 10,20,40$, we obtained a sequence of unstructured grids for the accuracy analysis under grid refinement. In the following figures, cell size $I = N\sqrt[3]{6}$. The solution is taken as converged when the L_2 norm of the residual is reduced to machine zero.

Figure 2.5 shows the error norms (L_1 and L_∞) of the present numerical solution comparing with the exact solution. The averaged slope is about 1.98 for the 2nd-order cases, and 2.73 for the 3rd-order cases. The results show that the nearly optimum order of accuracy is attained for the 2nd-order and 3rd-order cases, respectively. Based on that we are satisfied with our results of order of accuracy for the 2nd –order and 3rd-order cases. Figure 2.6(a) gives the contour of the solution on the fine mesh for the 3rd-order method, which shows the advection solution in a clear wave pattern. The number of DOFs used is 3.84×10^6.

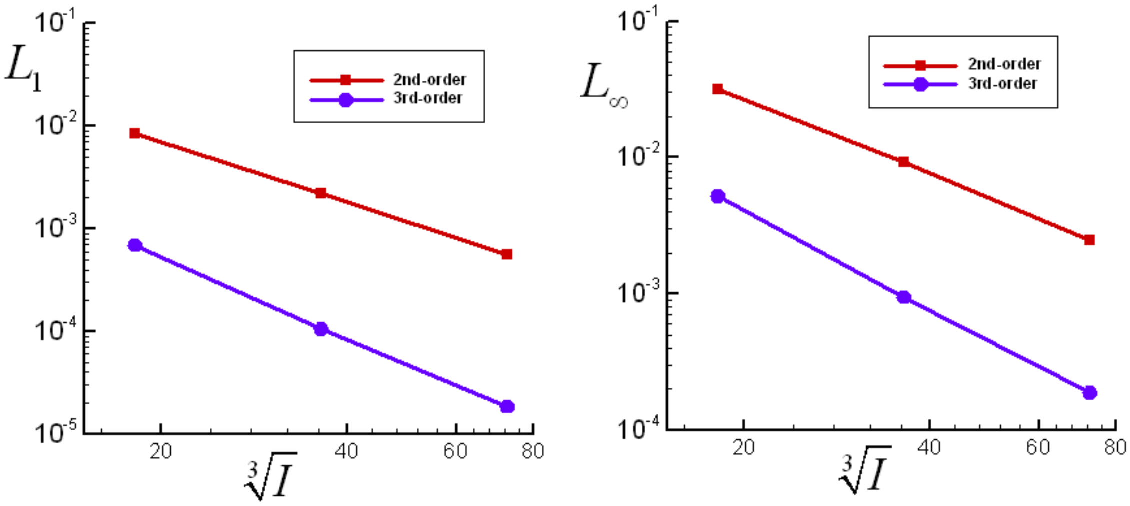
2.5.2 2D flow around a circle

The inviscid flow around a circle with $Mach = 0.2$ was computed to test the effectiveness of the curved boundary treatment in the present quadrature-free method for solving Euler equation. A coarse mesh with $16 \times 8 \times 2$ triangles and a fine mesh with $32 \times 16 \times 2$ triangles are used. Here only the coarse grid is shown in Figure 2.8(a). The implicit LU-SGS method is developed here for time integration. Figure 2.9 compares the converged solutions for 2nd-order, 3rd-order, and 4th-order cases on the coarse mesh and the 2nd-order on the fine mesh. It is shown that the higher-order method indeed gives better results as expected, in terms of smoother contour lines and more symmetric flow field. The simplified curved boundary treatment works compatibly with the present high-order method without loss of accuracy from the curved boundary (at least visually). Also the 4th-order scheme using the coarse mesh with less DOFs (10240 DOFs) shows better result than the 2nd-order scheme (actually FV scheme) using the fine mesh with more DOFs (12288 DOFs). The benefit of high-order method is manifested clearly here.



(a) Mesh ($10 \times 10 \times 6$ tetrahedra) **(b) Contours of converged solution**

Figure 2.4 3D linear advection problem.

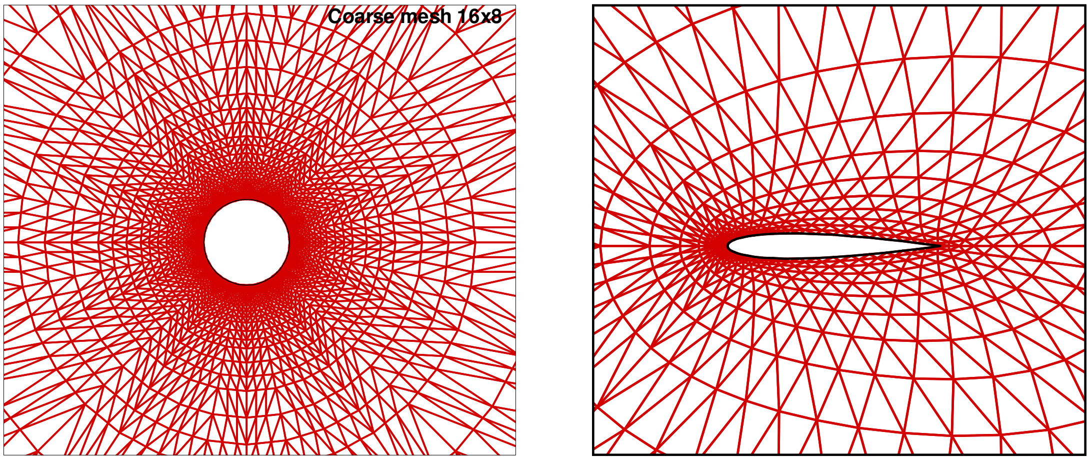


(a) L_1 norm of error **(b) L_∞ norm of error**

Figure 2.5 Accuracy study for linear advection problem.

2.5.3 2D flow around NACA0012 airfoil

Since the curved boundary treatment we use here approximates the curved boundary segments with arc, it is not enough to test only the ideal cases, i.e., circles or spheres. Here a 2D NACA0012 airfoil is used to further test the present quadrature-free SV method with the simplified curved boundary treatment. A coarse mesh with 1536 cells, as in Figure 2.8(b), was generated for subsonic flow with $Mach = 0.4, \alpha = 5^{\circ}$ by using 2nd-order, 3rd-order, and 4th-order schemes. The implicit LU-SGS method is developed here for time integration. From the converged solution shown in Figure 2.10, we see that better solution resolution is obtained from higher order scheme. Therefore the present quadrature-free SV method works effectively with the aforementioned simple curved boundary treatment preserving high-order accuracy on the general non-circular curved boundaries.



(a) Coarse mesh for circle: 256 cells **(b) Mesh for NACA0012 airfoil: 1536 cells**

Figure 2.6 Mesh for 2D Euler equation.

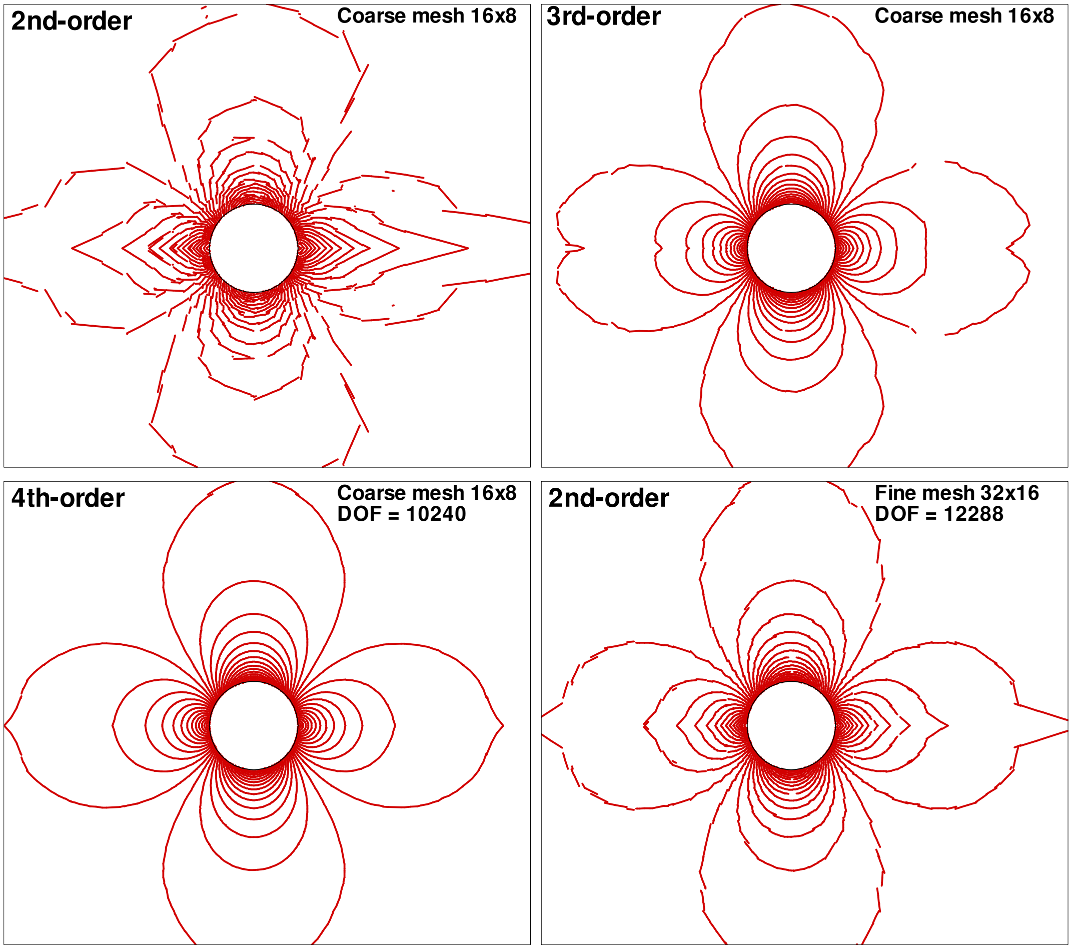
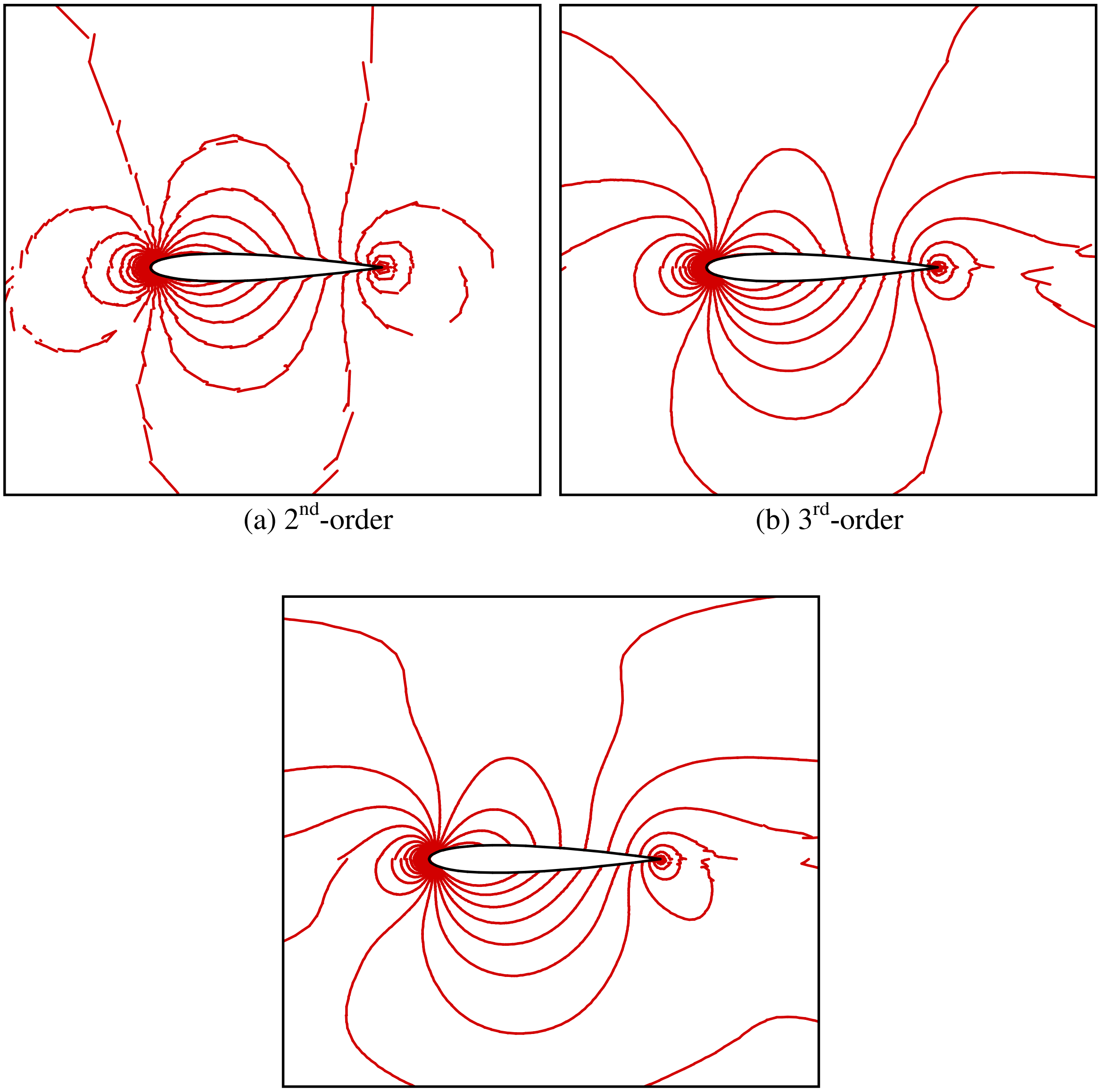


Figure 2.7 Mach contours for 2D inviscid flow around a circle $(Mach = 0.2)$.



(a) 2nd-order

(b) 3rd-order

(c) 4th-order

Figure 2.8 Mach contours for inviscid flow around NACA0012 airfoil.
$(Mach = 0.4, \alpha = 5^{\circ})$

2.6 Conclusions

The quadrature-free spectral volume method has been extended to 3D conservation laws including linear scalar advection equation and nonlinear Euler equations. From the nodal values on a selected set of optimized and geometrically similar nodes within each SV, we found a set of universal shape functions for face integrals, which avoids the use of quadrature formulas without losing the properties of compactness and robustness that are inherent to the SV method. In high-order computations for 3D problems, it has been shown that this new approach greatly reduces the number of flux calculations per SV that required in the traditional SV method. Several representative inviscid cases that have analytical exact solutions were used to test the new quadrature-free SV method. It has been found that the near optimum order of accuracy can be obtained in both L_1 and L_∞ norms for both 2nd and 3rd-order simulations in 3D. This shows that the new approach preserves the stability and accuracy. In addition, the test case of inviscid flow over a circle, NACA0012 airfoil and a sphere demonstrates the ability of the new approach to effectively handle curved boundaries using a simple curved wall treatment.

CHAPTER 3 PARAMETER-FREE GENERALIZED MOMENT LIMITER

In this chapter we present a parameter-free high-order generalized moment limiter for arbitrary mesh. Firstly a parameter-free discontinuity marker, which is the key component of the limiter, is proposed with the advantage of detecting only the discontinuities and excluding the smooth extrema on arbitrary mesh without involving any user-specified free parameter. Secondly a generalized moment limiter is designed for arbitrary mesh and all kind of high-order methods in general. The present high-order limiter is naturally compact and efficient, suitable for massively parallel computing. Since we use the Spectral Difference (SD) method to test the new limiter here and the new implicit line solver in Chapter 4, a brief review of SD is also included in this chapter. A literature survey is given below for research background of this old and new problem.

3.1 Background and motivation

A nonlinear hyperbolic conservation law can generate discontinuities even if the initial solution is smooth. A significant computational challenge with a nonlinear hyperbolic conservation law is the resolution of such discontinuities, which has been a very active area of research for over four decades. However, any linear scheme higher than first order accuracy cannot generate monotonic solutions, according to the Godunov theorem [42]. That means linear schemes of 2nd-order and higher will produce spurious oscillations near discontinuities due to the so-called Gibbs phenomenon, which can result in numerical instability and non-physical data, such as negative pressure or density. Early research work on shock-capturing relied on numerical diffusion to smear the discontinuities so that they can be captured as part of the numerical solution [125,68,78,57]. Besides the existence of user-defined parameters, the historical drawback of the artificial viscosity approach is that the added dissipation terms cannot guarantee only locally effective, and often they are too dissipative in other flow regions. Later, another type of approach was developed based on flux limiting, which introduced numerical diffusion implicitly. Flux-limiting adjusts the fluxes going in and out of a computational cell with the goal of reducing or removing spurious oscillations. Pioneering work in flux limiting includes the FCT [16], the MUSCL and related methods [119,121,96,69,49], and TVD methods [48,140]. However, the flux-limiting and TVD methods suffered from accuracy-degradation to first-order at local extrema in smooth regions.

Now the problem is: how to sharply capture discontinuity while preserving high-order accuracy elsewhere in the solution field, particularly at smooth extrema, on arbitrary unstructured meshes?

High-order (3rd-order and higher) shock-capturing algorithms have the potential to obtain sharp non-oscillatory shock transition and simultaneously preserve accuracy in smooth regions. The challenge of producing oscillation-free numerical solutions is tougher for high-order methods than for lower order ones because of much reduced numerical dissipation. The artificial viscosity method has been improved [114,31,39] to minimize undesirable dissipation by using a spectral vanishing viscosity approach based on high-order derivatives of the strain rate tensor, though there still exist user-defined parameters that can be mesh or problem dependent. The ENO [49] and WENO methods [58] used the idea of adaptive stencils in the reconstruction procedure based on the smoothness of the local numerical solution. However, due to a lack of compactness, the implementation of both ENO and

WENO methods is complicated on arbitrary unstructured meshes, especially for 3D problems. The MP5 [113] scheme preserves monotonicity very well by using a fixed stencil of 5 cells and a relatively simple limiting procedure, but it is not easy to be extended to 2D or 3D on arbitrary meshes.

In order to be compatible with the modern compact unstructured-grids based high-order methods such as DG, SV, and SD, it is necessary to require that the designed limiter for those methods be compact and suitable for arbitrary meshes. There have been many notable developments in limiters for high-order methods in the last decade. Many of the limiters employ the so-called “troubled cell” (TC) approach, in which “oscillatory” cells are marked first, and the solutions in these cells are re-generated to remove or reduce the oscillations satisfying certain criteria such as mean-preserving. The idea is first developed in [27], and then further extended in [14]. In [27,29], a limiter developed for the finite volume method [7] was used. The moment limiter developed in [14] can be viewed as the generalization of the minmod limiter [121] to higher order derivatives or moments. The central DG scheme proposed in [72] is a further generalization of the MUSCL scheme and the moment limiter. Other more recent developments include the use of WENO [92] and Hermite WENO [91,77] schemes to generate the reconstruction in “troubled cells”. High-order limiters based on artificial viscosity have also been investigated by various researchers [55,89]. In the present study, our focus is on the TC approach.

There are two major components in the TC approach: the marking or detection of “troubled cells”, and the data limiting (or remapping) in these cells. In developing the present moment-based limiter, we set to achieve several goals: 1. free of user adjustable parameters; 2. capable of preserving accuracy at smooth regions including smooth extrema;

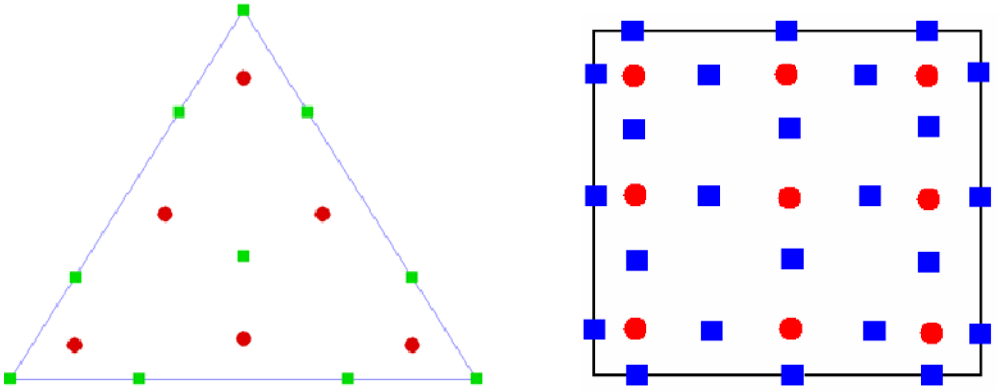
3. compact and efficient for arbitrary unstructured meshes. The requirement of no-user adjustable parameters is very important for a general purpose production-type flow solver, which can be applied to a wide variety of problems. If a limiter's success hinges on a "suitable" parameter which depends on the solution, the mesh and the order of accuracy, the limiter will more likely fail than succeed in real world applications. In the present study, we compare several markers investigated in [90], namely, the minmod TVB marker [27], the KXRCF marker developed by Krivodonova et al. in [64], and the Harten marker [47], with the present parameter-free accuracy-preserving TVD marker. For the limiter step, we extend the approach in [63] and [72] to arbitrary unstructured meshes in an efficient manner. There are important differences between the present moment limiter and those in [63] and [72]. Numerical results show that the present limiter can preserve accuracy at smooth regions, while capturing discontinuities. Now the present limiting technique has been extended to a p-adaptive DG method by Kuzmin [65].

3.2 Review of the spectral difference method

Consider the following hyperbolic conservation law,

$$\frac{\partial Q}{\partial t} + \nabla \cdot F = 0 \tag{3.1}$$

on domain $\Omega \times [0, T]$ and $\Omega \subset \mathrm{R}^3$ with the initial conditions within Ω and appropriate boundary conditions on $\partial\Omega$. The conservative solution variable Q can be a scalar or a vector, and the generalized flux F can be a scalar, vector, or even tensor. In the case of the Euler equations, Q is the vector of conservative variables. Domain Ω is partitioned into non-overlapping triangular or quadrilateral cells (or elements). In the SD method, two sets of points, i.e., the solution points and flux points are defined in each element. The solution points are the locations where the nodal values of the state variable Q are specified. Flux points are the locations where the nodal values of fluxes are computed. The DOFs in the SD method are the conservative variables at the solution points. Figure 3.1 displays the placement of solution and flux points for the third-order SD schemes on triangular and quadrilateral cells.



(a) Triangular mesh **(b) Quadrilateral mesh**

Figure 3.1 Solution (red solid circles) and flux points (green/blue solid squares).

Given the solution $Q_{j,i}$ at the j-th solution point within cell i (denoted as $\vec{r}_{j,i}$), an element-wise degree p polynomial can be constructed using Lagrange-type polynomial base,

$$P_i^p(\vec{r}) = \sum_{j=1}^{m} L_{j,i}(\vec{r}) Q_{j,i}, \tag{3.2}$$

where $L_{j,i}(\vec{r})$ are the Lagrange shape functions. With (3.2), the solutions at the flux points can be computed. Since the solutions are discontinuous across element boundaries, the fluxes at the element interfaces are not uniquely defined. Obviously, in order to ensure conservation, the normal component of the flux vector on each face should be identical for the two cells sharing the face. A one dimensional approximate Riemann solver (for example, Roe flux in this paper) is then employed in the face normal direction to compute the common normal flux $\hat{F}(Q^{-}, Q^{+}, \vec{n})$. Since the tangential component of the flux does not affect the conservation property, we have the complete freedom to choose it at the face flux points. Let the unit vector in the tangential direction be $\vec{l}$ as shown in Figure 3.2.

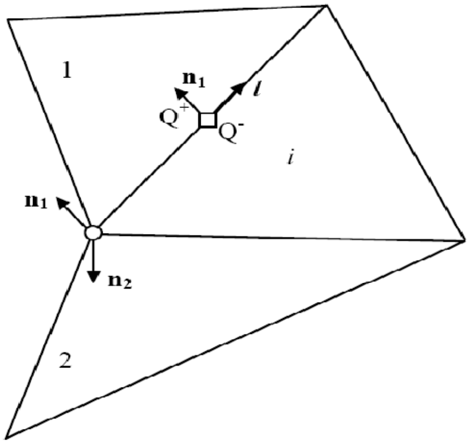


Figure 3.2. Flux computation for triangular or tetrahedral mesh

Here we offer two possibilities. One is to use a unique tangential component by averaging the two tangential components from both sides of the face,

$$F_l = F_l(Q^{-}, Q^{+}, \vec{l}) = \frac{1}{2}\{[\vec{F}(Q^{-}) + \vec{F}(Q^{+})] \cdot \vec{l}\}. \tag{3.3}$$

The other option is to use their own tangential components separately, allowing discontinuous tangential components on the element interfaces. For cell *i*, the tangential component is $\vec{F}(Q^{-}) \cdot \vec{l}$, and for its neighbor, $\vec{F}(Q^{+}) \cdot \vec{l}$. For a corner flux point in cell *i*, two faces (viewed from cell *i*) share the corner point, as shown in Fig. 2. The full flux vector at corner point can be uniquely determined from the two normal Riemann flux components $\hat{F}_1 = \hat{F} \cdot \vec{n}_1$ and $\hat{F}_2 = \hat{F} \cdot \vec{n}_2$. In spite that the fluxes at a cell corner point do not have the same value for all the cells sharing the corner, local conservation is guaranteed because neighboring cells do share a common normal flux at all flux points. Once the fluxes at all flux points are re-computed, they are used to form a *p*+1 degree polynomial,

$$\vec{P}_i^{p+1}(\vec{r}) = \sum_{l=1}^{m_{p+1}} Z_{l,i}(\vec{r})\vec{F}_{l,i}, \tag{3.4}$$

where $\vec{F}_{l,i} = \vec{F}(\vec{r}_{l,i})$, and $Z_{l,i}(\vec{r})$ are the set of Lagrange shape functions defined by $\vec{r}_{l,i}$. The divergence of the flux at the solution points can be easily computed as,

$$\nabla \cdot \vec{P}_i^{p+1}(\vec{r}) = \sum_{l=1}^{m_{p+1}} \nabla Z_{l,i}(\vec{r}) \cdot \vec{F}_{l,i}, \tag{3.5}$$

Finally the semi-discrete scheme to update the solution unknowns can be written as,

$$\frac{dQ_{j,i}}{dt} + \sum_{l=1}^{m_{p+1}} \nabla Z_{l,i}\left(\vec{r}_{j,i}\right) \cdot \vec{F}_{l,i} = 0. \tag{3.6}$$

The SD method for quadrilateral or hexahedral grid is identical to the staggered grid multi-domain spectral method [60,61]. It is particularly attractive because all the spatial operators are one-dimensional in nature. In the original staggered-grid method, the solution and flux points are the Chebyshev-Gauss and Chebyshev-Gauss-Lobatto points. Recently, it was found [53,117] that these flux points results in a weak instability. New stable fluxes points were suggested in [53,117]. In the present study, we employ the Legendre-Gauss points plus the two end points as the flux points, as suggested in [53]. In an actual implementation, each physical element (possibly curved) is first transformed into a standard element (square). The governing equations are also transformed from the physical space to the computational space as follows,

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0, \qquad (3.7)$$

where

$$\begin{bmatrix} \widetilde{F} \\ \tilde{G} \end{bmatrix} = |J| \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix}, \qquad \tilde{Q} = |J|Q$$

The Lagrange interpolation shape functions in one direction for the conservative solution variable Q and fluxes Lagrange can be written as follows, respectively,

$$h_i(X) = \prod_{s=1,s\neq i}^{N} \left(\frac{X - X_s}{X_i - X_s}\right); \; l_{i+1/2}(X) = \prod_{s=0,s\neq i}^{N} \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}}\right) \qquad (3.8)$$

The reconstructed solution for the conservative variables in the standard element is just the tensor products of the three one-dimensional polynomials,

$$\tilde{Q}(\xi, \eta) = \sum_{j=1}^{N} \sum_{i=1}^{N} \tilde{Q}_{i,j} h_i(\xi) h_j(\eta). \qquad (3.9)$$

Similarly, the reconstructed flux polynomials take the following form:

$$\tilde{F}(\xi, \eta) = \sum_{j=1}^{N} \sum_{i=0}^{N} \tilde{F}_{i+1/2,j} l_{i+1/2}(\xi) h_j(\eta), \qquad (3.10)$$

$$\tilde{G}(\xi, \eta) = \sum_{j=0}^{N} \sum_{i=1}^{N} G_{i,j+1/2} h_i(\xi) l_{j+1/2}(\eta). \qquad (3.11)$$

For the inviscid flux, a Riemann solver is employed to compute a common flux at the interfaces to ensure conservation and stability. Time integration is done by using either explicit TVD or SSP Runge-Kutta scheme [101,102] or an implicit BLU-SGS scheme [109].

3.3 Evaluation of several existing troubled cells markers

In this section, we review and evaluate several troubled-cell detecting methods found in the literature. We present a parameter-free TVD marker in next section. Qiu and Shu [90] investigated seven markers currently used in the CFD community, and found that the minomd TVB marker [27], the marker developed by Krivodonova et al. named KXRCF in [64], and the Harten [47] marker are the best three among the seven markers they studied based on the amount of spurious oscillations in the solution, and the total number of cells marked. These three markers are chosen in the current study, and are evaluated next.

Consider the following 1D scalar conservation law,

$$\begin{cases} u_t + f(u)_x = 0, x \in \Omega, \\ \quad u(x,0) = u_0(x). \end{cases} \tag{3.12}$$

The computational domain Ω is partitioned into *N* cells with $p+1$ solution points and $p+2$ flux points in each cell. In the following description, h_i, $\bar{u}_i$, and $u_{j,i}$ denote the mesh size of cell i, the average solution and the value of the reconstructed solution polynomial at the j-th flux point of the i-th cell, respectively.

3.3.1 Minmod TVB Marker

A user specified parameter *M* is chosen, which is of the order of the solution's second derivative in a smooth region. Then the differences between the solutions at the cell interfaces (from left and right side, respectively) and the cell-averaged solution are examined. Denote these differences $\Delta u_{i,L} = \bar{u}_i - u_{1,i}$ and $\Delta u_{i,R} = u_{p+2,i} - \bar{u}_i$. If the following inequalities are satisfied,

$$|\Delta u_{i.L}| \leq Mh_i^2 \ and \ |\Delta u_{i.R}| \leq Mh_i^2 \tag{3.13}$$

the solution in cell i is considered smooth, and thus the cell is NOT a troubled cell. Otherwise, compute the following quantities,

$$\Delta\tilde{u}_{i,L} = minmod\left(\Delta u_{i,L}, \overline{u}_i - \overline{u}_{i-1}, \overline{u}_{i+1} - \overline{u}_i\right), \tag{3.14}$$

$$\Delta\tilde{u}_{i,R} = minmod\left(\Delta u_{i,R}, \overline{u}_i - \overline{u}_{i-1}, \overline{u}_{i+1} - \overline{u}_i\right), \tag{3.15}$$

where the $minmod$ function is defined as

$$minmod(a_1, a_2, \dots, a_n) \\ = \begin{cases} s \cdot \min\limits_{1\le k\le n} |a_k| & if\ sign(a_1) = sign(a_2) = \cdots = sign(a_n) = s \\ 0 & otherwise. \end{cases} \tag{3.16}$$

If either $\Delta u_{i,L}$ or $\Delta u_{i,R}$ is modified in (3.14) or (3.15), i.e., $\Delta\tilde{u}_{i,L} \neq \Delta u_{i,L}$ or $\Delta\tilde{u}_{i,R} \neq \Delta u_{i,R}$, the cell is marked as a troubled cell.

Equations (3.14) and (3.15) are similar to the MUSCL scheme [121] in spirit, but less restrictive. In order to explain this, assume the solution to be linear with a slope of S_i in cell i. Then we have

$$\Delta u_{i,L} = \Delta u_{i,R} = S_i h_i/2. \tag{3.17}$$

Define two more slopes using

$$S_{i+1/2} = \frac{\overline{u}_{i+1} - \overline{u}_i}{x_{i+1} - x_i}, \qquad S_{i-1/2} = \frac{\overline{u}_i - \overline{u}_{i-1}}{x_i - x_{i-1}}, \tag{3.18}$$

then the following equation is equivalent to (3.14) and (3.15),

$$\tilde{S}_i = minmod\left(S_i, S_{i-1/2}\frac{(h_{i-1} + h_i)}{h_i}, S_{i+1/2}\frac{(h_i + h_{i+1})}{h_i}\right) \tag{3.19}$$

where $\tilde{S}_i$ is the limited slope. We have used $x_{i+1} - x_i = (h_i + h_{i+1})/2$ and $x_i - x_{i-1} = (h_{i-1} + h_i)/2$ in (3.19).

If the mesh is uniform, then the factors

$$\frac{h_{i-1}+h_i}{h_i}=\frac{h_i+h_{i+1}}{h_i}=2;$$

If the mesh is extremely non-uniform, the above factors can approach 1.

In practice, we could use a factor $\beta \in (1,2]$ to represent the above factors in case of general non-uniform mesh, and (3.19) can be expressed as,

$$\tilde{S}_i = minmod\left(S_i, \beta\frac{\overline{u}_i-\overline{u}_{i-1}}{x_i-x_{i-1}}, \beta\frac{\overline{u}_{i+1}-\overline{u}_i}{x_{i+1}-x_i}\right) \tag{3.20}$$

A larger β gives rise to the fewer number of cells to be marked, yet with the price of possibly missing some troubled cells that need data limiting. A good compromise is $\beta = 1.5$. Obviously if the solution is locally linear on one cell, then the cell is not marked because $S_i = \frac{\overline{u}_i-\overline{u}_{i-1}}{x_i-x_{i-1}} = \frac{\overline{u}_{i+1}-\overline{u}_i}{x_{i+1}-x_i}$.

As pointed out in [90], $M(> 0)$ is a free parameter, which depends on the solution of the problem. For scalar problems it is possible to estimate M if the solution is smooth [27] (M is proportional to the second derivative of the initial condition at smooth extrema). However it is more difficult to estimate M for the systems case, such as the Euler and N-S equations. If M is chosen too small, more cells than necessary will be marked as troubled cells. If M is chosen too large, spurious oscillations may appear.

3.3.2 KXRCF Marker

In [64] Krivodonova, Xin, Remacle, Chevaugeon, and Flaherty proposed a shock-detection technique based on DG's super-convergence property at the outflow boundaries of an element in smooth regions. This method was termed as the KXRCF marker. The boundary of a cell, ∂I_i, can be partitioned into two portions: the inflow boundary ∂I_i^- where flow goes into the cell, and the outflow boundary ∂I_i^+ where flow exits the cell. In the 1D case, if the wave speed $f'(u)$ is positive at the left interface, then the left face $(x_{i-1/2})$ is an inflow boundary; otherwise, the left face is an outflow boundary. The right face can be classified in exactly the opposite fashion. In an actual implementation, we use the averaged wave speed from both sides of a face to determine if it is an inflow or outflow boundary. The KXRCF marker checks the solution on the inflow boundary to determine troubled cells. Without loss of generality, let's assume the inflow boundary is the left interface for cell i. Then compute the following quantity L_i,

$$L_i = \frac{\left|u_{1,i} - u_{p+2,i-1}\right|}{h_i^{(p+2)/2}|\overline{u}_i|}. \tag{3.21}$$

If $L_i > 1$, then cell i is marked as a troubled cell. Note that since DG's super-convergence property occurs only in a smooth region, it is possible that the KXRCF marker excessively mark some cells in continuous but not smooth regions.

3.3.3 Harten/Modified Harten Marker

The Harten marker was originally developed in [47] and further modified in [90]. Here is the basic idea. First extend the reconstructed solution polynomials from the neighboring cells $u_{i-1}(x)$ and $u_{i+1}(x)$ into cell i. Then compute the differences between the average extended polynomials and the average of cell i. In 1D, a jump (discontinuity) within cell i can cause one extension above the current cell average and the other below the current cell average. Therefore the Harten marker can be formulated as follows. Compute

$$F_{i(z)} = \frac{1}{h_i}\left\{\int_{x_{i-1/2}}^{z} u_{i-1}(x)dx + \int_{z}^{x_{i+1/2}} u_{i+1}(x)dx\right\} - \overline{u}_i. \tag{3.22}$$

If

$$F_i\left(x_{i+\frac{1}{2}}\right)\cdot F_i\left(x_{i-\frac{1}{2}}\right)\leq 0, \tag{3.23}$$

then a discontinuity possibly exists within cell i. To improve its performance at smooth extrema, the cell-averaged degree p derivatives between the neighboring cells and the current cell are also compared. Therefore the marker is expressed as,

if

$$\begin{gathered} F_i\left(x_{i+\frac{1}{2}}\right)\cdot F_i\left(x_{i-\frac{1}{2}}\right)\leq 0, \\ \text{and} \\ \left|\overline{u}_i^{(p)}\right| > \theta\left|\overline{u}_{i-1}^{(p)}\right| \; and \; \left|\overline{u}_i^{(p)}\right| > \theta\left|\overline{u}_{i+1}^{(p)}\right| \end{gathered} \tag{3.24}$$

cell i is marked as a troubled cell. We take the same value for the constant θ $(= 1.5)$ in the numerical tests as in [90].

We can make the following observations regarding the Harten marker. When the polynomial degree p is high, the extension of the reconstructed solution polynomials from the neighboring cells might be strange and unexpected near a discontinuity, and may fail to mark a shock, as shown in Fig. 7. In this case, the extended polynomials from both sides have cell averaged solutions larger than the current cell. Therefore this strategy may fail to mark a discontinuity in a high-order scheme. The Harten marker is difficult to implement for unstructured grids in multiple dimensions.

3.3.4 Drawbacks with the above markers

To illustrate the performance of the above three markers, examples of both smooth and discontinuous solution profiles have been used:

1) A smooth sine function, $u = \sin(2\pi x)\,,\ \ 0 \le x \le 1;$

2) A combination of smooth and discontinuous profiles: a smooth Gaussian, a square pulse, a triangle, and half an ellipse [63], which is defined as,

$$u_0(x) = \frac{G(x,\beta,z-\delta) + G(x,\beta,z+\delta) + 4G(x,\beta,z)}{6}, \qquad if\ -0.8 \le x \le -0.6;$$

$$u_0(x) = 1, \qquad if\ -0.4 \le x \le -0.2;$$

$$u_0(x) = 1 - |10(x-0.1)|, \qquad if\ \ 0. \le x \le 0.2;$$

$$u_0(x) = \frac{F(x,\alpha,a-\delta) + F(x,\alpha,a+\delta) + 4G(x,\alpha,z)}{6}, \qquad if\ \ 0.4 \le x \le 0.6;$$

$$u_0(x) = 0, \qquad otherwise.$$

Here $G(x,\beta,z) = e^{-\beta(x-z)^2},\ F(x,\alpha,a) = \sqrt{\max(1-\alpha^2(x-a)^2, 0)},$

$$a = 0.5, z = -0.7, \delta = 0.005, \alpha = 10, \beta = log2/(36\delta^2).$$

3) An oscillating shock profile obtained when solving nonlinear hyperbolic equations.

The marked cells for the above profiles are plotted in the following figures, on which the solid black lines stand for the initial profile, and the elevated red squares represent the troubled cell. The performances and drawbacks for the above three marker are evaluated based on those results.

The Minmod TVB marker works well for the scalar cases, as shown in Figure 3.3(a) and 3.3(b), where no cell is marked as troubled cell for the smooth sine wave, and only the cells at the discontinuity region are marked as troubled cells. Here we estimated M from [27] by computing the maximum absolute value of the second derivatives of the initial solution in smooth regions for each of the two cases. However, for the complex oscillating shock profile case in Figure 3.4, it is difficult to give good estimation of M from the profile. It appears that

$M = 40$ is good that only the two cells at the discontinuity are marked as troubled cells, but we got this $M = 40$ by ad hoc testing. For the system cases such as Euler and Navier-Stokes equations, it is more difficult to estimate M.

The KXRCF marker detects the discontinuities as shown in Figure 3.5(b) and Figure 3.6. It also works well for the smooth sine wave case in Figure 3.5(a) as well as the smooth Gaussian extremum in Figure 3.6(a) (see the close-up view in Figure 3.6(b)), where no troubled cell at the local smooth extrema is marked. This is expected because the KXRCF marker is exactly based on the super-convergence property on the elements' outflow boundaries in smooth regions. However, in continuous but not smooth regions, such as the vicinity of $x = -0.8$ in Figure 3.5(b) or $x = -0.16$ in Figure 3.6(a), the KXRCF marker excessively marks the cells in those continuous regions as troubled cells.

The Modified Harten marker gives good results in the smooth sine wave case, as shown in Figure 3.7. The results for the discontinuous profile case are acceptable, but its performance is sensitive to the interpolation order of polynomial as shown in Figure 3.8, where some more cells are marked when $p = 5$ than the $p = 2$ case. This sensitivity can cause a serious problem in the high-order cases as shown in Figure 3.9, where the necessary condition (3.23) of the Harten marker fails to mark the shock cell. This is because that the extensions of the solution polynomials from the neighboring cells can become large in the current cell, and the integral values from the left and the right cells are both positive, i.e.

$$F_i(x_{i-1/2}) = 1.886, \ \ F_i(x_{i+1/2}) = 7.22.$$

That is why the Modified Harten marker fails in this typical case.

The above testing results show some critical drawbacks for the three markers, as summarized below:

1) The free parameters in the Minmod TVB marker can have decisive effects on the performance of the marker.

2) The KXRCF marker can mark too many cells in continuous regions as troubled cells.

3) The Harten marker can fail to detect a shock at a high-order setting, due to the unexpected polynomial extensions from the neighboring cells. In addition, the Harten marker is difficult to implement in 2D and 3D.

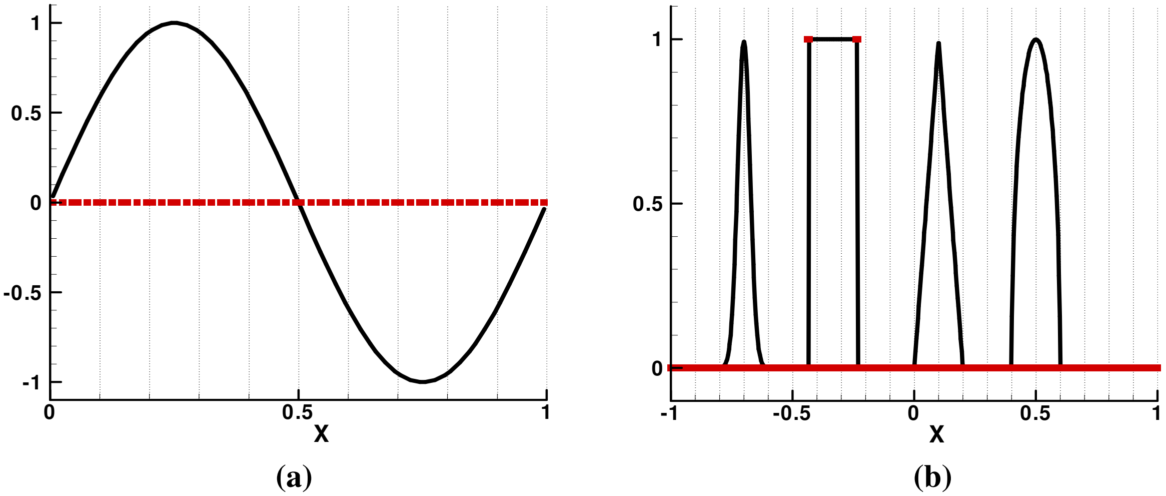
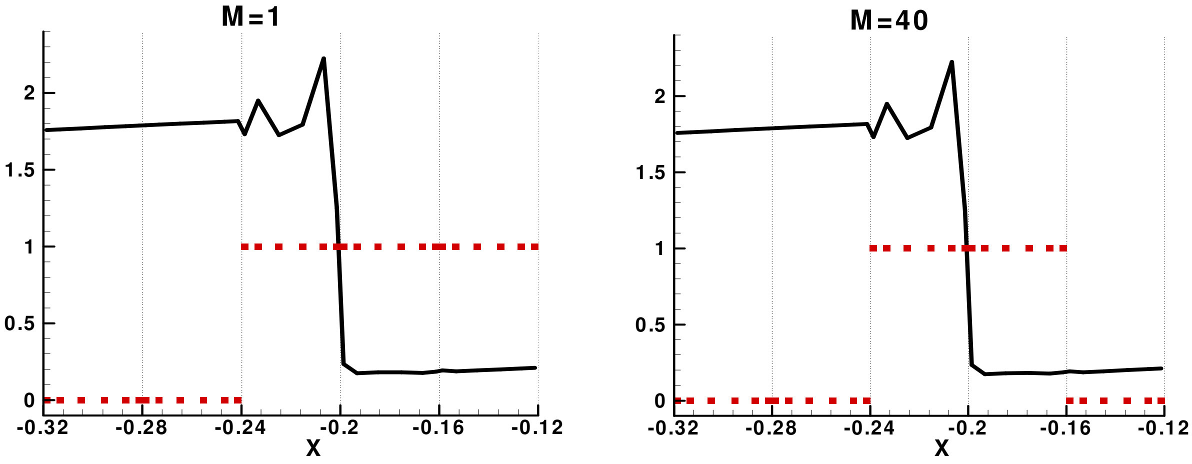
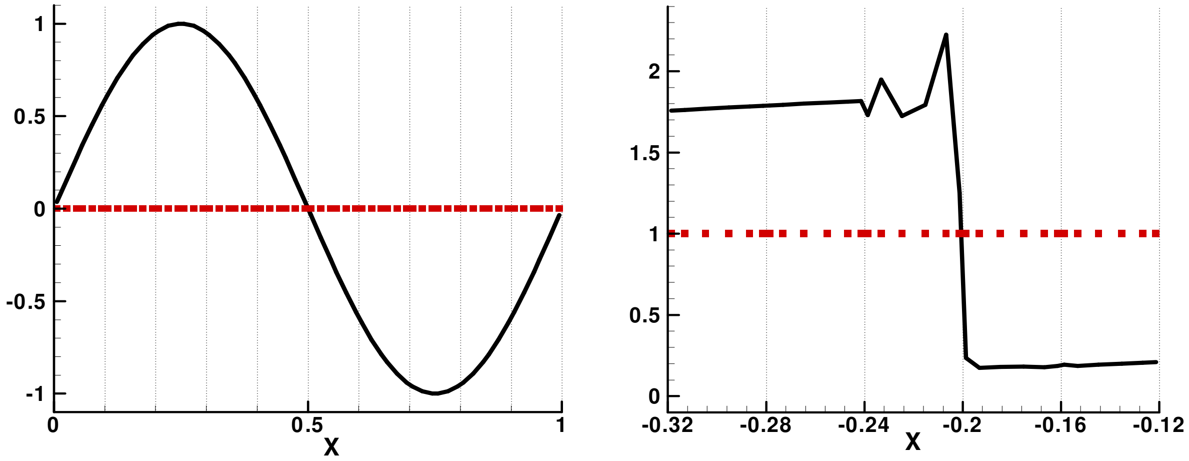


Figure 3.3 Minmod TVB marker by using *M* from [5] (p = 2).
(a) sine wave, 20 cells; (b) discontinuous profile [18], p=2, 200 cells.



(a) (b)

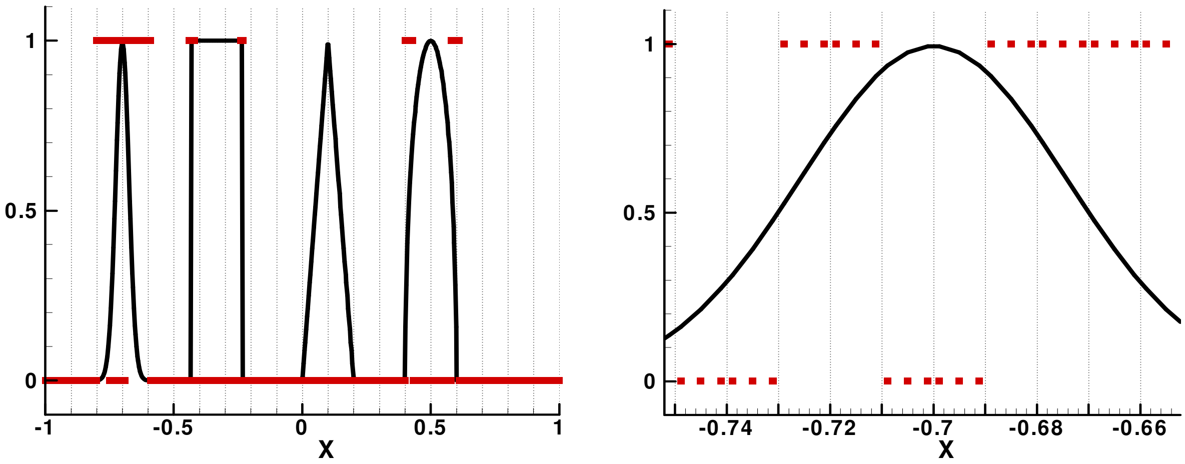
Figure 3.4 Minmod TVB marker for the oscillating shock profile with different M. (5 cells, $p = 6$)



(a) (b)

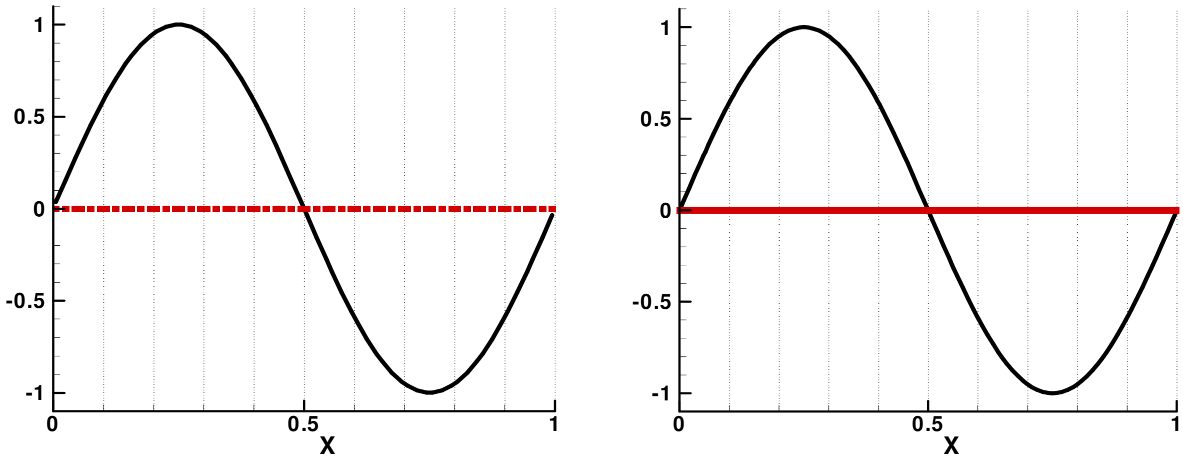
Figure 3.5 KXRCF marker

(a) sine wave, 20 cells, $p = 2$; (b) an oscillating shock profile (5 cells, $p = 6$).



(a) (b)

Figure 3.6 KXRCF marker for a discontinuous profile [18], 200 cells, $p = 2$. (b) is the close-up view for the Gaussian peak (the first from left) in (a).



(a) (b)

Figure 3.7 Modified Harten marker for a sine wave with 20 cells. (a) $p = 2$; (b) $p = 5$.

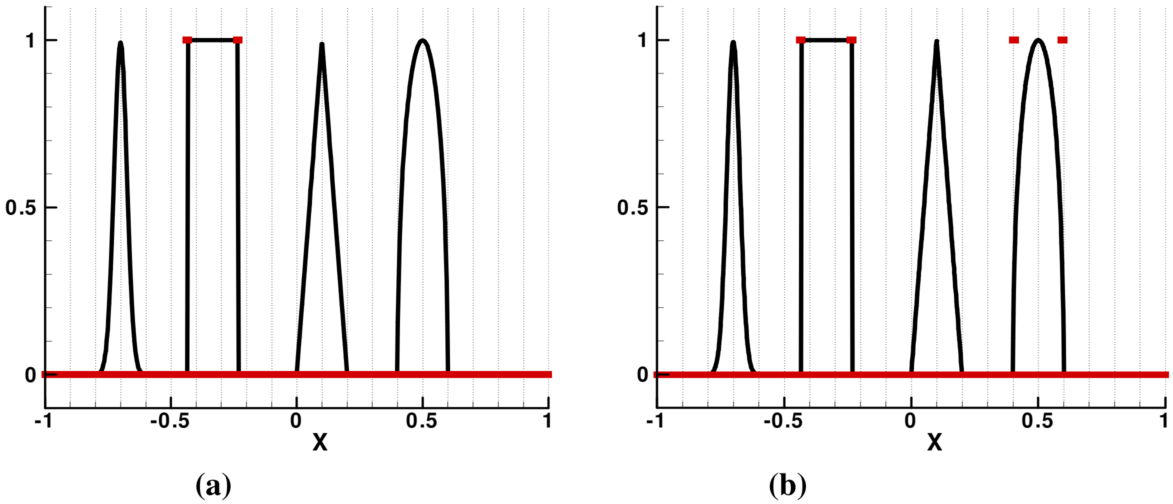


Figure 3.8 Modified marker for the discontinuous profile with 200 cells. (a) $p = 2$; (b) $p = 5$.

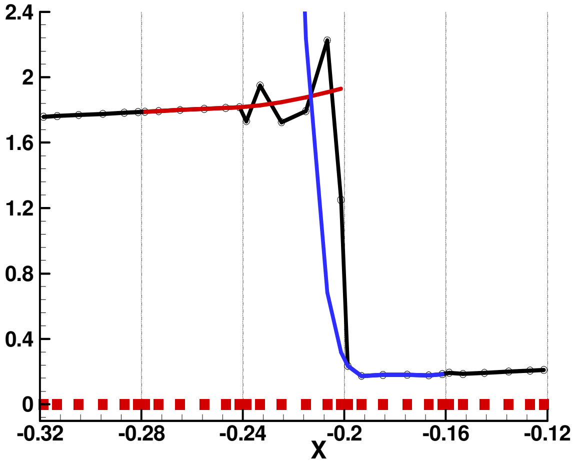


Figure 3.9 Harten condition (22) (5 cells, p=6) cells. Circle: solution points; Blue line: extension from right; Red line: extension from left.

3.4 Accuracy-Preserving TVD Marker

The above analyses bring some hints for us to set up some criteria to design a marker and a limiter. For production-level high order CFD codes, users-specified problem-dependent parameters are not desired. Although it is impossible to design a perfect marker, one design goal we hope to achieve is a marker free of user-specified parameters. Another design criterion is a marker that performs consistently regardless of mesh size and accuracy order of the scheme. The final criterion is a marker which is easy and efficient to implement, and can be applied to arbitrary unstructured grids.

The present Accuracy-Preserving TVD (AP-TVD) marker satisfies the above three criteria.

The key idea inside the new marker is based on the simple observation that if an extremum is smooth, then the first derivative of the solution should be locally monotonic. In order to recover the smooth extrema that are wrongly marked as troubled cells, the minmod function with favorable TVD property is activated on the cell-averaged first-derivatives to check if they are locally monotonic. In the minmod TVB marker, if parameter M is 0, it becomes a TVD marker. A well known drawback of the TVD marker is that cells at smooth solution extrema are marked. So the Minmod TVB marker needs a user-specified parameter M to “manually” unmark those extrema. In the present marker, we don’t need any free-parameter. Instead, we utilize the minmod function again, yet on the cell-averaged first-derivatives to unmark those extrema that wrongly marked as troubled cells by the minmod function acting on cell-averaged values.

The present marker follows the procedures as,

1) Compute the cell averaged solutions at each cell. Then compute the min and max cell averages for cell i from a local stencil using,

$$\overline{u}_{max,i} = \max\left(\overline{u}_{i-1}, \overline{u}_i, \overline{u}_{i+1}\right) \text{ and } \overline{u}_{max,i} = \max\left(\overline{u}_{i-1}, \overline{u}_i, \overline{u}_{i+1}\right). \qquad (3.25)$$

If

$$u_{j,i} > 1.001 \cdot \overline{u}_{max,i} \text{ or } u_{j,i} < 0.999 \cdot \overline{u}_{min,i}, \; (j = 1, p+2), \qquad (3.26)$$

cell i is considered as a possible troubled cell, which is further examined in the next step. The coefficients 1.001 and 0.999 in (3.26) are not problem-dependent free parameters. They are used to overcome machine error when comparing two real numbers so as to avoid the trivial case that the solution is constant in the neighborhood.

2) This step is aimed to unmark those cells at local extrema that are excessively marked as troubled cells in the first step (3.26). If an extremum is smooth, the first derivative of the solution should be locally monotonic. Therefore, a minmod TVD marker is applied to see if the second derivative is bounded by the slopes computed with the cell-averaged first-derivatives. Compute

$$\tilde{u}_i^{(2)} = minmod(\overline{u}_i^{(2)}, \beta\frac{\overline{u}_i^{(1)} - \overline{u}_{i-1}^{(1)}}{x_i - x_{i-1}}, \beta\frac{\overline{u}_{i+1}^{(1)} - \overline{u}_i^{(1)}}{x_{i+1} - x_i}) \qquad (3.27)$$

If $\tilde{u}_i^{(2)} = \overline{u}_i^{(2)}$, the cell is unmarked as a troubled cell. Otherwise, the cell is confirmed as a troubled cell. Obviously, this marker works for $p > 1$.

In order to compare the performance of the present new marker, the AP-TVD marker, with the Minmod TVB marker, the KXRCF marker, and the Modified Harten marker, we use the same three testing cases as before. Figure 3.10 shows that the present AP-TVD marker performs consistently well at the local extrema regions for all the polynomial order $p > 1$.

No cell is marked as a troubled cell in this smooth case as expected. Figure 3.11 shows that the present marker indeed detects the discontinuities without excessively marking other cells in smooth regions. It also shows the consistently good performance of detecting discontinuity for all the polynomial order $p > 1$. Figure 3.12 shows the present marker only marks the two cells at the discontinuity as expected, no elsewhere, in contrast to the other markers.

Comparing with the three "preferred" markers from [89], the present p-exact TVD marker has shown the advantages that 1) it has no free-parameter thus is problem-independent; 2) it is efficient in terms of the number of marked cells over the total number of cells and it performs well in marking the discontinuities; 3) it is compact and easy to implement for arbitrary unstructured meshes.

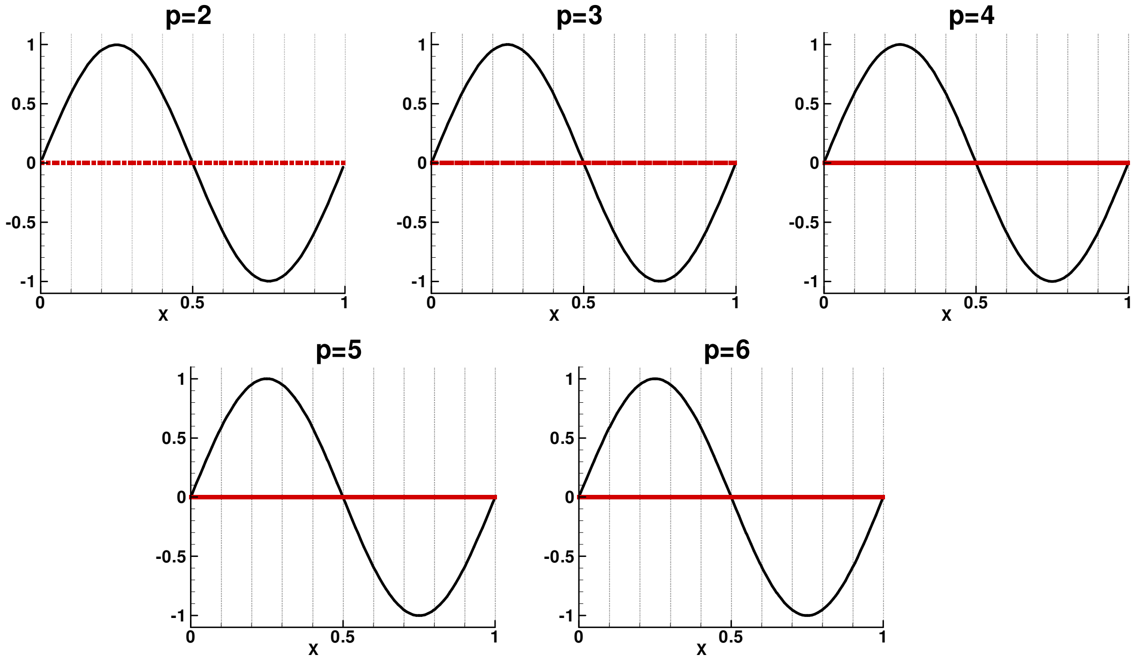


Figure 3.10 AP-TVD marker for the sine wave, 20 cells.

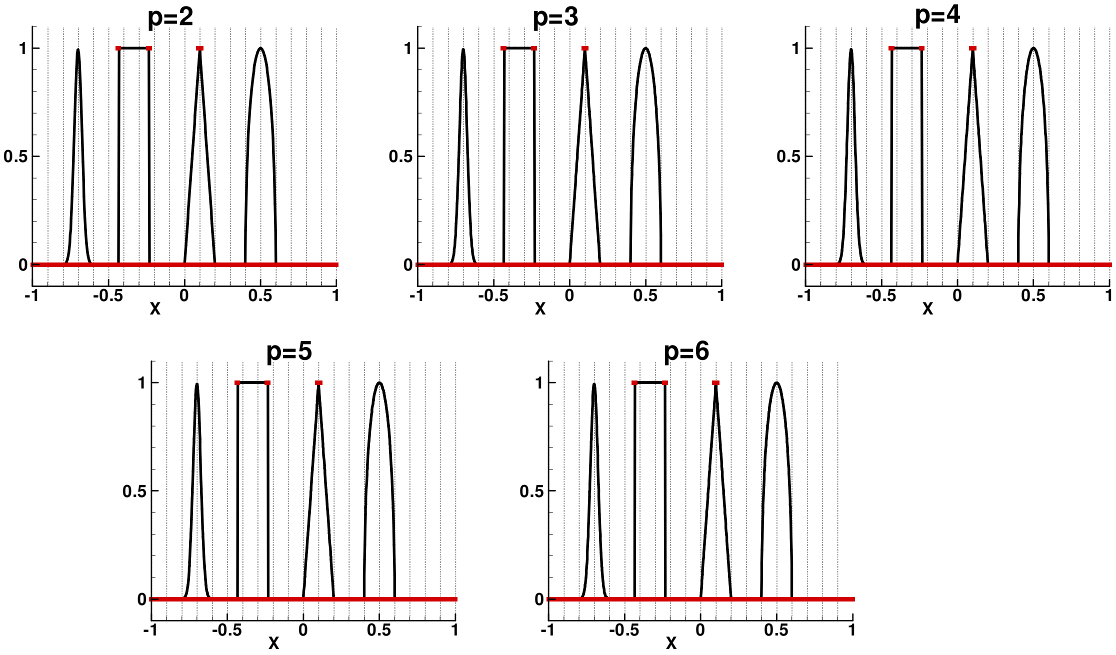


Figure 3.11 AP-TVD marker for the discontinuous profile [18], 200 cells.

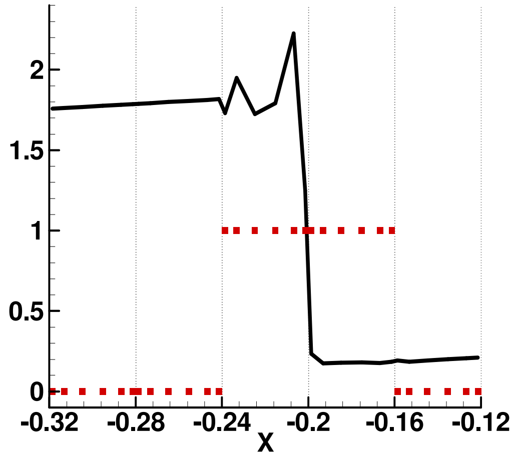


Figure 3.12 AP-TVD marker for the oscillating shock profile. (5 cells, $p = 6$).

3.5 Formulation of the Generalized Moment Limiter

Next we present a p-exact high-order accuracy-preserving limiter based on the moment limiter [14,72] and cell averages. The present limiter uses a Taylor-series-like expansion for the reconstruction, which is similar to that in [72]. The difference is that the expansion is performed with respect to the cell-averaged derivatives, rather than the derivatives at a specific point such as the cell centroid. Then these cell-averaged derivatives are limited in a hierarchical manner starting from the highest derivative. Combined with the AP-TVD marker, this new limiting technique exhibits the following properties: 1) free of problem-dependent parameters; 2) unstructured-grid based, easy to implement for 3D arbitrary meshes, and compact for parallel computing; 3) capable of suppressing spurious oscillations near solution discontinuities without loss of accuracy at the local extrema in the smooth regions. We will call this limiting technique “parameter-free generalized moment limiter” (or termed as “PFGM limiter”).

In the SD method the solution points are used to construct a degree p polynomial that can recover the conservative variables at the flux points. This reconstruction can produce spurious oscillations near a shock wave. Therefore a new non-oscillatory reconstruction is needed in the troubled cells. The following idea is followed. First the original degree p solution polynomial within a “troubled cell” is replaced with an equivalent polynomial based on the cell-averaged derivatives up to degree p. Then the high-order derivatives are hierarchically limited using the cell-averaged derivatives of one degree lower. In case that the highest derivative is not altered, the original polynomial is preserved. This procedure can be easily implemented for unstructured-grid based high-order methods.

Let’s consider the 1D case first. Let the original solution polynomial before limiting be $u_x(x)$, and the limited polynomial be $Y_i(x)$ within cell i. First we express $u_i(x)$ in terms of the cell-averaged derivatives up to degree p,

$$\begin{aligned} u_i(x) = \overline{u}_i + \overline{u}_i^{(1)}(x - x_i) \\ +\tfrac{1}{2}\overline{u}_i^{(2)}\left[(x - x_i)^2 - \tfrac{1}{12}h_i^2\right] \\ +\tfrac{1}{6}\overline{u}_i^{(3)}\left[(x - x_i)^3 - \tfrac{1}{4}h_i^2(x - x_i)\right] \\ +\tfrac{1}{24}\overline{u}_i^{(4)}\left[(x - x_i)^4 - \tfrac{1}{2}h_i^2(x - x_i)^2 + \tfrac{7}{240}h_i^4\right] \\ +\cdots \end{aligned} \tag{3.28}$$

where x_i represents the cell centroid coordinate. (3.28) is functionally equivalent to (3.2).

Next the cell-averaged derivatives are limited in a hierarchical manner by using a minmod-type limiter. Starting from the highest-order derivative, $\overline{u}^{(p)}$ is limited from,

$$\overline{Y}_i^{(p)} = minmod\left(\overline{u}_i^{(p)}, \beta\frac{\overline{u}_i^{(p-1)} - \overline{u}_{i-1}^{(p-1)}}{x_i - x_{i-1}}, \beta\frac{\overline{u}_{i+1}^{(p-1)} - \overline{u}_i^{(p-1)}}{x_{i+1} - x_i}\right) \tag{3.29}$$

If $\overline{Y}_i^{(p)} = \overline{u}_i^{(p)}$, then the highest derivative is not altered. No further limiting is required, and solution remains the same. Otherwise, the limiting process proceeds to the next lower derivative in a similar fashion,

$$\overline{Y}_i^{(k)} = minmod\left(\overline{u}_i^{(k)}, \beta\frac{\overline{u}_i^{(k-1)} - \overline{u}_{i-1}^{(k-1)}}{x_i - x_{i-1}}, \beta\frac{\overline{u}_{i+1}^{(k-1)} - \overline{u}_i^{(k-1)}}{x_{i+1} - x_i}\right), k = p - 1. \tag{3.30}$$

If $\overline{Y}_i^{(p-1)} = \overline{u}_i^{(p-1)}$, none of the lower derivative are further limited, i.e.,

$$\overline{Y}_i^{(k)} = \overline{u}_i^{(k)}, (k = p - 2, \ldots, 1). \tag{3.31}$$

Otherwise, the process continues in a similar fashion hierarchically until the first derivative is limited. In order to preserve the mean, the zero-th derivative (the mean) is retained, i.e., $\overline{Y}_i = \overline{u}_i$.

Finally the limited polynomial is written as,

$$\begin{aligned} Y_i(x) = \overline{Y}_i + \overline{Y}_i^{(1)}(x - x_i) & \\ + \tfrac{1}{2}\overline{Y}_i^{(2)}\left[(x - x_i)^2 - \tfrac{1}{12}h_i^2\right] & \\ + \tfrac{1}{6}\overline{Y}_i^{(3)}\left[(x - x_i)^3 - \tfrac{1}{4}h_i^2(x - x_i)\right] & \\ + \tfrac{1}{24}\overline{Y}_i^{(4)}\left[(x - x_i)^4 - \tfrac{1}{2}h_i^2(x - x_i)^2 + \tfrac{7}{240}h_i^4\right] & \\ + \cdots & \end{aligned} \tag{3.32}$$

Note that this limiter is compact, only involving data from its immediate neighbors, and easy to implement.

Next we present an efficient extension to multi-dimensional unstructured grids. Similar to the 1D case, we first express the solution polynomial with respect to the cell-averaged derivatives,

$$\begin{aligned} u_i(x,y) = \overline{u}_i + \overline{u}_{x,i}\Delta x + \overline{u}_{y,i}\Delta y & \\ + \tfrac{1}{2}\overline{u}_{xx,i}[\Delta x^2 - I_{xx}] + \tfrac{1}{2}\overline{u}_{yy,i}\left[\Delta y^2 - I_{yy}\right] & \\ + \overline{u}_{xy,i}\left[\Delta x\Delta y - I_{xy}\right], & \end{aligned} \tag{3.33}$$

where

$$\Delta x = x - x_i, \Delta y = y - y_i, \Delta x^2 = (x - x_i)^2, \Delta y^2 = (y - y_i)^2,$$

$$x_i \equiv \frac{1}{V_i}\int_{V_i} x dV, y_i \equiv \frac{1}{V_i}\int_{V_i} y dV,$$

$$I_{xx} \equiv \frac{1}{V_i}\int_{V_i} \Delta x^2 dV\,, I_{yy} \equiv \frac{1}{V_i}\int_{V_i} \Delta y^2 dV\,, I_{xy} \equiv \frac{1}{V_i}\int_{V_i} \Delta x \Delta y dV.$$

We proceed to limit the cell-averaged derivatives involved in (3.33) for the troubled cells. In multiple dimensions, especially in 3D, the efficiency of the limiter is a very important criterion. In order to achieve the highest efficiency, we decide to limit the derivatives of the same degree altogether with a scalar factor between 0 and 1, i.e., the limited polynomial can be written as

$$\begin{aligned} Y_i(x,y) &= \overline{u}_i + \alpha_i^{(1)}\left(\overline{u}_{x,i}\Delta x + \overline{u}_{y,i}\Delta y\right) \\ &+\alpha_i^{(2)}\left\{\frac{1}{2}\overline{u}_{xx,i}[\Delta x^2 - I_{xx}] + \frac{1}{2}\overline{u}_{yy,i}\left[\Delta y^2 - I_{yy}\right] + \overline{u}_{xy,i}\left[\Delta x \Delta y - I_{xy}\right]\right\}, \end{aligned} \tag{3.34}$$

where $\alpha_i^{(1)}$ and $\alpha_i^{(2)}$ are the scalar limiters in [0, 1] for the first and second derivatives on cell i. The essential 1D idea is then generalized into 2D and 3D. The limiter is conducted in the following steps assuming $p = 2$:

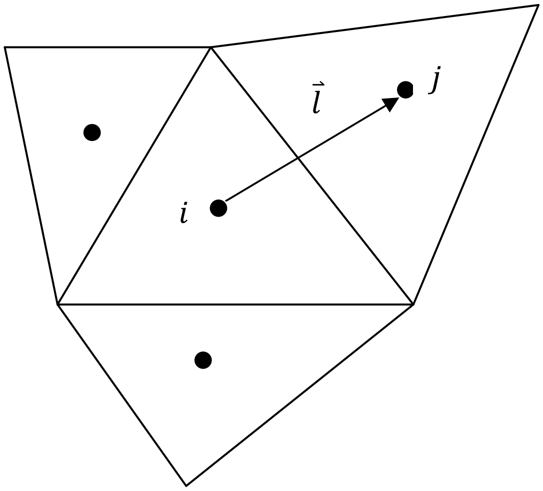


Figure 3.13 Sketch of multi-dimensional limiting

1. Compute the cell-averaged 2nd order derivatives in the troubled cell, and the cell-averaged 1st order derivatives in the troubled cell and its immediate face neighbors, as shown in Figure 3.13.

   Note: here for the case $p = 2$, the 2nd order derivatives are constants, and the 1st order derivatives are linear, so the cell-averaged 1st order derivatives are actually the first order derivatives at the cell centroids, that is,

   $$\overline{u}_{x,i} = u_{x,i}(x_i, y_i), \;\; \overline{u}_{y,i} = u_{y,i}(x_i, y_i).$$

   Similarly for general high order case of p degree, the p-th order derivatives are constants and the (p-1)th order derivatives are linear, therefore

   $$\overline{u}_{x,i}^{(p-1)} = u_{x,i}^{(p-1)}(x_i, y_i), \;\; \overline{u}_{y,i}^{(p-1)} = u_{y,i}^{(p-1)}(x_i, y_i).$$

   The equivalence between a cell-averaged derivative of order p-1 and the correspond--ing centroid value provides us a tool to do limiting on the highest order derivatives as in Step 3.

2. Assume one of the face neighbors is cell j. Define the unit vector connecting the centroids of cell i and cell j as $\vec{l}$. Compute the 2nd order derivative in $\vec{l}$ direction according to

   $$u_{ll,i} = \overline{u}_{xx,i} l_x^2 + \overline{u}_{yy,i} l_y^2 + 2\overline{u}_{xy,i} l_x l_y. \tag{3.35}$$

   $u_{ll,i}$ is to be examined next to determine whether limiting is necessary.

3. Compute the first derivative in $\vec{l}$ direction for both cell i and j,

   $$u_{l,i}(x_i, y_i) = \overline{u}_{x,i} l_x + \overline{u}_{y,i} l_y, \;\; u_{l,j}(x_j, y_j) = \overline{u}_{x,j} l_x + \overline{u}_{y,j} l_y. \tag{3.36}$$

   Now we can estimate a “non-oscillatory” 2nd-derivative using

$$\tilde{u}_{ll,i} \equiv \frac{u_{l,j}(x_j,y_j)-u_{l,i}(x_i,y_i)}{|\vec{r}_j-\vec{r}_i|}. \tag{3.37}$$

4. Finally the scalar limiter for this face is computed according to

$$\alpha_{ij}^{(2)} = minmod(1,\frac{\beta\tilde{u}_{ll,i}}{u_{ll,i}}). \tag{3.38}$$

The steps are repeated for the other faces. Finally, the scalar limiter for the cell i is the minimum of those computed for the faces, i.e.

$$\alpha_i^{(2)} = \min_{\mathrm{j}}(\alpha_{ij}^{(2)}). \tag{3.39}$$

If $\alpha_i^{(2)} = 1$, the 2nd order derivatives are not altered, and the solution polynomial remains the same. Otherwise, continue to limit the 1st order derivatives in a similar fashion by finding $\alpha_i^{(1)}$, i.e.

$$\alpha_{ij}^{(1)} = minmod(1,\frac{\beta\tilde{u}_{l,i}}{u_{l,i}}). \tag{3.40}$$

where

$$\tilde{u}_{l,i} \equiv \frac{\overline{u}_j - \overline{u}_i}{|\vec{r}_j - \vec{r}_i|} \tag{3.41}$$

Then

$$\alpha_i^{(1)} = \min_{\mathrm{j}}(\alpha_{ij}^{(1)}) \tag{3.42}$$

As can be seen, this generalized moment limiter keeps its compactness for arbitrary unstructured meshes because only cell-averaged values or derivatives on a local cell and its immediate neighboring cells are involved within the limiting process. Also the present limiter preserves a locally degree p polynomial, therefore satisfying the p-exact property.

3.6 Numerical Tests

In this section we provide extensive numerical experimental results to demonstrate the performance of the PFGM limiter described as above. In the numerical tests, for unsteady problems, the three-stage explicit TVD Runge-Kutta scheme [76] was used for time integration; for the steady transonic flows the implicit BLU-SGS [23] method was used.

3.6.1 Accuracy study for linear scalar wave equation

Consider the 1D linear wave equation,

$$u_t + u_x = 0, \qquad (3.43)$$

with initial condition $u(x,0) = \sin(2\pi x)$ and periodic boundary conditions. The CFL number ($\mathrm{CFL} = f'(u)\Delta t/\Delta x$) used for each case is as follows: 1) $\mathrm{CFL} = 0.01$ for $p = 1,2,3$; 2) $\mathrm{CFL} = 0.001$ for $p = 4,5$. These CFL numbers are small enough so that the error is dominated by the spatial discretization. In this test, the AP-TVD marker is turned off and all the cells are marked so that the present generalized moment limiter is applied to every cell in order to test the accuracy of the present limiter alone on smooth solution field. If the AP-TVD marker is turned on, then none of the cells is marked for this smooth solution as expected and the results is the same as the original unlimited schemes. The L_1 and L_∞ error norms at $t = 1$ for various schemes with and without the limiter are shown in Table 3.1 with L_1 error norms plotted out in Figure 3.14. We can see that the present limiter preserves the designed order of accuracy of the original SD method, although the magnitude of the error is larger than the unlimited schemes.

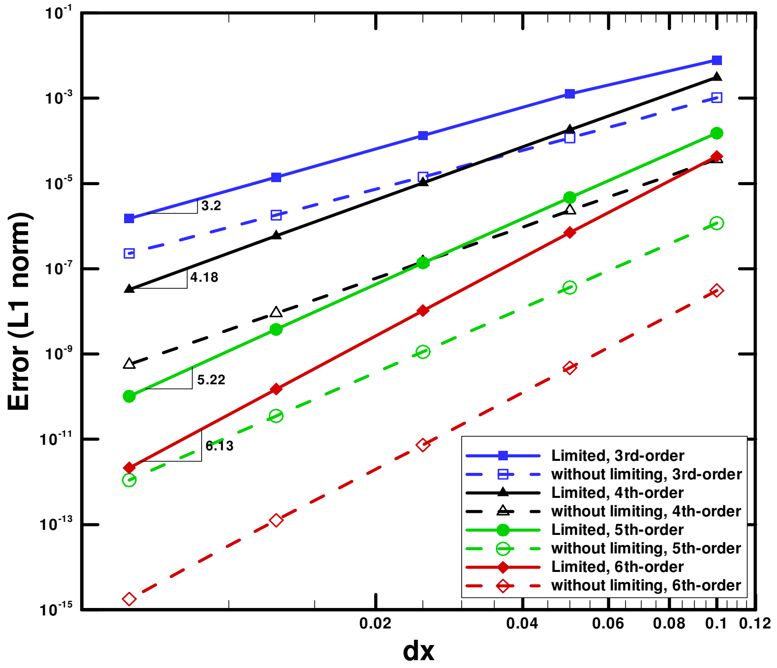


Figure 3.14 Accuracy study with linear advection equation (3.43) at *t*=1.

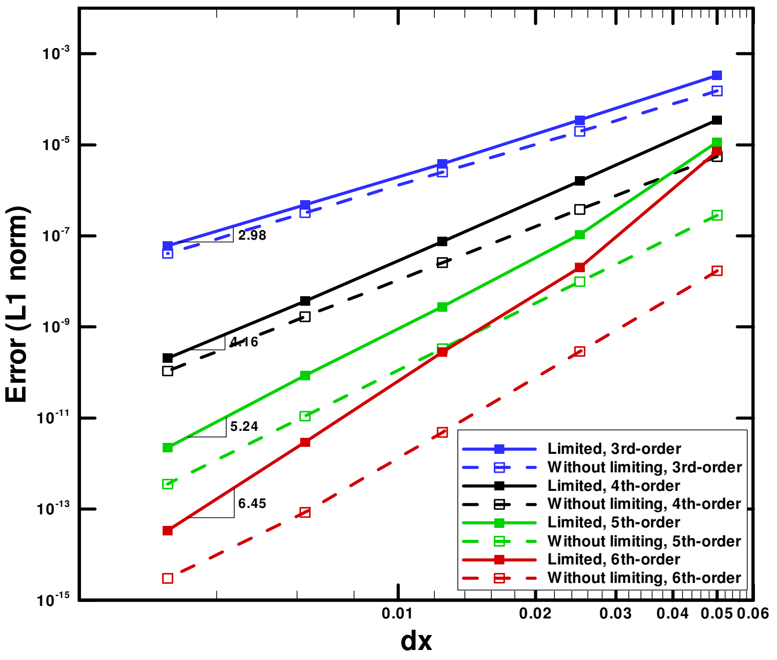


Figure 3.15 Accuracy study with non-linear Burgers equation (3.44) at *t*=0.1.

Table 3.1 1D linear scalar equation (3.43) at *t*=1

		SD with the present limiter				SD with no limiter			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
p=1	10	1.1576E-01		2.6194E-01		6.2796E-02		1.0096E-01	
	20	4.4521E-02	1.38	1.0353E-01	1.34	1.6648E-02	1.92	2.6315E-02	1.94
	40	1.2363E-02	1.85	4.3987E-02	1.23	4.2294E-03	1.98	6.6904E-03	1.98
	80	3.0796E-03	2.01	1.6468E-02	1.42	1.0626E-03	1.99	1.6763E-03	2.00
	160	7.5324E-04	2.03	6.0384E-03	1.45	2.6617E-04	2.00	4.1896E-04	2.00
p=2	10	7.8126E-03		1.9328E-02		1.0324E-03		2.5496E-03	
	20	1.2731E-03	2.62	4.6999E-03	2.04	1.1699E-04	3.14	2.6973E-04	3.24
	40	1.3321E-04	3.26	8.5111E-04	2.47	1.4260E-05	3.04	3.4615E-05	2.96
	80	1.3979E-05	3.25	1.4978E-04	2.51	1.8119E-06	2.98	4.5531E-06	2.93
	160	1.5226E-06	3.20	2.6891E-05	2.48	2.2916E-07	2.98	5.8315E-07	2.96
p=3	10	3.0792E-03		6.9129E-03		3.7241E-05		9.1629E-05	
	20	1.8082E-04	4.09	7.4760E-04	3.21	2.3173E-06	4.01	5.7505E-06	3.99
	40	1.0360E-05	4.13	7.6408E-05	3.29	1.4469E-07	4.00	3.6057E-07	4.00
	80	5.8874E-07	4.14	7.6032E-06	3.33	9.0404E-09	4.00	2.2606E-08	4.00
	160	3.2429E-08	4.18	7.4394E-07	3.35	5.6498E-10	4.00	1.4209E-09	3.99
p=4	10	1.5234E-04		4.1695E-04		1.1771E-06		3.0429E-06	
	20	4.6880E-06	5.02	2.1391E-05	4.28	3.6393E-08	5.02	9.6516E-08	4.98
	40	1.3730E-07	5.09	1.0898E-06	4.29	1.1283E-09	5.01	3.0140E-09	5.00
	80	3.7880E-09	5.18	5.5677E-08	4.29	3.5235E-11	5.00	9.4237E-11	5.00
	160	1.0190E-10	5.22	2.7379E-09	4.35	1.1011E-12	5.00	2.9455E-12	5.00
p=5	10	4.3204E-05		1.3471E-04		3.0958E-08		8.7921E-08	
	20	7.0644E-07	5.93	3.7905E-06	5.15	4.7381E-10	6.03	1.3735E-09	6.00
	40	1.0409E-08	6.08	1.0376E-07	5.19	7.3468E-12	6.01	2.2221E-11	5.95
	80	1.4958E-10	6.12	2.7882E-09	5.22	1.2642E-13	5.86	4.5155E-13	5.62
	160	2.1312E-12	6.13	7.3848E-11	5.24	1.7917E-15	6.14	5.0502E-15	6.48

3.6.2 Accuracy study for non-linear Burgers equation

The Burgers equation resembles the Euler or NS equation due to its nonlinear convection term. Consider the 1D Burgers equation without diffusion term,

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{3.44}$$

with initial condition $u(x,0) = 1 + \sin(\pi x)$, periodic boundary conditions. The CFL number used for each case is as follows: 1) $\mathrm{CFL} = 0.01$ for $p = 1{,}2{,}3$; 2) $\mathrm{CFL} = 0.001$ for $p = 4{,}5$. Again here the present limiter is applied to all the cells in order to test the performance of the limiter alone on smooth solution field. The L_1 and L_∞ error norms at $t = 0.1$ (when the solution is still smooth) for various schemes with and without the limiter are given in Table 3.2 with L_1 error norms plotted out in Figure 3.15. The results show that the present limiter preserves the designed order of accuracy of the original SD method. The results are quite similar to the linear scalar wave case.

Table 3.2 1D Burgers equation (3.44) at *t*=0.1

		SD with the present limiter				SD with no limiter			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
p=1	20	7.0163E-03		3.2793E-02		3.4361E-03		1.1337E-02	
	40	1.7393E-03	2.01	1.5394E-02	1.09	9.8232E-04	1.81	3.1532E-03	1.85
	80	4.3983E-04	1.98	4.9412E-03	1.64	2.6785E-04	1.87	8.3466E-04	1.92
	160	1.0758E-04	2.03	1.6378E-03	1.59	7.0983E-05	1.92	2.1344E-04	1.97
	320	2.6454E-05	2.02	7.1175E-04	1.20	1.8452E-05	1.94	5.3898E-05	1.99
p=2	20	3.3415E-04		3.3977E-03		1.5235E-04		8.6051E-04	
	40	3.4772E-05	3.26	5.4903E-04	2.63	1.9766E-05	2.95	1.3631E-04	2.66
	80	3.7991E-06	3.19	9.0709E-05	2.60	2.5154E-06	2.97	1.9223E-05	2.83
	160	4.7668E-07	2.99	1.5226E-05	2.57	3.2030E-07	2.97	2.5484E-06	2.92
	320	6.0395E-08	2.98	2.7296E-06	2.48	4.1043E-08	2.96	3.2824E-07	2.96
p=3	20	3.5171E-05		3.8092E-04		5.5214E-06		4.2708E-05	
	40	1.5999E-06	4.46	2.8953E-05	3.72	3.8188E-07	3.85	3.1396E-06	3.77
	80	7.5129E-08	4.41	1.9343E-06	3.90	2.5783E-08	3.89	2.1012E-07	3.90
	160	3.7027E-09	4.34	1.2389E-07	3.96	1.6896E-09	3.93	1.3534E-08	3.96
	320	2.0730E-10	4.16	7.8455E-09	3.98	1.0801E-10	3.97	8.6140E-10	3.97
p=4	20	1.1463E-05		1.7368E-04		2.8332E-07		2.8161E-06	
	40	1.0677E-07	6.75	2.6742E-06	6.02	9.8744E-09	4.84	1.2271E-07	4.52
	80	2.7858E-09	5.26	1.2660E-07	4.40	3.3521E-10	4.88	4.4423E-09	4.79
	160	8.5677E-11	5.02	6.0825E-09	4.38	1.1054E-11	4.92	1.4857E-10	4.90
	320	2.2690E-12	5.24	3.0752E-10	4.31	3.5661E-13	4.95	4.7974E-12	4.95
p=5	20	7.1775E-06		6.9139E-05		1.7147E-08		1.4552E-07	
	40	2.0250E-08	8.47	3.1842E-07	7.76	2.9003E-10	5.89	3.6121E-09	5.33
	80	2.8244E-10	6.16	6.9941E-09	5.51	4.8737E-12	5.90	6.1939E-11	5.87
	160	2.9685E-12	6.57	1.5381E-10	5.51	8.5272E-14	5.84	1.0062E-12	5.94
	320	3.3919E-14	6.45	2.9638E-12	5.70	3.0339E-15	4.81	2.9418E-14	5.10

3.6.3 Combined smooth and discontinuous waves

For all the rest tests, the present Parameter-Free AP-TVD Marker and the present Generalized Moment Limiter (so called "PFGM Limiter") will work together to capture discontinuities. Here we first solve the 1D wave equation (3.43) at $t = 8$ with the initial condition [63] set up to be the exact solution as plotted in Figure 3.16. Periodic boundary conditions were used. A uniform mesh is used with total of 200 cells. The CFL number used for each case is as follows: 1) $\mathrm{CFL} = 0.01$ for $p = 1,2,3$; 2) $\mathrm{CFL} = 0.001$ for $p = 4,5$. The long time evolution ($t = 8$) was considered in order to demonstrate high-order accuracy and low dissipation of the present schemes. The numerical solution is plotted at each solution point (red square). As seen that the present PFGM limiter yields good results at both the smooth region (as for the local extrema of the first jump) and discontinuities.

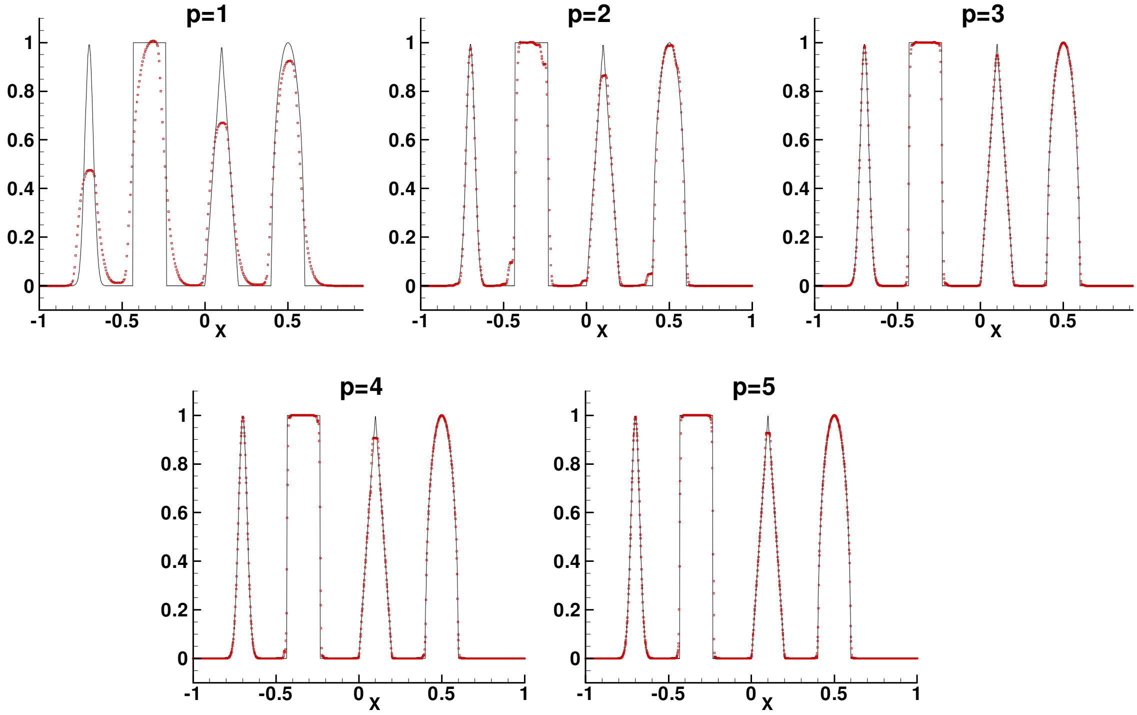


Figure 3.16 Solution of linear advection problem at *t*=8, *N*=200, *p*=1,2,3,4,5. Solid line: exact solution; red square: solution points.

3.6.4 Burgers equation with shock

In this example the Burgers equation (3.44) was solved with the same initial conditions and periodic boundary conditions as in 3.6.2, but until $t = 0.8$ when a shock appears. The CFL number used for each case is as follows: 1) $\mathrm{CFL} = 0.01$ for $p = 1,2,3$; 2) $\mathrm{CFL} = 0.001$ for $p = 4,5$. A mesh with 100 uniform cells was used with the present PFGM limiter of various orders. The shock was captured sharply without oscillations, as shown in Figure 3.17.

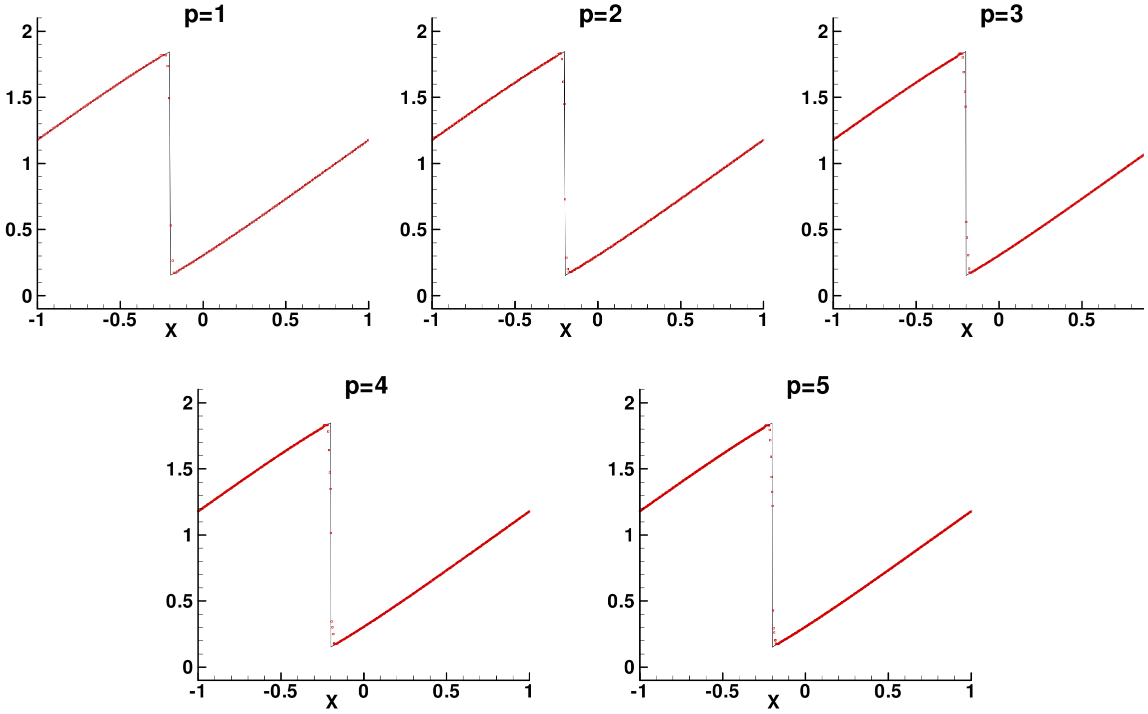


Figure 3.17 Solution of Burgers equation (3.44) at t=0.8, *N*=100, *p*=1,2,3,4,5. Solid line: exact solution; red square: solution points.

3.6.5 Sod shock-tube problem

Sod shock-tube problem was solved to test the present PFGM limiter for the Euler equations,

$$u_t + f(u)_x = 0 \tag{3.45}$$

where

$$u = (\rho, \rho v, E)^T, f(u) = \left(\rho v, \rho v^2 + p, v(E+p)\right)^T,$$

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2, \gamma = 1.4,$$

and ρ, v, E, p are the density, velocity, total energy, and pressure, respectively. The initial condition is

$$(\rho, p, v) = \begin{cases} \quad (1,1,0) & for\ x < 0, \\ (0.125, 0.1, 0) & \quad for\ x \geq 0. \end{cases}$$

In Figure 3.18, the computed density at t=2 with the present PFGM limiter is compared with the exact solution for p=1,2,3,4. The time step size used for each case is as follows: 1) $dt = 0.001$ for $p = 1,2$; 2) $dt = 0.0005$ for $p = 3,4$. Note that the solutions appear oscillation-free, and both the shock and contact were well captured. The higher-order scheme appears to yield better results.

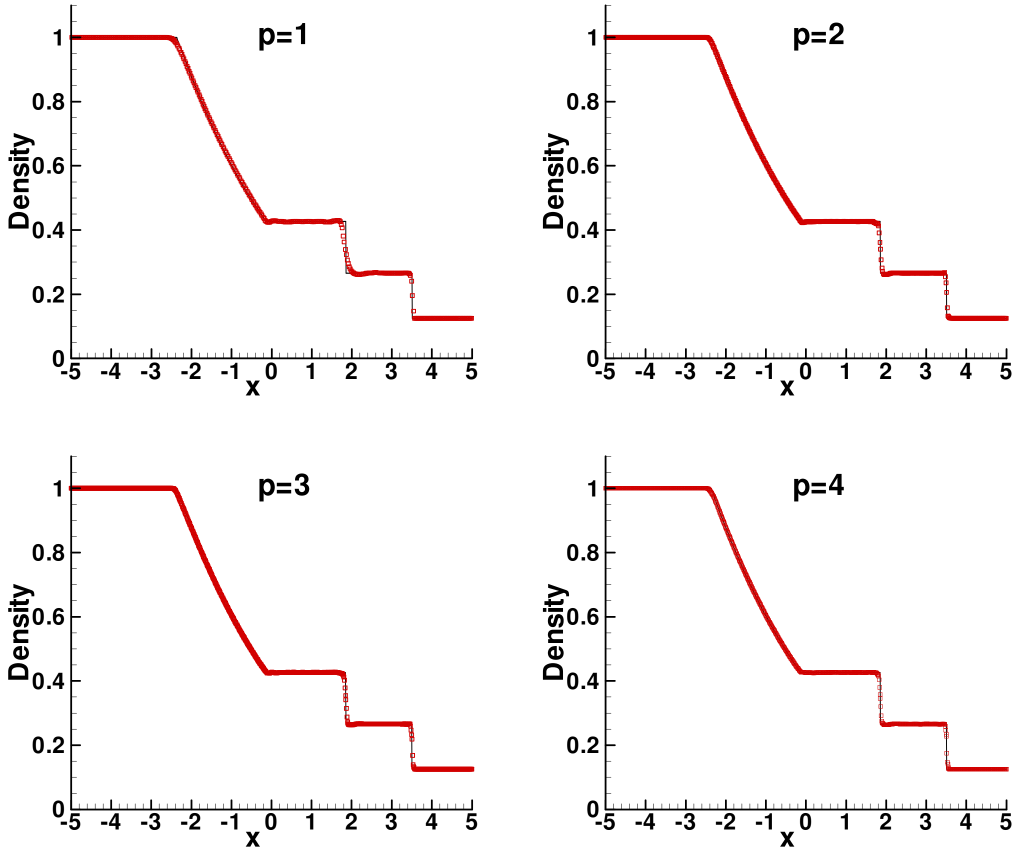


Figure 3.18 Sod problem, *t*=2, *N*=200 cells, *p*=1,2,3,4.
Solid line: exact solution; Red square: solution points.

3.6.6 Shock acoustic-wave interaction

The problem of shock-acoustic wave interaction [58] was solved to show the advantage of the present high-order limiter for the problems with both shock waves and complex smooth features. We solved the Euler equations (3.45) with a moving Mach=3 shock interacting with a sine wave in density, i.e., initially,

$$(\rho, p, v) = \begin{cases} (3.857143, 10.333333, 2.629369) & for\ x \le -4, \\ (1 + 0.2\sin(5x), 1, 0) & for\ x \ge -4. \end{cases} \tag{3.46}$$

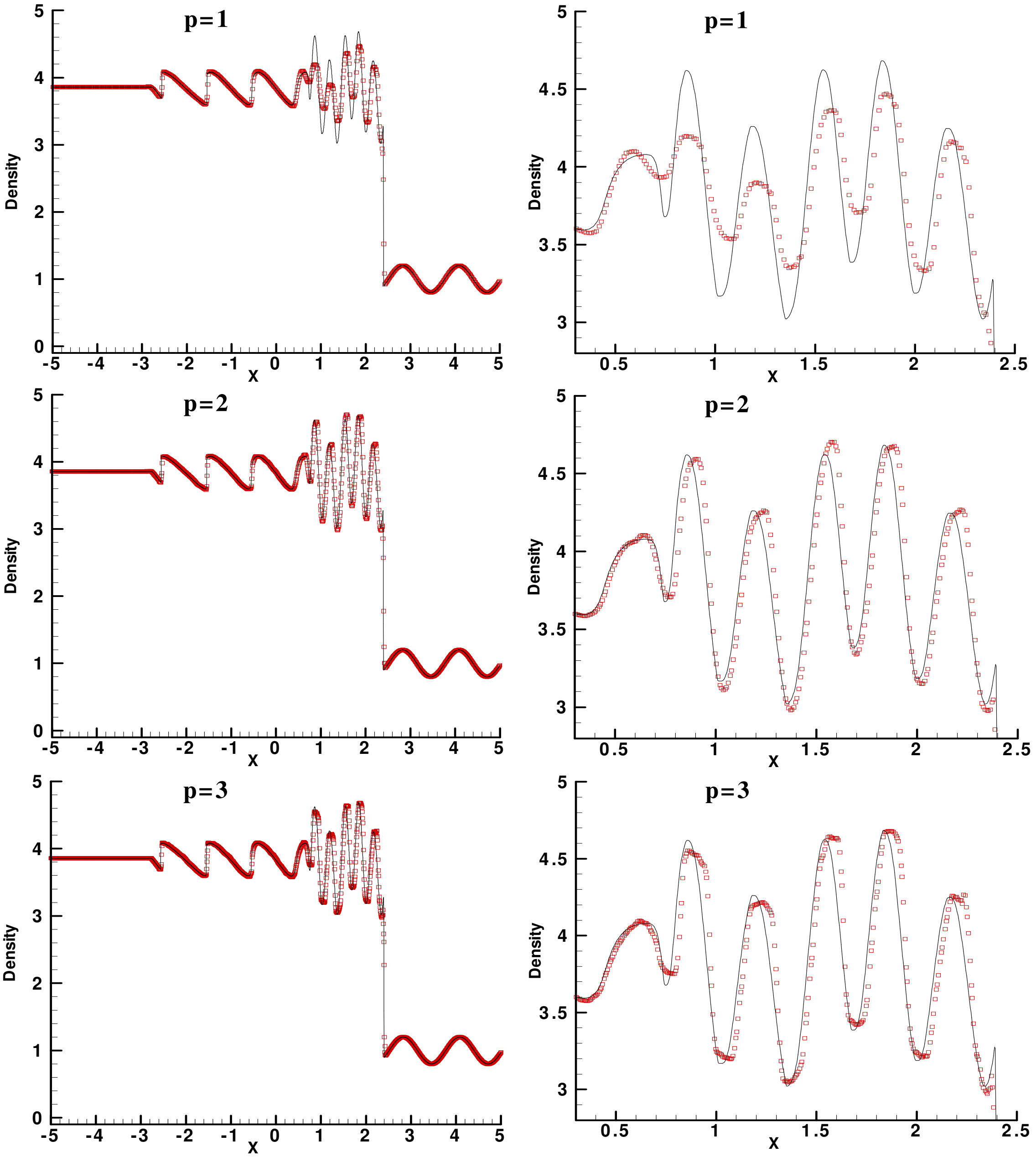


Figure 3.19 The shock-acoustic interaction problem, *t*=1.8, *N*=400 cells, *p*=1,2,3. Solid line: exact solution; red square: solution points. Right: close-up view for the complex smooth region in the left graphs.

For comparison, a converged solution using a second-order MUSCL scheme on a grid of 3,200 cells is used as the ‘‘exact’’ solution. In Figure 3.19, the converged solution of density at *t*=1.8 is compared with the “exact” solution for *p*=1, 2, 3 with the present PFGM limiter on a medium mesh with N=400 cells and time step size dt=0.0005. It shows that the smooth local extrema are better recovered if using the present limiter in higher-order form (p=2, 3). A close-up view of the complex smooth region is also given aside for each case in Figure 3.19.

3.6.7 Shock vortex interaction

From now on we test the present limiter for 2D inviscid flow problems with discontinuities. The conservative form of the 2D Euler equation can be written as

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0, \tag{3.47}$$

where Q is the conservative solution variables, F, G are the inviscid flux given below,

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

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

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

with ratio of specific heat $\gamma=1.4$.

The shock vortex problem describes the interaction between a stationary shock wave and a vortex, and is a good test for the PFGM limiter in resolving both discontinuities and important smooth features. The flow conditions are the same as in [58]. The computational domain is taken to be $[0,2]\times[0,1]$. A stationary shock with a pre-shock Mach number of $M_s=1.1$ is positioned at $x=0.5$ and normal to the x-axis. Its left state is $(\rho, u, v, p)=$

$(1, \sqrt{\gamma}, 0,1)$. An isentropic vortex ($\frac{p}{\rho^{\gamma}} = const.$) is superposed to the flow left to the shock and centers at $(x_c, y_c) = (0.25,0.5)$. Therefore the flow variables on the left side of the shock are as follows

$$u = M_s\sqrt{\gamma} + \varepsilon\tau e^{\alpha(1-\tau^2)}sin\theta,$$

$$v = -\varepsilon\tau e^{\alpha(1-\tau^2)}cos\theta,$$

$$\rho = \left(1 - \frac{(\gamma-1)\varepsilon^2 e^{2\alpha(1-\tau^2)}}{4\alpha\gamma}\right)^{\frac{1}{\gamma-1}},$$

$$p = \left(1 - \frac{(\gamma-1)\varepsilon^2 e^{2\alpha(1-\tau^2)}}{4\alpha\gamma}\right)^{\frac{\gamma}{\gamma-1}},$$

where $\tau = r/r_c$ and $r = \sqrt{((x-x_c)^2 + (y-y_c)^2)}$. Here ε denotes the strength of the vortex, α is the decay rate of the vortex; and r_c is the critical radius for which the vortex has the maximum strength. They are set to be $\varepsilon = 0.3, \alpha = 0.204, r_c = 0.05$.

The 3rd order SD method was employed as the base scheme in the simulation on a coarse mesh of 86×35 cells in order to have almost the same numbers of degree of freedom as in [58] (where the WENO method was used) for comparison purposes. The time step size used is dt=0.0005. The grids are uniform in *y*-direction and clustered near the shock in *x*-direction. The boundary conditions for the top and bottom boundaries are set to symmetry, or slip wall. The computed solution fields (pressure contours) for different time moments are given in Figure 3.21 through Figure 3.25 to compare the 3rd-order PFGM limiter and the 2nd-order linear limiter (in which the solution at the troubled cells is assumed linear).

The snapshots at t=0.05, t=0.20, and t=0.35 are shown in Figure 3.21, Figure 3.22, and Figure 3.23, respectively. We can see that the 3rd-order PFGM limiter recovers the smooth vortex (an extremum) much better than the linear limiter, and the shock discontinuity has been more sharply captured as well. It appears the present simulation captures the shock waves with a higher resolution than [58, FIG.15]. This can be seen from a black/white graph as shown in Figure 20 (at t=0.2) to compare with that in [58, FIG.15] under the same conditions. Not like [58, FIG.15], the present result of pressure contour at the shock interface does not extend to the top and bottom boundary. This is actually expected, because pressure there is discontinuous, so there should be no contour. Therefore the color plots are needed to clearly show the regions before and after the shock.

Figure 24 and Figure 25 show snapshots for later moments, t=0.6 and t=0.8 using the 3rd-order PFGM limiter and the linear limiter, respectively. We can see here that the reflective boundary takes effects as time goes long enough when one of the shock bifurcations reaches the top boundary and reflects. Figure 25(a) shows that the reflection is well captured and in the meanwhile the smooth vortex (an extremum) is preserved as well, which gives better resolution than the linear limiter as shown in Figure 25(b). Again the 3rd-order PFGM limiter gives better results than the linear limiter in terms of less numerical noise and better-resolved vortex.

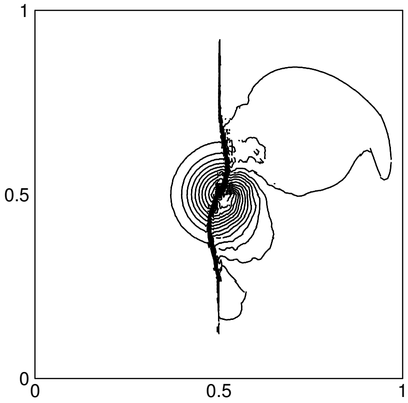
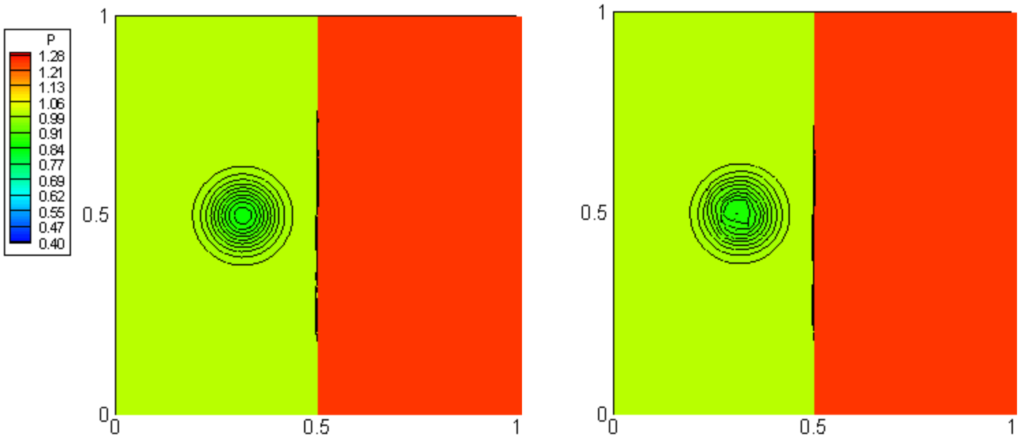
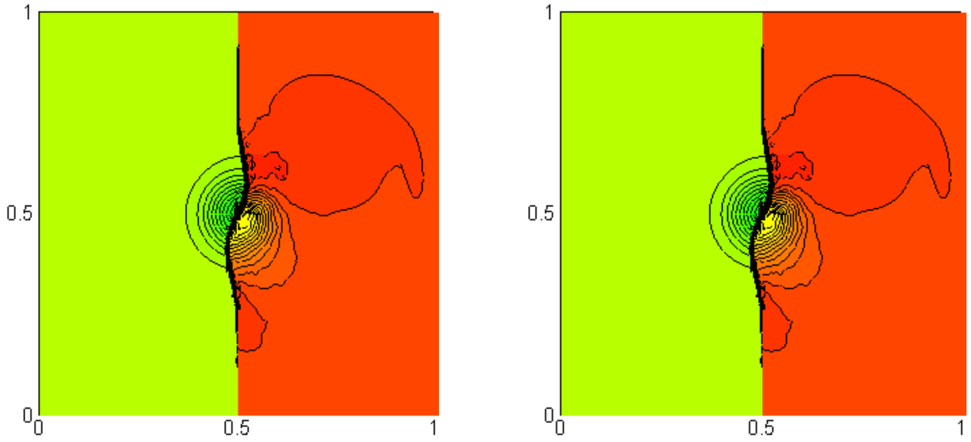


Figure 3.20 Pressure contour for 2D shock-vortex interaction, *t*=0.2.



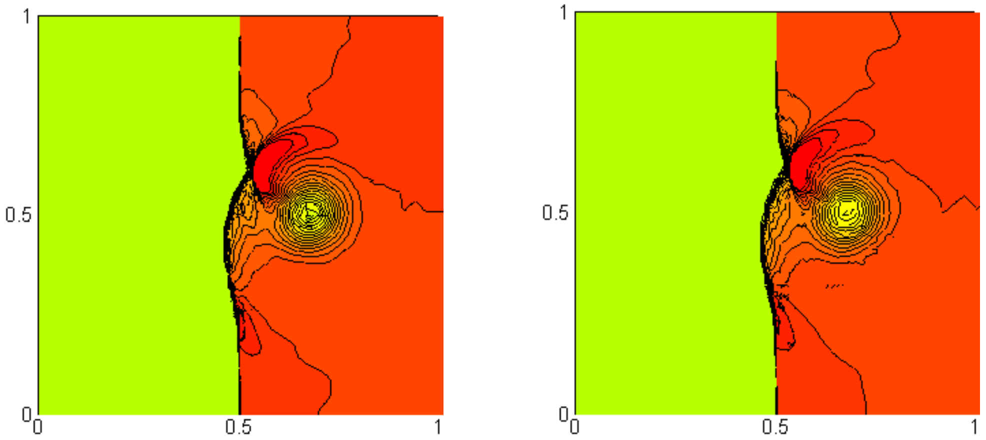
(a) **(b)**

Figure 3.21 Pressure contours for the shock-vortex interaction at *t*=0.05. (Base scheme: 3rd-order SD. 61 contours from 0.4~1.29.) (a) 3rd-order PFGM limiter; (b) Linear limiter.



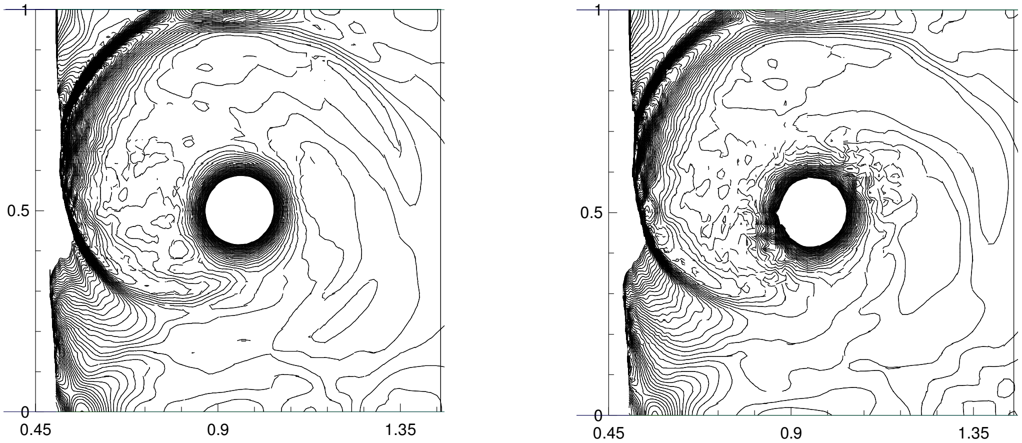
(a) **(b)**

Figure 3.22 Pressure contours for the shock-vortex interaction at *t*=0.2. (Base scheme: 3rd-order SD. 61 contours from 0.4~1.29.) (a) 3rd-order PFGM limiter; (b) Linear limiter.



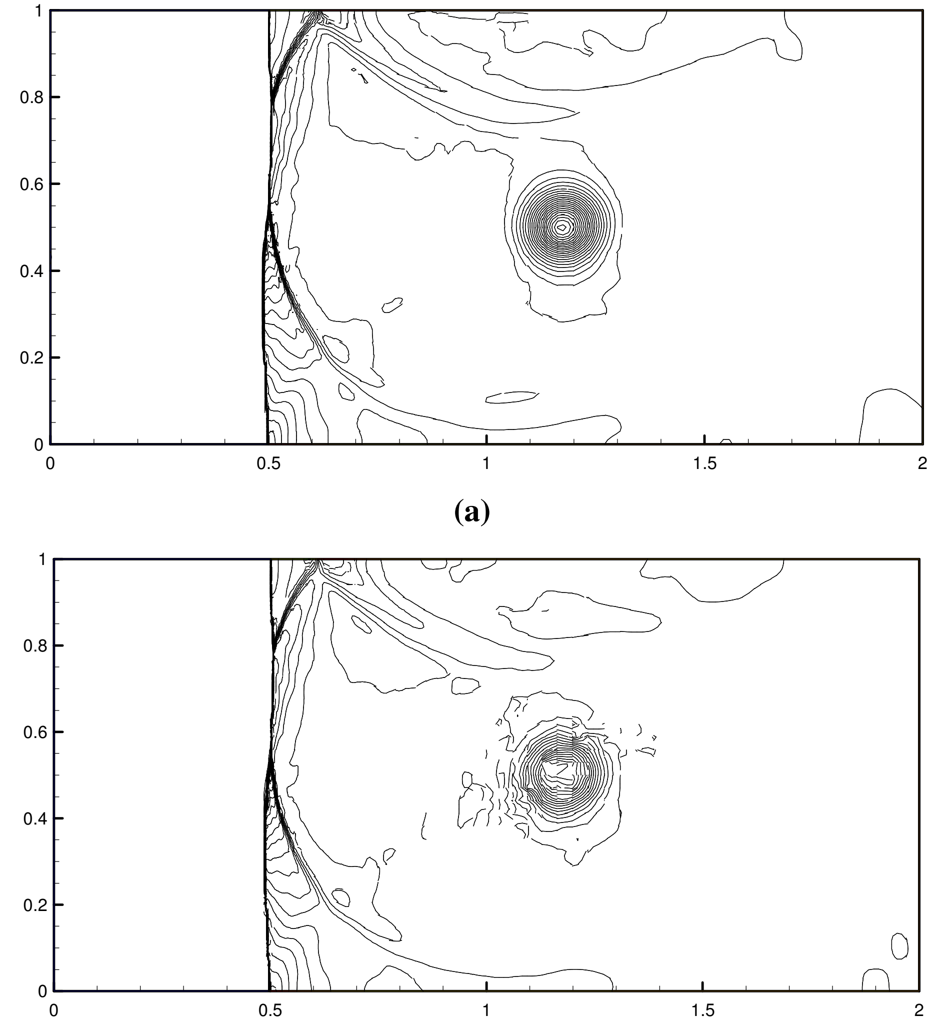
(a) (b)

Figure 3.23 Pressure contours for the shock-vortex interaction at *t*=0.35. (Base scheme: 3^{rd}-order SD. 61 contours from 0.4~1.29.) (a) 3^{rd}-order PFGM limiter; (b) Linear limiter.



(a) (b)

Figure 3.24 Pressure contours for the 2D shock-vortex interaction at *t*=0.6. (Base scheme: 3^{rd}-order SD. 90 contours from 1.19~1.37.) (a) 3^{rd}-order PFGM limiter; (b) Linear limiter.



(a)

(b)

Figure 3.25 Pressure contours for the 2D shock-vortex interaction at *t*=0.8. (Base scheme: 3rd-order SD. 90 contours from 1.19~1.37.) (a) 3rd-order PFGM limiter; (b) Linear limiter.

3.6.8 Oblique shock reflection by a wedge

This example considers a Mach 2 flow passing a wedge of 20°. Notice that in 3.6.7 the normal shock is aligned with the grid, while in this example we don't have this luxury. The state ahead of the shock is set to be $(\rho, u, v, p) = (1.4, 2, 0, 1)$. The boundary conditions are as follows: 1) supersonic inlet at the inlet on the left side; 2) inviscid wall boundary condition for the wall; 3) simple extrapolation boundary condition for the upper boundary and the outlet on the right end. A coarse mesh (400 elements, 20 boundary elements) was used for this case, as shown in Fig. 24b. The density contours in Fig. 24a shows that the present 3rd-order PFGM limiter captured the shock sharply (within one element). Only the cells at the shock are marked (in red), and the typical marked cells when the shock is formed are shown in Fig. 24b. As we can see, the AP-TVD marker works well as expected.

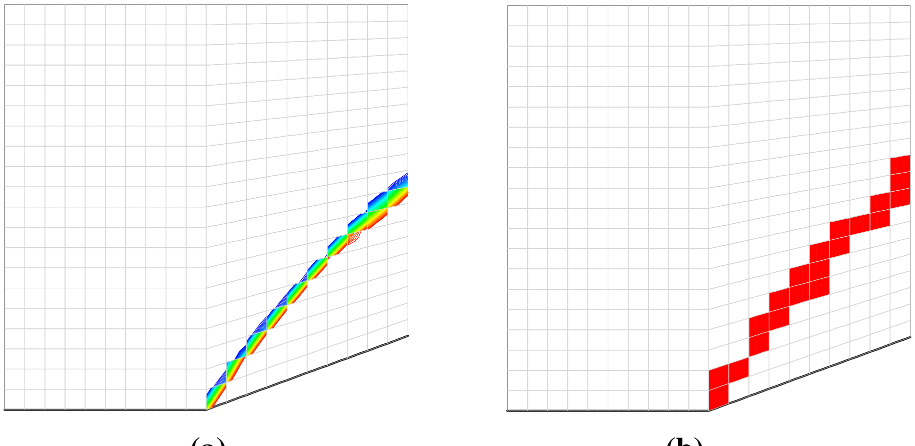
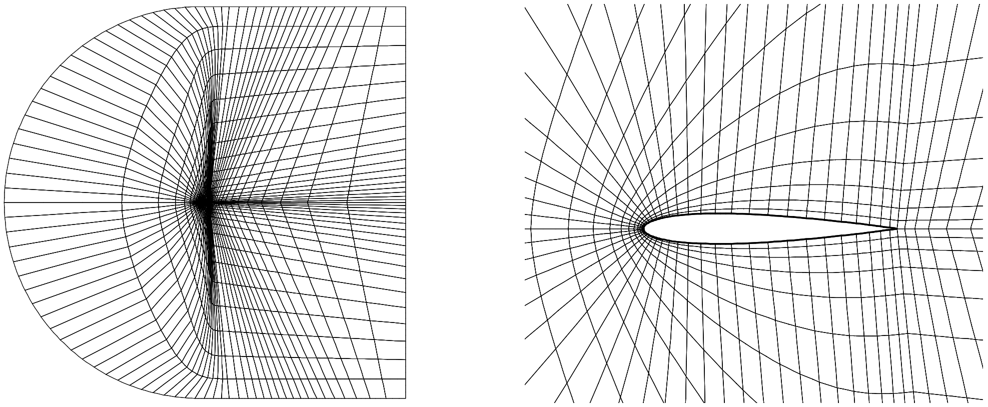


Figure 3.26 Mach 2 flow past a wedge of 20° by using the 3rd-order PFGM limiter with the SD method (400 elements, 20 boundary elements). (a) Density contour; (b) Marked cells.

3.6.9 Transonic flow over NACA0012 airfoil

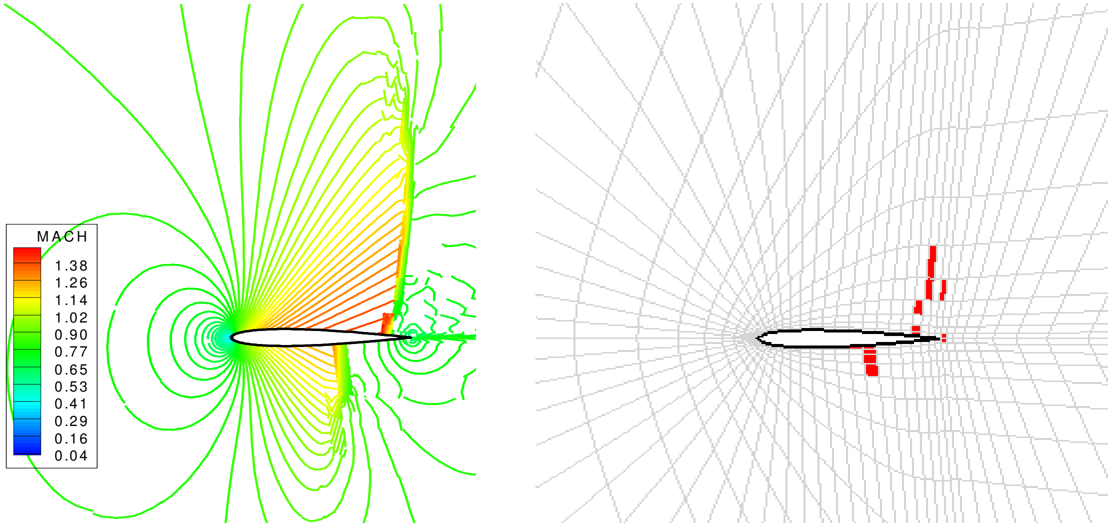
This example is the transonic flow over a NACA0012 airfoil at Mach 0.85 and an angle of attack $\alpha = 1°$, characterized by the existence of two shocks, one on the upper surface and one on the lower surface. To demonstrate the advantage of the present high-order limiter, we used a relatively coarse mesh (1584 hexahedral elements, 52 elements on the upper and lower wall surfaces) as shown in Figure 3.27. The implicit BLU-SGS scheme was employed in a compact form [23] for time integration in this case.

Figure 3.28(a) shows the Mach contours obtained with the 3rd-order PFGM limiter, and Figure 3.28(b) gives a snapshot of the typical distribution of the marked cells. It is shown that the present limiter is indeed able to eliminate the spurious oscillations and capture the shock discontinuities sharply while maintaining the high-order accuracy at smooth regions. It was noticed that the marked cells are located just in the vicinity of the upper and lower shock discontinuities, and the average number of the marked cells during the BLU-SGS implicit time iterations is a very small percentage (about 2%) of the total number of cells. Therefore it shows that the present AP-TVD marker works well and efficiently for multidimensional cases.



(a) (b)

Figure 3.27 The unstructured hexahedral meshes for the NACA0012 airfoil in transonic flow (1584 elements, 52 wall boundary elements). (a) the whole domain; (b) close-up view around the airfoil.



(a) (b)

Figure 3.28 The transonic flow over NACA0012 airfoil ($M_\infty = 0.85, \alpha = 1°$) by using the 3rd-order PFGM limiter in the SD method. (a) Mach contours; (b) the marked cells (red) at the 1000th implicit time step.

3.7 Conclusions

Three design criteria have been set for a general purpose limiter: 1) free of user-specified parameters; 2) capable of preserving a local degree p polynomial; 3) applicable to arbitrary unstructured meshes. The parameter-free generalized moment (PFGM) limiter developed in the present study appears to meet all of the criteria. The limiter is composed two components: an efficient accuracy preserving TVD marker for "troubled cells" based on cell-averaged state variables, and a hierarchical generalized moment limiter capable of handling arbitrary unstructured meshes. The PFGM limiter has been implemented and tested for a high-order SD method, although it can be easily applied to all other similar high-order methods. The AP-TVD marker is based on the cell-averaged solutions and solution derivatives, and is quite efficient to implement. It appears that smooth extrema are not marked, while the discontinuous cells are consistently marked, without the use of any user-specified parameter. The AP-TVD marker compares favorably against several markers in the literature, such as the TVB marker, KXRCF marker, or the Harten marker. Accuracy studies confirmed that the limiter is capable of preserving accuracy in smooth regions. Numerical tests for a wide variety of problems in 1D and 2D with both discontinuities and smooth features demonstrated the capability and usefulness of the PFGM limiter.

CHAPTER 4 EFFICIENT LINE IMPLICIT METHOD

To speed up solution convergence in high Reynolds number flows, in this chapter we re-examine the current state-of-art implicit method research and present an efficient line implicit solver with several new features including: a) compact scheme combining a line BLU-SGS solver for the lined-up cells within the thin boundary layer coupled with a cell BLU-SGS solver for other less stiff flow regions; b) low memory storage requirement for the implicit method due to the BLU-SGS partial line solver/partial cell solver scheme and an efficient low-storage strategy for LU decomposition of the cell Jacobians; c) robust and accurate viscous fluxes for anisotropic grids based on the second approach of Bassi and Rebay (BR2); d) generic and compact formulation to be programmed as a black box so as to be easily applied in general high-order methods.

4.1 Background and motivation

The high-order spatial operators bring benefit of higher-order accuracy. However they are much stiffer than low-order ones, and the stiffness increases with the polynomial degree order. So far the severe stiffness problem is the major drawback that hinders high-order methods to be applied as widely as low-order methods in industry. A simple illustration for this well-known problem is given in Figure 4.1, where due to the versatileness of polynomial interpolation small changes (from green dots to red dots) on the multiple DOF values in one cell can cause large shape change (from dash line to solid line) on the cell solution profile particularly on the cell interfaces.

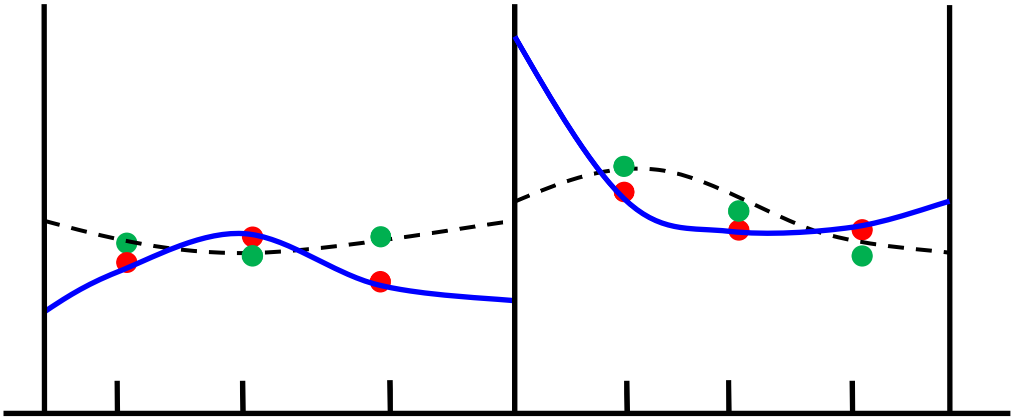


Figure 4.1 Illustration of stiffness due to high-order spatial operator

Therefore it is much more difficult to make a high-order simulation converged to steady state. The situation becomes even worse when this stiff spatial operator combines with the anisotropy induced stiffness, which is usually caused by highly clustered grids in the boundary layer for high Reynolds number viscous flow problems. In that case the explicit time integrator is too slow due to the CFL condition. The implicit methods allow much larger stable CFL number, yet with the drawback of much larger computer memory storage. Our research focus here is implicit method with low-storage requirement.

Despite our focus on implicit methods, it is worthy to note that some representative explicit methods have significantly improved stability limit compared with their predecessors. For example, the Fourth-Order Four-Stage Runge-Kutta scheme developed by Jameson, Schmidt and Turkel [57] has been used in many production CFD codes (2nd-order n space) due to its large stability limit. Another major explicit scheme is the Strong-Stability-Preserving (SSP) Runge-Kutta Schemes, which was originally developed by Shu [101], and Shu and Osher [102] (with its original name as TVD Runge-Kutta schemes) and was further studied by many researchers [44,105]. The SSP-RK scheme is more stable due to its TVD or

SSP property. However, for high Reynolds number viscous flows, those above-mentioned explicit methods can be one or more order slower than the implicit methods in terms of CPU time, as shown in Section 4.6.

Many an early low-order implicit method ends up with a large nonlinear algebraic system with sparse block matrices to be solved iteratively for each temporal iteration step. An effective algorithm for such systems is essential even for low-order case. Currently the basic ideas of the implicit methods or multi-grids methods for high-order methods were extended from the low-order ones; none of them are specifically designed for high-order methods. Now that there are multiple DOFs in one high-order cell, the computer core memory occupied by the cell Jacobian matrices is much larger than that in low-order case. For example, the memory requirement for polynomial construction of degree higher than three might be prohibitive for a 3D engineering problem. Therefore the main challenge will be to develop effective and low storage implicit methods for high-order operators. A good implicit algorithm can also serve as a "smoother" for geometric or p-multigrid approaches to further speed up convergence.

Many types of implicit algorithms have been successfully developed for unstructured grid based solvers in the last two decades, for example, the element Jacobi, Gauss-Seidel, precondition GMRES [98,12], matrix free Krylov [93], lower-upper symmetry Gauss-Seidel (LU-SGS) [23,99], and line implicit algorithms [80]. The present line implicit method was developed by following the philosophy of LU-SGS because of its favorable feature of low-storage combined with fast convergence due to its unique procedure of inner forward and backward sweep iterations.

The original LU-SGS approach was developed by Yoon and Jameson [141] to solve compressible flow on structured grids, and demonstrated high solution efficiency with low storage requirements. Later, it was extended and applied to hybrid structured and unstructured grids [103]. Unstructured-grid-based LU-SGS schemes have demonstrated performance similar to that on structured grids [99]. In the original LU-SGS scheme, a special first-order approximation in numerical flux is employed to linearize the left-hand side and result in the reduction of the block diagonal matrices to diagonal matrices. As a result, LU-SGS does not require any extra memory compared to explicit methods and is free from any matrix inversion. All of the off-diagonal matrices still contribute to the implicit operator through one forward and one backward sweep of a Gauss–Seidel iteration, thus significantly improving efficiency over an explicit scheme. However the special first-order approximation used in the original LU-SGS to give diagonal matrices does degrade convergence rate, especially after several orders of convergence [23]. To further improve the convergence rate, Chen & Wang [23] and Jameson & Caughey [56] developed a block (preconditioned) non-linear LU-SGS (BLU-SGS) approach, which ends up with a block diagonal matrix. The non-linear BLU-SGS method shows much faster convergence than the original LU-SGS with only a small portion of memory usage increase. The non-linear BLU-SGS also shows faster convergence and much less memory required than the Fully Linearized Implicit method even for 2nd-order case, because i) BLU-SGS only store the diagonal block matrices with no need to store off-diagonal matrices; ii) BLU-SGS solves a non-linear system for each time step thus achieving faster convergence rate.

Besides the stiffness due to high-order spatial operator, another kind of stiffness is the anisotropy induced stiffness, which is usually caused by highly clustered grids in the boundary layer for high Reynolds number viscous flow problems. High aspect ratio grids are required in order to properly resolve boundary layer and wake regions in high Reynolds number viscous flows. Unfortunately these grids introduce numerical stiffness that severely reduces the convergence rate. Indeed, the higher the Reynolds, the more grid stretching is required, and the worse the convergence rate becomes. This poses particular difficulties for simulating flight Reynolds number flows for large aircraft, where the required meshes may contain stretching ratios in excess of 100,000 to 1. One might consider a fully implicit solver to handle the presence of highly stretched cells, but the computer time needed in a fully implicit solver to compute Jacobian matrices, which costs the major portion of total CPU time, grows rapidly with an increasing number of grid cells, particularly in 3D high-order cases. Moreover, the memory usage of a fully implicit solver is too high for practical 3D application even in low-order cases, let alone in high-order schemes.

In fact, the anisotropy induced stiffness arises because the traditional cell implicit methods (such as cell LU-SGS) have no strong solution coupling in the strong geometric coupling direction of anisotropic cell alignment. The line implicit solvers create the solution lines in the anisotropic regions based on the directions of strong coupling in terms of both convection and diffusion, and solve the flow on the lines in a coupled manner to overcome the anisotropy induced stiffness, thus obtain much faster convergence rate. On the other hand, the line solvers result in block tri-diagonal system which can be solved efficiently. Therefore both physical and numerical features make the line solvers favorable for high Reynolds number flows with high aspect ratio mesh. The idea of line implicit approach has been explored for structured grid based flow solvers, for example, approximate factorization (AF) and alternating direction implicit (ADI) algorithms are equivalent to applying a “line” implicit scheme in each of the coordinate directions. In the community of unstructured grids based high-order methods, the idea of line-implicit approach was developed by Mavriplis [81] and Fidkowski et al. [38] as a smoother for multi-grid solver with different strategies to create the solution lines within the unstructured grids.

The present line implicit method was developed based on the BLU-SGS method and line implicit approach with improved efficiency and robustness for high Reynolds number flows.

4.2 A compact and generic non-linear BLU-SGS formulation

The intrinsic compact feature of the residual operator in the non-linear BLU-SGS method allows us to program a compact and generic cell implicit solver to be applied as a black box by general high-order CFD methods. Consider the following hyperbolic conservation law

$$\frac{\partial Q}{\partial t} + \nabla \cdot F = 0, \tag{4.1}$$

where Q is the local DOFs, and F is flux (inviscid and/or viscous flux). Its semi-discretized equation for a compact high-order method on the current cell *c* at time level *n*+1 can be formulated in a general form as,

$$M_c \frac{\partial \tilde{Q}_c}{\partial t} = \tilde{R}_c\left(\tilde{Q}_c^{n+1}, \tilde{Q}_{nb}^{n+1}\right), \tag{4.2}$$

where $\tilde{Q}$ is the global DOFs in a cell, and M_c is a relatively small cell-based "mass matrix". $\tilde{R}_c$ is the global spatial residual of the current cell *c*. The subscript "*c*" denotes for current cell and "*nb*" for neighboring cells. M_c is an identity matrix in the FV, SV and SD methods, but not in the DG method. For the followings we set $M_c = 1$ for simplicity since only the

FV, SV, and SD methods are employed to test the present implicit solver. R_c is a function of the current cell and its neighboring cells, therefore (4.2) is coupled with those equations for other cells in the domain to form a large system of first-order differential equations.

The Backward Euler scheme for temporal discretization of (4.2) gives,

$$\frac{\Delta\tilde{Q}_c^{n+1}}{\Delta t} = \tilde{R}_c\left(\tilde{Q}_c^n, \tilde{Q}_{nb}^n\right) + \frac{\partial\tilde{R}_c}{\partial\tilde{Q}_c}\Delta\tilde{Q}_c^{n+1} + \sum_{nb}\frac{\partial\tilde{R}_c}{\partial\tilde{Q}_{nb}}\Delta\tilde{Q}_{nb}^{n+1}, \tag{4.3}$$

where $\Delta\tilde{Q}^{n+1} = \tilde{Q}^{n+1} - \tilde{Q}^n$. Here only first-order time difference is used as an example; the extension to 2nd-order or higher temporal discretization is straightforward. Notice that $\frac{\partial\tilde{R}_c}{\partial\tilde{Q}_c}$ is the diagonal block element and $\frac{\partial\tilde{R}_c}{\partial\tilde{Q}_{nb}}$ for off-diagonal block elements. The essence of LU-SGS is to keep diagonal terms implicitly and off-diagonal terms “formal explicitly” while doing multiple inner sweeping iteration steps back and forth within one time step to account the nonlinearity from the off-diagonal terms. Denote k as an inner sweeping step, then (4.3) is solved through,

$$\frac{\Delta\tilde{Q}_c^{k+1}}{\Delta t} = \tilde{R}_c\left(\tilde{Q}_c^n, \tilde{Q}_{nb}^n\right) + \frac{\partial\tilde{R}_c}{\partial\tilde{Q}_c}\Delta\tilde{Q}_c^{k+1} + \sum_{nb}\frac{\partial\tilde{R}_c}{\partial\tilde{Q}_{nb}}\Delta\tilde{Q}_{nb}^{k+1}. \tag{4.4}$$

where $\Delta\tilde{Q}^{k+1} = \tilde{Q}^{k+1} - \tilde{Q}^n$. Further manipulation of terms in (4.4) yields,

$$\left[\frac{I}{\Delta t} - \frac{\partial\tilde{R}_c}{\partial\tilde{Q}_c}\right]\mathbb{D}\tilde{Q}_c^{k+1} = \tilde{R}_c\left(\tilde{Q}_c^n, \tilde{Q}_{nb}^n\right) + \frac{\partial\tilde{R}_c}{\partial\tilde{Q}_c}\Delta\tilde{Q}_c^k + \sum_{nb}\frac{\partial\tilde{R}_c}{\partial\tilde{Q}_{nb}}\Delta\tilde{Q}_{nb}^{k+1} - \frac{I}{\Delta t}\Delta\tilde{Q}_c^k \tag{4.5}$$

which can be simplified as,

$$\left[\frac{I}{\Delta t} - \frac{\partial\tilde{R}_c}{\partial\tilde{Q}_c}\right]\mathbb{D}\tilde{Q}_c^{k+1} = \tilde{R}_c\left(\tilde{Q}_c^n, \tilde{Q}_{nb}^{k+1}\right) - \frac{\Delta\tilde{Q}_c^k}{\Delta t}, \tag{4.6}$$

where $\mathbb{D}\tilde{Q}_c^{k+1} \equiv \Delta\tilde{Q}_c^{k+1} - \Delta\tilde{Q}_c^k = \tilde{Q}_c^{k+1} - \tilde{Q}_c^k$. At k+1 inner sweep step, not all the neighbors $\tilde{Q}_{nb}^{k+1}$ have been updated in (4.6). Then the latest available values $\tilde{Q}_{nb}^*$ are used for $\tilde{Q}_{nb}^{k+1}$, and (4.6) becomes

$$\left[\frac{I}{\Delta t}-\frac{\partial \tilde{R}_c}{\partial \tilde{Q}_c}\right]\mathbb{D}\tilde{Q}_c^{k+1}=\tilde{R}_c\left(\tilde{Q}_c^n,\tilde{Q}_{nb}^*\right)-\frac{\Delta\tilde{Q}_c^k}{\Delta t}, \tag{4.7}$$

Note that here we actually solve the non-linear system (4.3) at each time step by using (4.7). The left-hand-side matrix (a diagonal preconditioner) is a cell-based local matrix with size of $(\#DOFs\ per\ cell)^2$, depending on spatial accuracy order, for example, 24×24 for a 3rd-order 2D problem. The right-hand-side is actually the latest unsteady residual for the current cell c, and can be treated as a black box. If the residual for each cell is found from the DOFs on the current cell and its neighbors, the small matrix in (4.7) is easily solved by using LU decomposition technique to update the DOFs. Therefore the solver for (4.7) can be programmed as a compact black box which is independent of numerical flux formulation or spatial discretization in general sense. This is significant because in high-order case the spatial reconstruction can be complicated. Either iteration error or maximum sweep iteration steps can be used to control inner iteration numbers. The efficient way to store the left-hand-side matrix is to only store its LU decomposition matrices instead of itself. Also as a variation, the costly left-hand-side matrix can be frozen for some time steps to further save computer time.

4.3 The BR2 viscous flux for the SD method

How to accurately and efficiently formulate the viscous fluxes is of importance not only for solution accuracy but also for convergence rate. Many efforts have been made to formulate viscous flux in context of high-order methods, and the major methods include the averaging scheme [60,61], the local DG scheme (LDG) by Cockburn and Shu[25], the second approach of Bassi and Rebay (BR2) by Bassi et al. [13] for the DG method, the interior penalty (IP) method by Douglas and Dupont [35] and Kannan and Wang [59], the

Recovery scheme for DG by van Leer et al. [123,124], and more recently flux reconstruction scheme by Hyun [54], etc.

From our tests in Section 4.5 it has been found that the BR2 scheme is more robust than the simple averaging gradient scheme for highly stretched grids. The BR2 scheme is compact with only immediate neighboring cells involved within. In the followings the BR2 scheme, which was originally developed for DG method, is re-formulated in context of the SD method.

We only consider 2D quadrilateral or 3D hexahedral meshes in the boundary layers, which are generally used in high Reynolds number flows due to the well-known fact that quadrilateral or hexahedral meshes are more robust and efficient than triangular or tetrahedral meshes in boundary layer region. Suppose two adjacent cells on the left and right of a common interface, respectively, as shown in Figure 4.2.

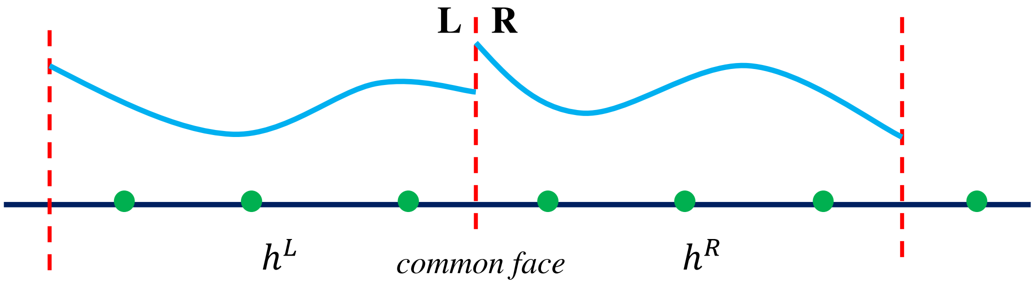


Figure 4.2 BR2 viscous flux scheme on quadrilateral or hexahedral mesh

The viscous part of the flux vector in (4.1), $F^{vis} = F^{vis}(Q, \nabla Q)$ is a function of both solution and solution gradient. In BR2 the common interface solution value is given by

$$Q_f^{com} = \frac{1}{2}[Q_f^L + Q_f^R] \tag{4.8}$$

and the gradient on the interface can be computed as

$$\nabla Q_f^{com} = \frac{1}{2}[\nabla Q^L + \nabla Q^R] + \frac{(p+1)^2}{2}[\frac{1}{h^L}\left(Q_f^{com} - Q^L\right) - \frac{1}{h^R}(Q^{com} - Q^R)]\vec{n} \tag{4.9}$$

where $p+1$ is the scheme accuracy order, h^L and h^R are the left and right cell size, respectively, and $\vec{n}$ is the interface normal.

4.4 Line implicit method

The present line solver was developed based on non-linear BLU-SGS method and preserves its nice features of low-storage, non-linear solution procedure, and generic compactness. However, in the boundary layer regions with highly-stretched grids the present line implicit method creates the solutions lines and solves the flow on the lines in a coupled manner to overcome the anisotropy induced stiffness, thus obtain much faster convergence rate than the pure cell BLU-SGS method.

Suppose highly stretched grids used in the boundary layer as shown in Figure 4.3, we connect several cells along the wall normal direction to form a solution line from each bottom wall cell. The cell numbers lined-up on each solution line can be different.

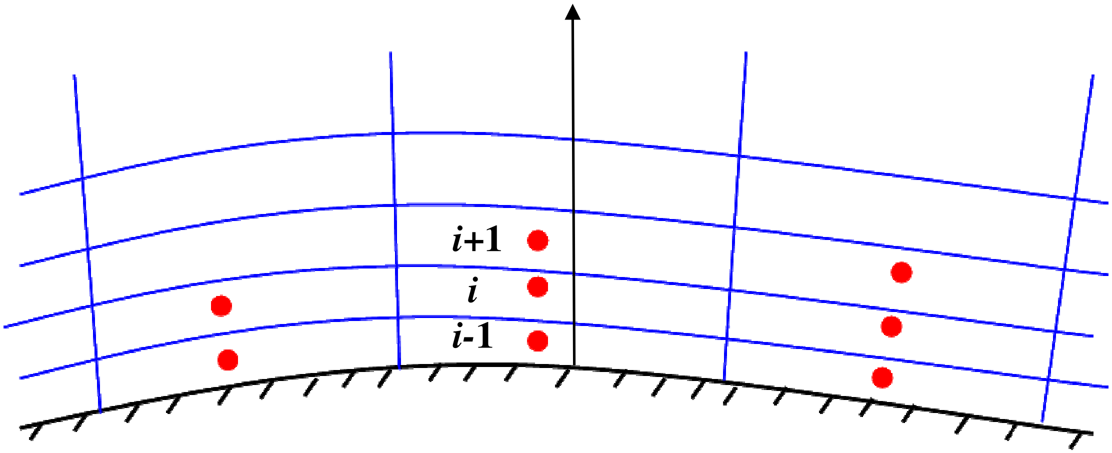


Figure 4.3 Lined-up cells in boundary layer

Cell *i*-1 and cell *i*+1 are also two of the neighboring cells for the current cell *i*. From (4.3) we have

$$\begin{aligned}\frac{\Delta\tilde{Q}_i^{n+1}}{\Delta t} &= \tilde{R}_i\left(\tilde{Q}_i^n,\tilde{Q}_{nb}^n\right) + \sum_{nb\neq i-1,i+1} \frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{nb}}\Delta\tilde{Q}_{nb}^{n+1} \\ &\quad + \frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{i-1}}\Delta\tilde{Q}_{i-1}^{n+1} + \frac{\partial\tilde{R}_i}{\partial\tilde{Q}_c}\Delta\tilde{Q}_i^{n+1} + \frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{i+1}}\Delta\tilde{Q}_{i+1}^{n+1}\end{aligned} \tag{4.10}$$

where $i = 2, \dots, ncl.$ ncl is the total number of cells on the line. Similar to the cell BLU-SGS procedure, applying inner sweeps to (4.10) yields

$$\begin{aligned}&\left[-\frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{i+1}}\right]\mathbb{D}\tilde{Q}_{i+1}^{k+1} + \left[\frac{I}{\Delta t} - \frac{\partial\tilde{R}_i}{\partial\tilde{Q}_i}\right]\mathbb{D}\tilde{Q}_i^{k+1} + \left[-\frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{i-1}}\right]\mathbb{D}\tilde{Q}_{i-1}^{k+1} \\ &\quad = \tilde{R}_i\left(\tilde{Q}_i^n,\tilde{Q}_{nb}^*\right) - \frac{\Delta\tilde{Q}_i^k}{\Delta t}, \qquad nb \neq i-1, i+1.\end{aligned} \tag{4.11}$$

The three left-hand-side matrices in (4.11) are computed numerically similarly as done in the cell matrix-free implicit solver. Assume a small quantity ε (i.e. $\varepsilon = 10^{-8}$), then a cell-based LHS Jacobian matrix in (4.11) can be computed by using the incremental residual due to a small incremental solution value with each DOF on the cell,

$$\frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{i+1}} \approx \frac{\tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{i+1}+\varepsilon,\tilde{Q}_{nb}\right) - \tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{i+1},\tilde{Q}_{nb}\right)}{\varepsilon}, nb \neq i, i+1, \tag{4.12}$$

$$\frac{\partial\tilde{R}_i}{\partial\tilde{Q}_i} \approx \frac{\tilde{R}_i\left(\tilde{Q}_i+\varepsilon,\tilde{Q}_{nb}\right) - \tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{nb}\right)}{\varepsilon}, nb \neq i, \tag{4.13}$$

$$\frac{\partial\tilde{R}_i}{\partial\tilde{Q}_{i-1}} \approx \frac{\tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{i-1}+\varepsilon,\tilde{Q}_{nb}\right) - \tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{i-1},\tilde{Q}_{nb}\right)}{\varepsilon}, nb \neq i, i-1. \tag{4.14}$$

Note $\tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{i+1},\tilde{Q}_{nb}\right) = \tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{nb}\right) = \tilde{R}_i\left(\tilde{Q}_i,\tilde{Q}_{i-1},\tilde{Q}_{nb}\right)$, all of which express the same residual for a DOF on the cell. Therefore a time-saving procedure is to compute the non-incremental residuals first for all the cells in the domain, then compute the incremental residuals for a cell, as shown in the CPU time tests in Section 4.5.4. The block tri-diagonal matrix formed in (4.11) can be solved efficiently using the block LU decomposition algorithm.

As seen here the lined-up cells are solved in a strong coupled manner by keeping all them implicitly in the left-hand-side matrix (preconditioner). The non-linear equation (4.10) is still solved at each time step without further linearization by using inner iterations and the latest available updates for the other neighbors. To achieve maximum efficiency, outside the anisotropic boundary layer region the cell-based BLU-SGS is applied. Therefore an inner sweep procedure starts from a forward line solver for the anisotropic boundary layer regions so as to relieve the stiffness before further computations, then a forward cell solver followed by a backward cell solver for other regions, finally a backward line solver for the boundary layer region.

The algorithm given in (4.11) is compact and generic, independent of spatial discretization. It can be used as a black box to speed up high-order cell-based implicit method in general.

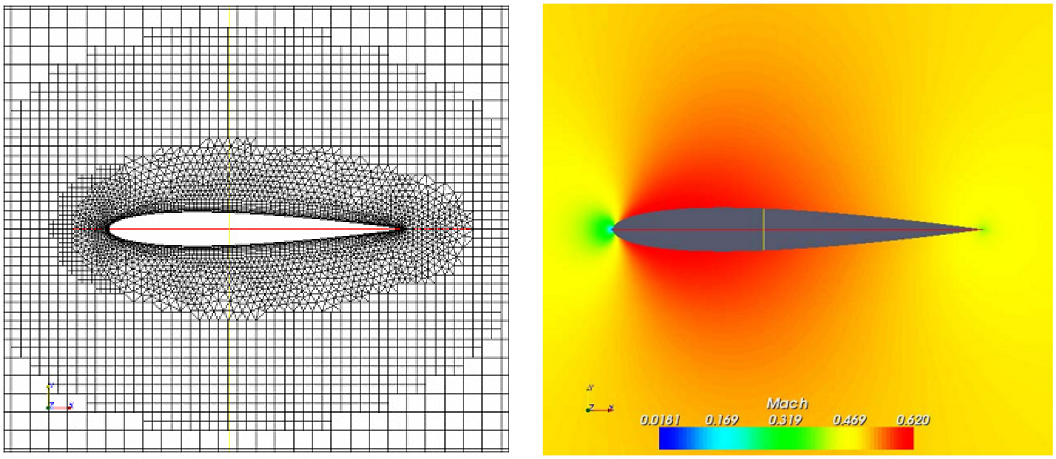
In this study, various solution line constructions including some of those proposed in [81,38] have been tested, and we found that based on the present partial cell/partial line BLU-SGS method, the anisotropy stiffness is indeed concentrated in the thin boundary layer near wall, and it seems that the simple line construction method of uniformly lining up several layers of grids above wall gives the satisfactory results to relieve the stiffness.

4.5 Numerical tests

The present line implicit method has been tested rigorously with a variety of numerical cases for its convergence rate. To demonstrate the true performance of the line solver, the best possible base cell solver, i.e. BLU-SGS method is first tested, then the convergence rate from the line solver is compared with the best results from the cell implicit solver.

4.5.1 Finite volume method test on the cell BLU-SGS solver

The present generic cell BLU-SGS formulation is tested with a face-based Finite Volume method developed by the present author by using simple subsonic inviscid flow around a NACA0012 airfoil, $Mach = 0.5, \alpha = 0°$. The inviscid Riemann flux is approximated by Rusanov flux. The explicit scheme used for comparison is three-stage TVD Rounge-Kutta scheme. A triangular and quadrilateral mixed mesh is used as in Figure 4.4(a). Figure 4.4(b) gives the symmetric flow field result of Mach contours as expected. As shown in Figure 4.5 the present cell implicit method is much faster than the explicit method, which validates the present cell LU-SGS method coupled with the FV scheme.



(a) **(b)**

Figure 4.4 Subsonic inviscid flow around NACA0012.
(a) Mixed mesh; (b) Mach contours.

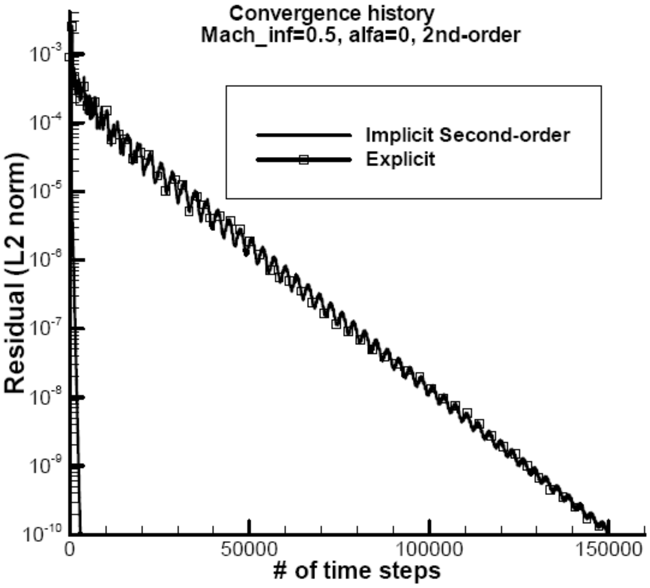
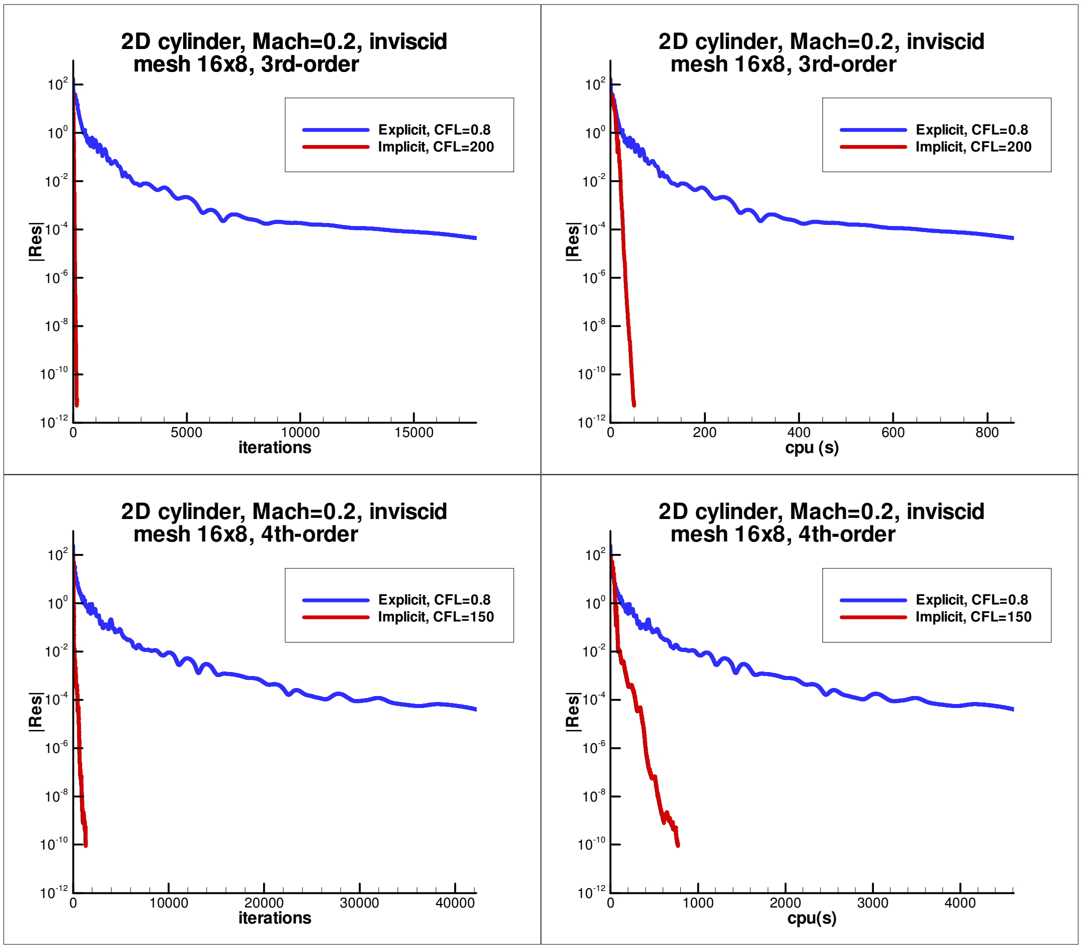


Figure 4.5 Comparison of convergence rate for NACA0012 subsonic flow.

4.5.2 Quadrature-Free SV method test on the cell BLU-SGS solver

The present generic cell BLU-SGS formulation is also tested with the Quadrature-Free SV method developed in Chapter 2 of the present dissertation by using the same test case (Figure 2.9) as in Section 2.4.4 as discussed in Chapter 2, i.e. the subsonic inviscid flow $(Mach = 0.2)$ around a 2D cylinder with curved boundary enforced in the boundary treatment. The inviscid Riemann flux is approximated by Rusanov flux. The explicit scheme used for comparison is three-stage TVD Rounge-Kutta scheme. Figure 4.6 shows the convergence history on the coarse triangular grids $(16 \times 8 \times 2)$ for 3rd-order and 4th-order cases. As shown in Figure 4.6, the maximum stable CFL number for the explicit method is $CFL = 0.8$, which is much less than that of the present implicit cell method, $CFL \geq 150$. It is also shown that both the iteration steps sand CPU time needed for the present cell implicit method to converge to machine zero is much faster than the explicit method. At first look it seems that the explicit method is too slow compared with the implicit method in Figure 4.6, but actually this is well-known true in general for high-order methods, therefore the present cell BLU-SGS method coupled with the QFSV scheme is valid.

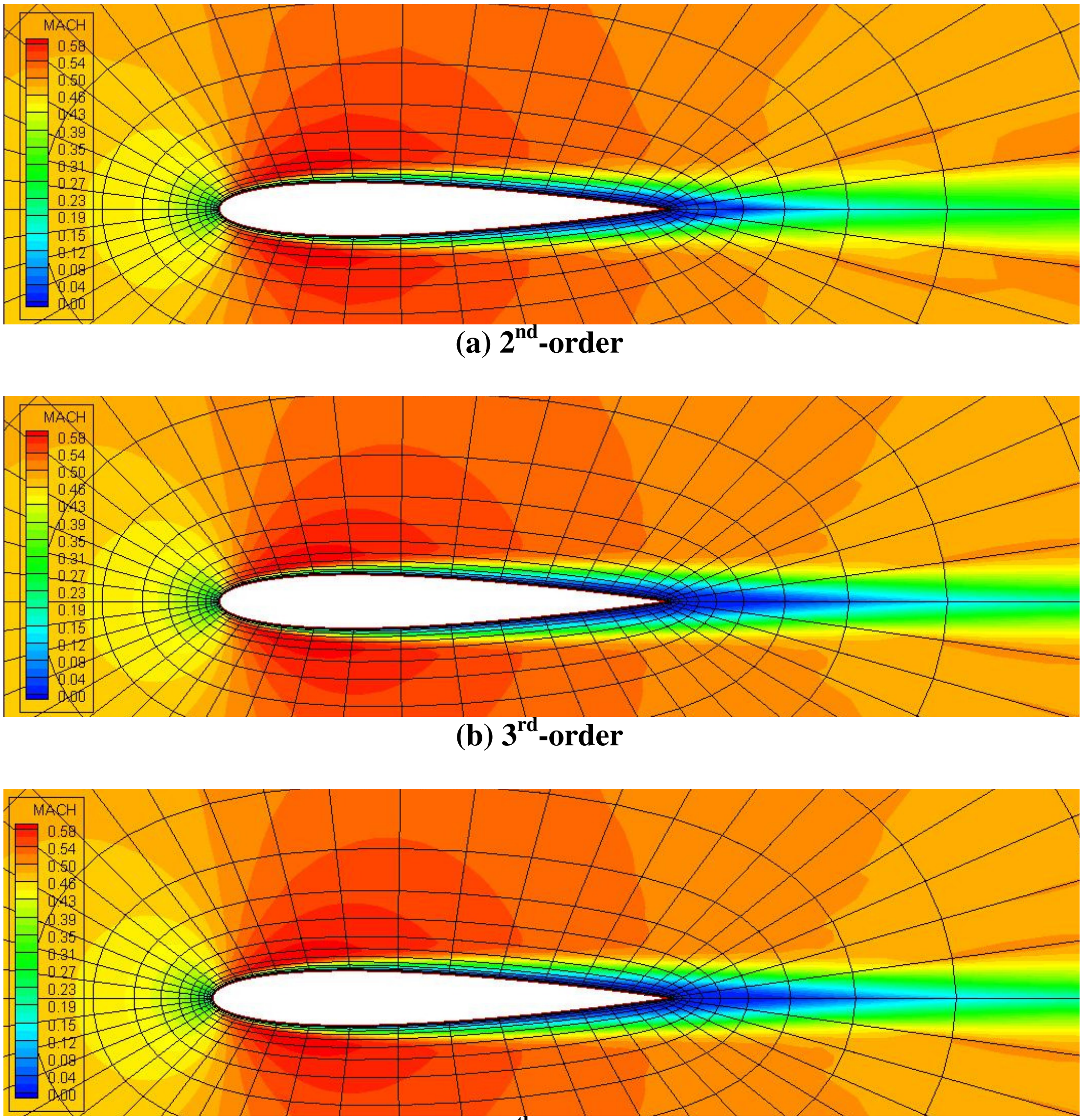


(a) **(b)**

Figure 4.6 Subsonic inviscid flow around cylinder.
(a) Iteration steps; (b) CPU time.

4.5.3 Test of SD with BR2 viscous flux on the cell BLU-SGS solver

Here we only test the effectiveness of the BR2 viscous fluxes with the Spectral Difference method coupled with the cell BLU-SGS solver. Other validations for SD with BLU-SGS can be found in [110,108]. The test case is a subsonic viscous laminar flow around NACA0012 with separation. The Reynolds number is $5000, Mach = 0.5, \alpha = 0°$. The two coarse grids used are: 1) total cell number = 640 and maximum wall grid aspect ratio AR= 60; 2) total cell number = 960 and maximum wall grid aspect ratio AR= 100. The results from the 2nd-order, 3rd-order, and 4th-order schemes are given in Figure 4.7 and Figure 4.8. Figure 4.7 shows the converged flow fields of Mach contours with the second grids (960 cells, AR=100) for different order of schemes. Also given in Figure 4.7 is the grids distribution, which is actually coarse compared with the usual grids used for low-order schemes. The converged surface friction results from both the upper and lower wall of NACA0012 are given in Figure 4.8. The results shows good flow symmetry around x-direction as expected. It shows that on both of the two coarse grids the 3rd-order and 4th-order schemes predict the separation point much better than the 2nd-order scheme, because the 3rd-order and 4th-order results converge together, but 2nd-order does not, as seen from the close-up views on the right side.

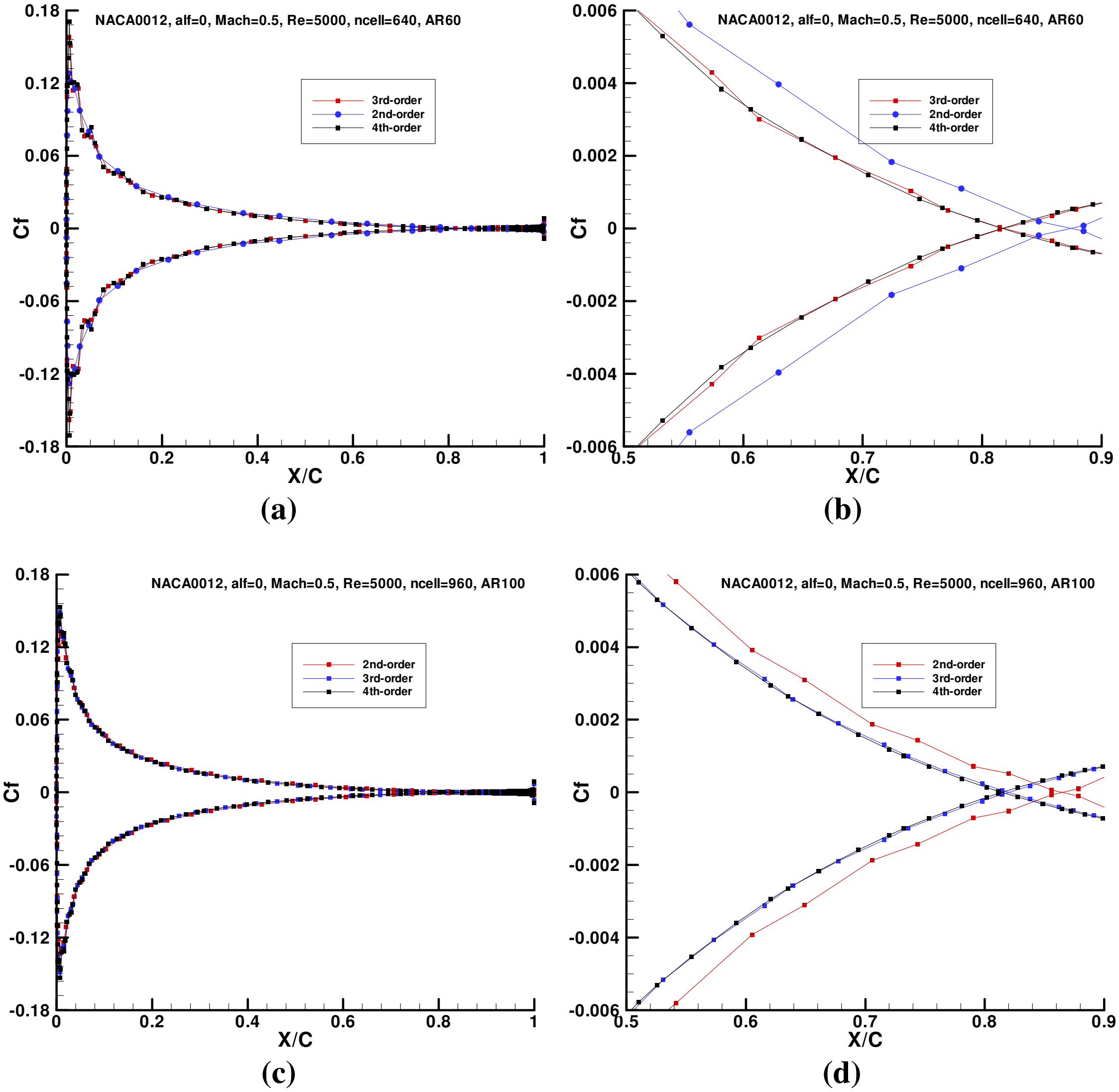


(a) 2nd-order

(b) 3rd-order

(c) 4th-order

Figure 4.7 Mach contours for subsonic viscous flow around NACA0012. Mach=0.5, Re=5000. Grids: 960 cells, Maximum AR=100.



(a)

(b)

(c)

(d)

Figure 4.8 Surface friction coefficient for viscous flow around NACA0012. Mach=0.5, Re=5000.

(a) 640 cells, Maximum AR=60; (b) Close-up view for (a);

(c) 960 cells, Maximum AR=100; (d) Close-up view for (c);

4.5.4 CPU time test for line implicit solver

CPU time costs are compared between the present BLU-SGS and the cell BLU-SGS method in Table 4.2 and Table 4.3 for 3D cells to verify the efficiency of the present numerical approach for LHS matrix computation and the present block tri-diagonal matrix solver. The full line solver with all the cells on the normal direction lined-up is used for easy distinguish from the cell implicit solver. The line solver has two more matrices to compute with size per cell as listed in Table 4.1. Table II gives the CPU times from a flat plate test case with straight-facet (linear) cells, where one time computation for the LHS matrices on the whole domain and one inner forward and backward sweep are given for both the cell BLU-SGS method and the present line BLU-SGS method. The line solver costs 2.4, 3.1, and 4.8 times CPU as much as the cell solver for 2nd-order, 3rd-order, and 4th-order schemes, respectively. Beside the major fact that there are two more matrices (off-diagonal) per cell in the line solver than in the cell solver, so the matrix cost can be 3 times as much as that in the cell solver, there are two opposite side effects on the above cost factors: 1) LU decomposition costs more CPU in higher-order matrix than in lower-order one; 2) No need to repeat the computation of the common non-incremental residuals for each cell matrix (by computing them first for all the cells on the domain) saves CPU time. In 2nd-order case, the second side effect surpasses the first effect, resulting in the factor of 2.4 less than 3; in 3rd-order case, both the two side effects almost balances to give the factor of 3.1; in the 4th-order case, LU decomposition costs a lot more than the saving from the common residual computations, thus the factor of 4.8 is bigger than 3.

Also Table II gives the CPU cost for one inner iteration step. It shows that the present efficient algorithm for block tri-diagonal matrix system costs is competitively close to that of 2nd-order scheme; for 3rd-order and 4th-order the ratio of the line solver and cell solver is close to the optimal factor of 3. Similar results are shown in Table III for NACA0012 with curved (quadratic) cells on wall boundary, and almost the same optimal ratio is obtained as the straight-facet linear cell case.

Table 4.1. LHS Matrix size per cell

Order	Cell BLU-SGS	Line BLU-SGS	Ratio
2	40^2	3×40^2	1
3	135^2	3×135^2	11.4
4	320^2	3×320^2	64

Table 4.2. CPU time (sec) test on flows on flat plate (grids $30 \times 20 \times 1$ cells)

Order	LHS		Ratio L/C	One inner iteration		Ratio L/C
	Cell BLU-SGS	Line BLU-SGS		Cell BLU-SGS	Line BLU-SGS	
2	1	2.36	**2.4**	0.071	0.096	**1.4**
3	9.2	28.3	**3.1**	0.324	0.81	**2.5**
4	58.3	281.8	**4.8**	0.920	2.76	**3.0**

Table 4.3. CPU time (sec) test on flows around NACA0012 (grids 768 cells)

Order	LHS		Ratio L/C	One inner iteration		Ratio L/C
	Cell BLU-SGS	Line BLU-SGS		Cell BLU-SGS	Line BLU-SGS	
2	1.3	3.0	**2.3**	0.093	0.12	**1.3**
3	11.8	36.6	**3.1**	0.41	1.0	**2.4**
4	74.9	357.8	**4.8**	1.2	3.4	**2.8**

4.5.5 Robustness test for the line implicit solver

To test if the line implicit solver is more robust than the cell solver for high Reynolds number flows, a relatively extreme case is used: 4th-order scheme, Reynolds number= 10^8, Mach=0.3, flat plate boundary layer flow, three coarse grids with 30×20 cells and three different maximum aspect ratios: max AR=100, 1000, 10000. The *CFL* number starts from $CFL_0 = 0.01$, then increases exponentially as iteration steps according to $CFL_n = CFL_0 * 1.2^n$ until reaching a maximum *CFL*. Figure 4.9(a) shows the expected good convergence to machine zero from the full line solver (with all the cells in wall normal direction lined up for each line). But the cell solver failed in this case: the iteration blows out after about 70 steps, as seen in Figure 4.9(b).

4.5.6 Grid aspect-ratio insensitivity test for the line implicit solver

The line implicit method eliminates the anisotropy stiffness by lining-up the cells in normal direction. The insensitivity of convergence rate to varying grid aspect ratios in the present line implicit method has already been demonstrated in Figure 4.9 (a) for 4th-order. Figure 4.10 shows the results for 2nd-order and 3rd-order cases, with Reynolds number= 10^5, Mach=0.3, flat plate boundary layer flow, three coarse grids with 30×20 cells and three different maximum aspect ratios: max AR=100, 1000, 10000.

4.5.7 2D viscous flow over flat plate with line implicit solver

Figure 4.11 gives the solution lines in different colors. Only several grid layers near the wall are lined up. It has been tested out that this simple uniformly lining-up achieves the best convergence results.

Figure 4.13 compares the convergence rate for a 2D flat plate boundary layer flow at $Mach = 0.3$, Reynolds number $= 10^4$ for 2nd- and 3rd-order, and 10^5 for 4th-order. The higher Reynolds number is used for 4th-order to show advantage of high-order scheme for the same grids. The grids with total of 30×20 cells are clustered near wall and leading edge of the flat plate. The maximum aspect ratio for wall grids is 225. 12 layers of cells are lined-up for 2nd- and 3rd-order, and 6 layers for 4th-order. The comparisons are played on the "fair ground" by tuning the cell solver to be fastest in each case. . It is shown that much less iteration steps (for example one order less steps for the 3rd-order case) needed for the line solver than the cell solver, which demonstrates the effectiveness of the present line solver to relieve anisotropy stiffness. The more interesting results are the significant savings (one to three times) on CPU time obtained from the line solver compared to the cell solver. Among them, the 3rd-order scheme performs best in terms of computer time saving.

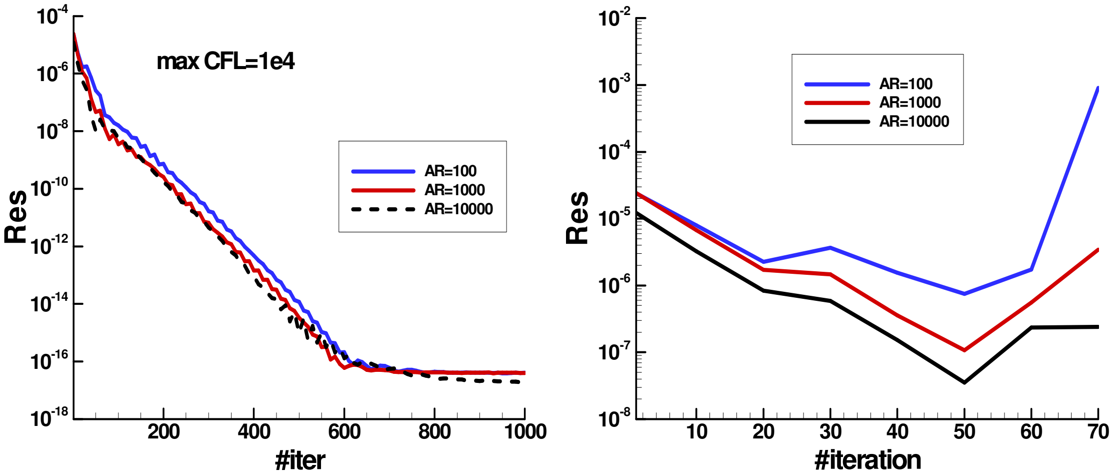
4.5.8 Subsonic viscous flow around NACA0012 with line implicit solver

This case is to test the present line implicit method on curved boundary with non-linear (quadratic) boundary cells. A subsonic laminar flow around NACA0012 with $Mach = 0.5,$ Reynolds number =5000 is computed by using a coarse grid with total of 960 cells and maximum wall grid aspect ratio of 100. The uniform line construction for several layers of cells near wall is illustrated in Figure 4.12 with the lines in different colors. Again this simple line construction gives good performance of convergence as used in the flat plate boundary layer cases.

Figure 4.14 compares the convergence rates of the present line implicit method by using different number of cell layers lined-up near wall. As seen in this case the optimal layer numbers in terms of CPU cost for the 2nd, 3rd-, and 4th-order schemes are 7, 9, and 7, respectively (in 4.13(c) the computation is deliberately stopped for other layer number once we found the optimal one is good enough.) The balance of combining the cell sweeps and line sweeps reaches its best from the above optimal layer numbers in this NACA0012 test case. These best results from the present line BLU-SGS solver are compared with the best results from the pure cell BLU-SGS solver in Figure 4.15. It is shown that the line solver is about twice as fast as the cell solver for the 2nd- and 3rd-order cases. For the 4th-order case, the line solver yields more than one time decrease of iteration steps compared with the cell solver, which shows the effectiveness of the line solve to overcome the anisotropy stiffness. But in this case, the line solver does not show much CPU time saving in this 4th-order case, partly because here the size of the LHS matrices is much larger than 2nd- and 3rd-order cases.

4.6 Conclusions

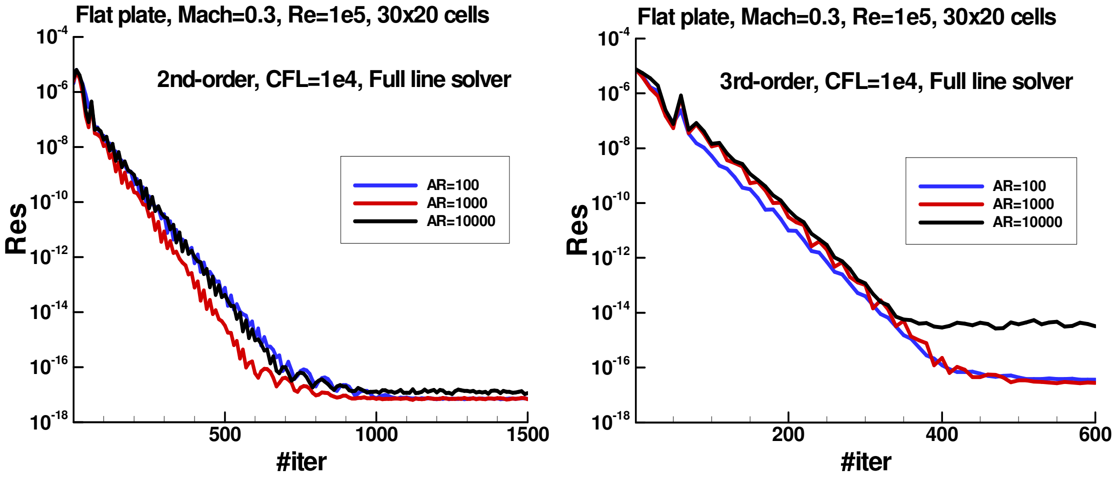
A low-storage, efficient and robust line implicit solver has been successfully developed to overcome anisotropy stiffness due to highly stretched grids in high Reynolds number boundary layer flows. The efficiency and robustness of the cell solver is important for building a line solver based on it. A compact formulation of the cell-based Block LU-SGS method is given and a generic cell-based implicit solver has been developed to be served as a black box for general use in implicit methods to speed up solution convergence, and its effectiveness has been demonstrated in the FV, SV, and SD methods. The second approach of Bassi and Rebay viscous flux (BR2) that was originally developed for DG methods has been coupled into the SD methods to enhance robustness of the cell implicit solver. The new line implicit method overcomes anisotropy stiffness by direct coupling the neighboring cells in the wall normal direction while preserving the low-storage and compactness features of the original BLU-SGS method. Up to 3 times of saving on CPU time has been demonstrated compared with the cell BLU-SGS solver. The present line implicit method also shows better robustness than the cell BLU-SGS solver in some high Reynolds number flows with high-order scheme. The present line implicit method is formulated in a compact and generic form and the solver has been programmed as a black box so as to be easily applied in general high-order methods.



(a) The full line solver

(b) The cell solver

Figure 4.9 Robustness test for full line solver with 2D flat plate flow. 4th-order scheme, Re=10^8, Mach=0.3, coarse mesh $30 \text{x} 20$ cells.



(a) 2nd-order

(b) 3rd-order

Figure 4.10 Robustness test for full line solver with 2D flat plate flow. Re=10^5, Mach=0.3, coarse mesh $30 \text{x} 20$ cells.

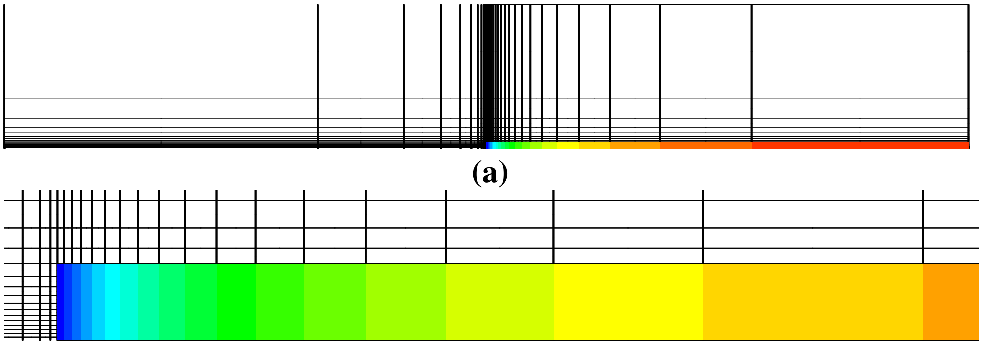


Figure 4.11 Grids and solution lines for flat plate boundary layer. (a) The whole domain; (b) Close-up view near wall.

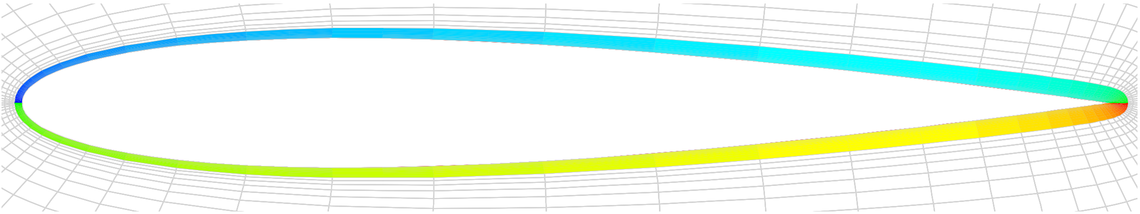


Figure 4.12 Grids and solution lines near wall for NACA0012 subsonic flow.

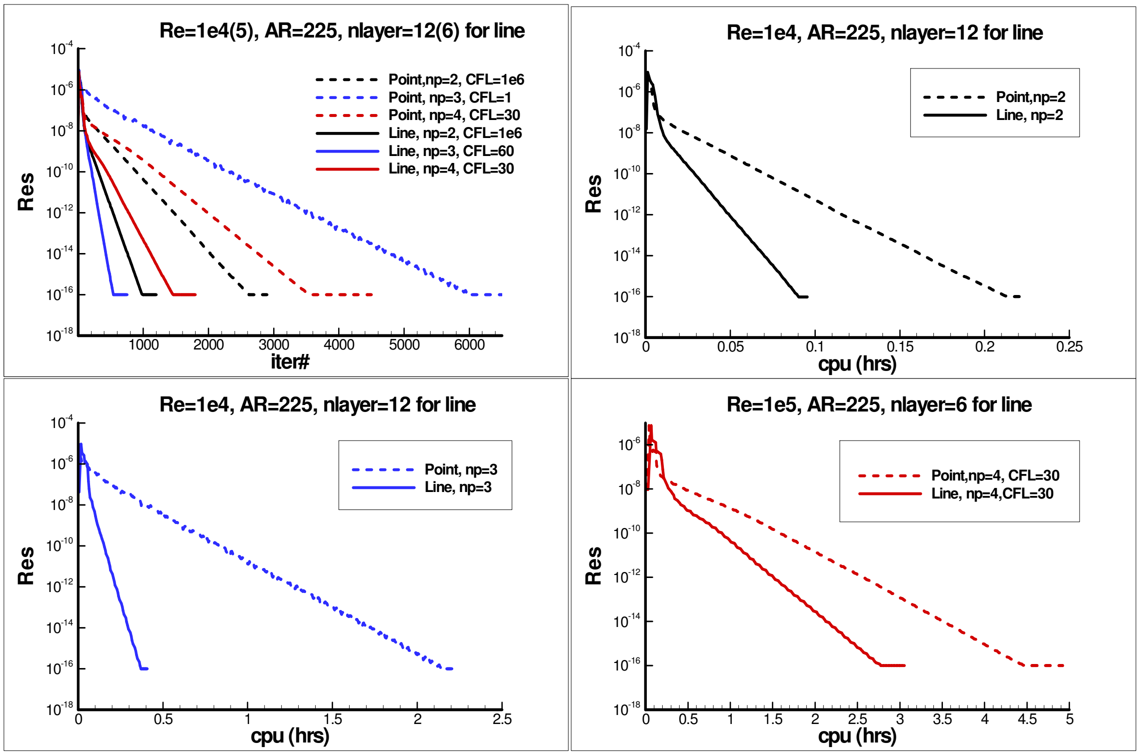
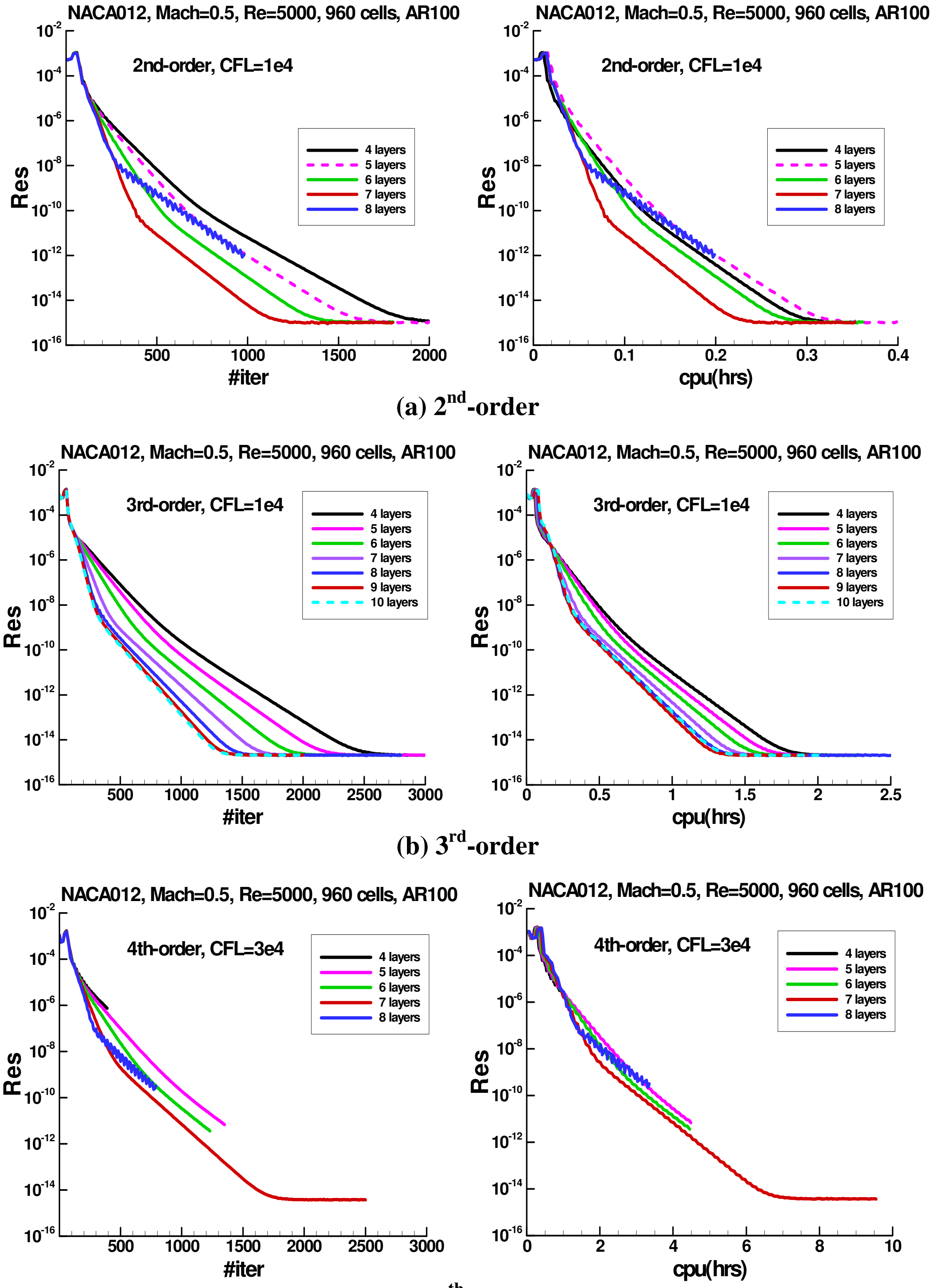


Figure 4.13 Comparisons of convergence for flat plate boundary layer. Maximum AR=225, Coarse mesh 30×20, Mach=0.3, $Re = 10^4$ for 2nd-, 3rd-order, and 10^5 for 4th-order. "Point" denotes for the cell BLU-SGS solver, "Line" for the present line BLU-SGS solver, "np" for accuracy order, "CFL" for best available CFL number. 12 layers of cells lined-up for 2nd- and 3rd-order, and 6 layers for 4th-order. Solid line: line solver; Dashed line: cell solver.

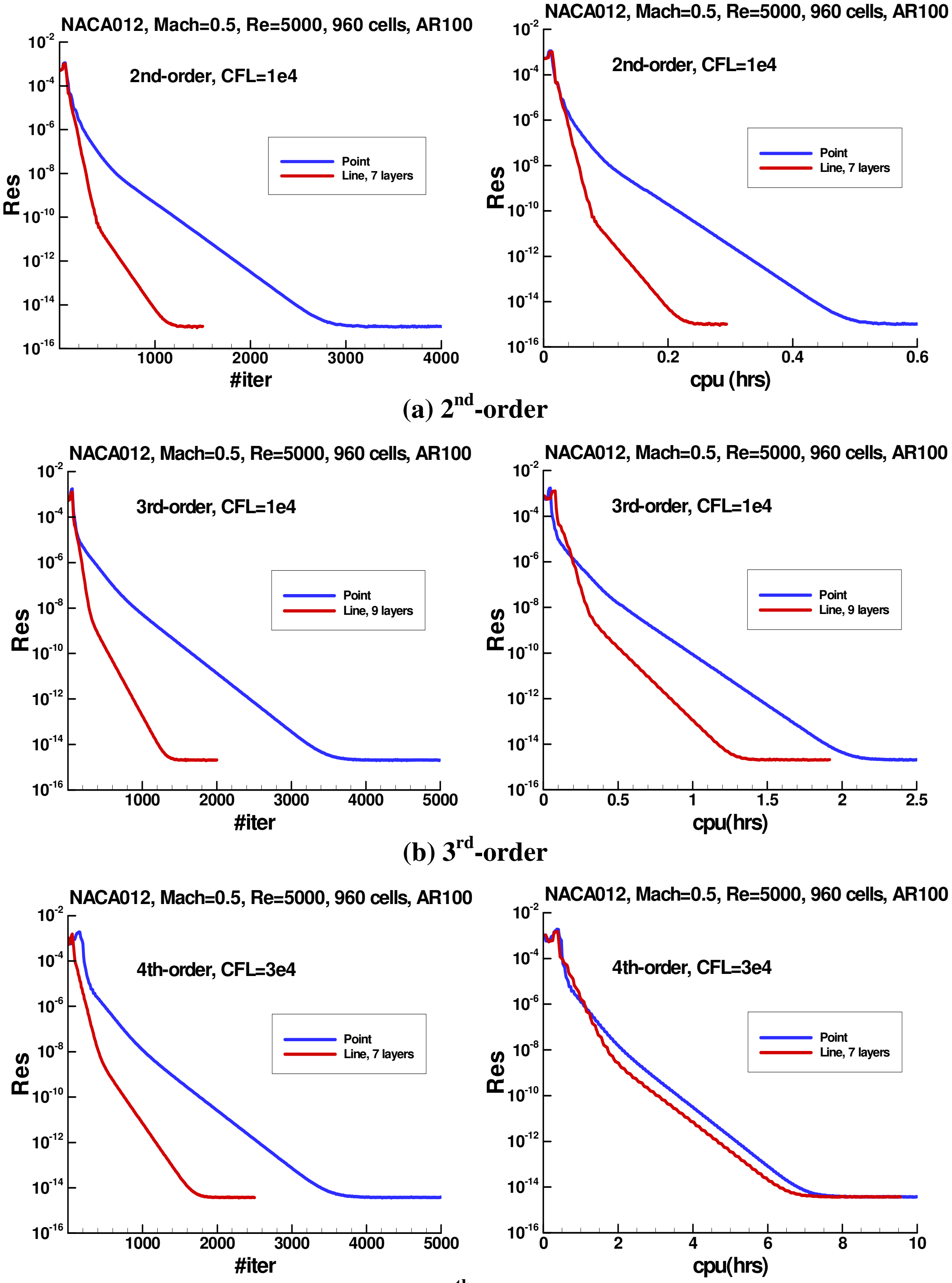


(a) 2nd-order

(b) 3rd-order

(c) 4th-order

Figure 4.14 Convergences with different layer numbers of lined-up cells.



(a) 2nd-order

(b) 3rd-order

(c) 4th-order

Figure 4.15 Convergences for NACA0012 laminar flow.

Red line (“Line”): the present line solver; Blue line (“Point”): the cell solver.

CHAPTER 5 HIGH-ORDER UNSTRUCTURED MESH GENERATION

In this chapter, we propose a novel and fully automatic algorithm that guarantee to resolve a common problem in high-order mesh generation, i.e. the gridline intersection problem for anisotropic 3D hexahedral boundary cells used in high Reynolds number flows. The present method is robust and fast for correction of self-intersection without changing the basic aspect ratio of the original grids or degrading the original grid quality.

5.1 Introduction

The curved boundaries should be accordingly represented by high-order boundary mesh in a high-order method. The linear mesh cells (with line-segments or planar facets) used to represent the curved boundaries in second order simulations is compatible with the linear data interpolations used in a second-order method. But for high-order methods, the error generated from the linear element representation for curved boundary must eventually affect not only the boundary region, but also transport elsewhere in the flow field resulting in degraded accuracy order and rendering uselessness of a higher-order scheme [13,126]. One can always use very fine linear cells on curved boundary to reduce this error, but cannot eliminate its pollution effect. Moreover, coarser mesh is actually expected in a high-order method; otherwise it loses its advantage compared with a low-order method. Currently the ability to generate suitable high-order meshes (at least quadratic for curved boundary) for 3D complex geometries is a significant limiting factor for applying high-order methods in industry, because almost all the available grid generation packages can only generate linear cells.

The traditional low-order mesh generation package only relies on the linear segments or planar facets, it has no knowledge of the real shape of the boundary. The usual ways to generate unstructured volume mesh with curved boundary is the so called "direct generation" method, which generates extra boundary vertices directly on the underlying representation of the curved boundary given by NURBS from a commercial CAD package. For the extra boundary vertices, there are various surface mesh generation algorithms, which can be classified as either 2D parametric space [67,41,20,32,116] or direct 3D [34,73,37,19,15]. However, the problem of the formation of high-order cells on arbitrary 3D surfaces is still under research. Besides, all the issues that have been addressed in traditional 2nd-order mesh generator, such as mesh quality-control, intersection check, and automatic generation as well as adaptive, anisotropic, parallel mesh generation, and geometry management, needs to be re-examined to produce a commercial high-order mesh generation package, which means to abandon the previously well-developed low-order mesh generator and re-invest huge amount of efforts for a brand-new one.

Another approach as a shortcut to generate high-order boundary mesh still utilizes the traditional low-order mesh generation package. Unlike the above direct way, first a low-order linear mesh is generated from the traditional mesh generation package, then extra vertices (depending on accuracy order needed) are topologically inserted into a low-order linear cell near the curved boundary, finally the newly inserted vertices are geometrically moved back to the curved boundary. This method serves as a post-processing step for the traditional mesh generator. This post-processing step does not cost much additional time and indeed saves the usefulness of the mesh generation package for curved boundary cells, yet except one problem educed from the insertion. The problem is self-intersection caused by curved cell faces that are too close to each other and may overlap. As an example, for the highly stretched boundary cells that used in high Reynolds number flow, a newly generated boundary gridline by inserting vertices might be intersected with some interior gridlines near the curved boundary, as illustrated in Figure 5.1.

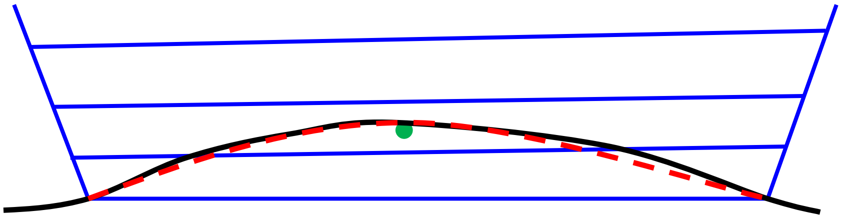


Figure 5.1 A self-intersected cell on curved boundary.

Black line: Curved wall; Blue line: Gridline for linear cell; Red dash: Boundary gridline for quadratic cell; Green dot: Inserted

The self-intersection problem is very common in high-order mesh generation and makes the mesh invalid in CFD simulations, thus should not be allowed. In this chapter a novel and fast algorithm has been developed to resolve intersections of boundary grid layers.

5.2 Algorithm

We only consider unstructured hexahedral mesh, which is usually used in high Reynolds number boundary layer flows. If the curved boundary cell is highly stretched, the self-intersection can happen in probably two ways: 1) any of the boundary edges of the boundary cell could intersect with the inner edge on the same face (Figure 5.2a); 2) some local extrema points on the boundary face could intersect with the opposite inner face (Figure 5.2b). The second case indicates that the mesh is too coarse to correctly represent some main geometric features of the curved boundary, therefore back to the linear mesh generation step, mesh refinement or adjustment has to be done in the mainstream flow direction. After that the self-intersection problem is transformed to the first case.

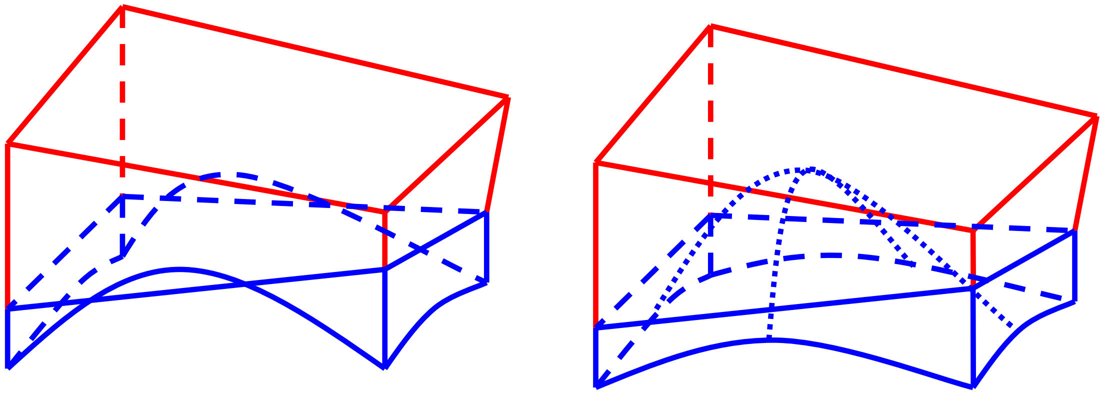


Figure 5.2 Curved boundary face for a non-linear cell.
(a) Edge intersection; (b) local extremum on curved boundary face.

Suppose a curved wall boundary that is approximated by a quadratic boundary curve, as illustrated in Figure 5.3. The quadratic boundary gridline is intersected with the linear interior gridline due to the stretched cell on the curved wall. We fix this problem by curving the interior edge $\overline{CD}$ (red solid line) into $\widehat{CD}$ (red dashed line). First a self-intersected boundary cell are identified if

$$d_{12} > factor \cdot d_{13}, \tag{5.1}$$

where d_{12} and d_{13} are the distance between point 1 and 2, and between point 1 and 3, respectively. Point 1, 2, and 3 denote for the mid-points on linear boundary gridline segment (between point A and B), curved boundary gridline segment, linear interior gridline segment (from C to D), respectively. $factor < 1$ is a safety factor, and usually set $factor = 0.5$.

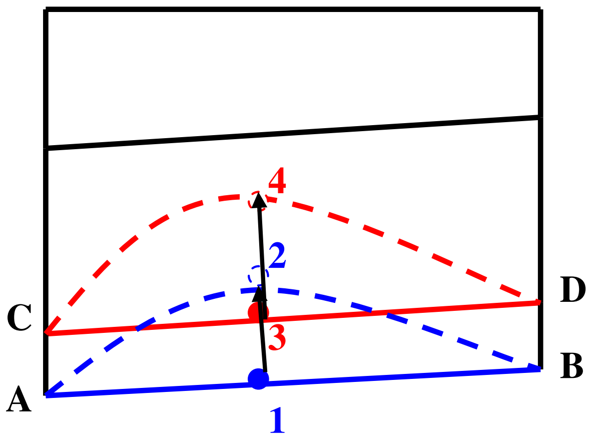


Figure 5.3 Curved edges for a non-linear cell.

Blue solid line: Linear boundary gridline; Blue dashed line: Quadratic boundary line; Red solid line: Linear interior gridline; Red dashed line: Curved interior gridline.

Then the interior gridline that is intersected with the curved boundary gridline is curved by lifting the mid-point (from point 3 to point 4) according to,

$$\vec{r}_4 = \vec{r}_3 + \vec{d}, \tag{5.2}$$

where $\vec{r}_3$ and $\vec{r}_4$ are the coordinates for the mid-points 3 and 4, respectively; $\vec{d} = \vec{r}_2 - \vec{r}_1$, and $\vec{r}_1$ and $\vec{r}_2$ are the coordinates for the mid-points 1 and 2, respectively. As seen here, the basic aspect ratio of the original mesh is not changed.

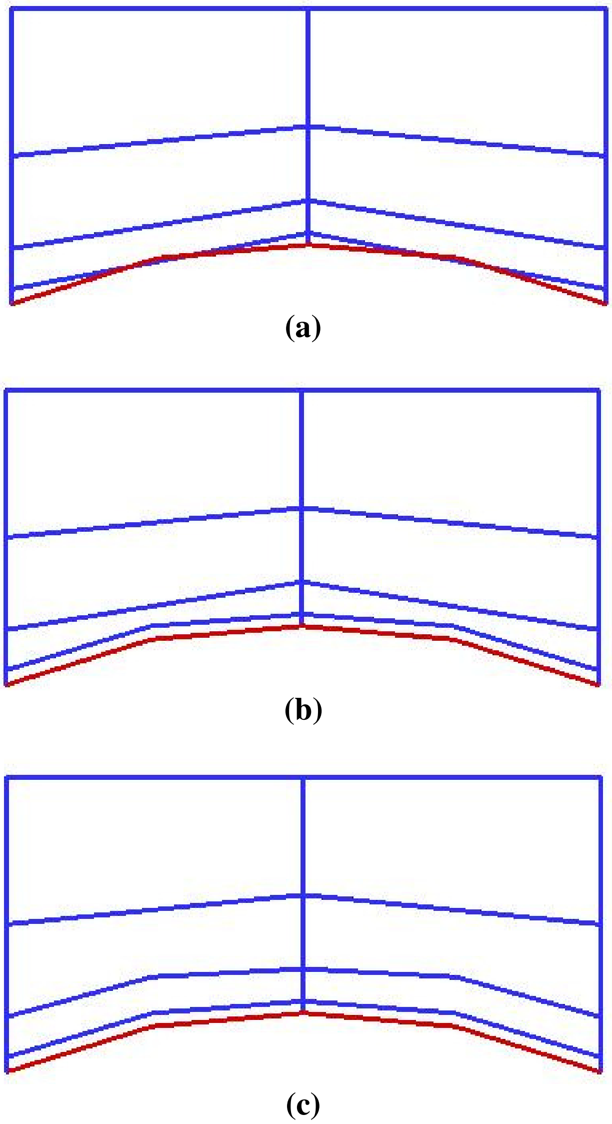
The above procedure is repeated for other edges of the interior cell until no interior cell needs to be modified. A recursive adjustment procedure is then applied to all the boundary cells.

This correction algorithm for self-intersection is fast because actually only a few grid layers above the wall need to be modified. Extension of the present algorithm to cubic cell or other high-order cell is straightforward. The isoparametric cell method widely used in high-order Finite Element method is still valid here to map the curved hexahedral cells (now including some interior cells) into standard cubic cell.

5.3 Test results

In the following tests hexahedral cells are used for unstructured mesh, but we will only show the self-intersection problem in the wall normal direction. Since these are unstructured meshes, the problem is identical in other directions with curved boundary. Figure 5.4 shows a test on a simple domain with 8 cells including 2 curved boundary cells. The two curved (quadratic) boundary gridlines is found to intersect with the two interior gridlines, respectively, in the original mesh in Figure 5.4(a). Figure 5.4(b) shows that the self-intersection is corrected by curving the two next upward interior edges after using the present algorithm with *factor*=0.5. Figure 5.4(c) shows that more interior edges are curved if using smaller *factor*=0.2.

Figure 5.5 gives the test results from NACA0012 airfoil with total of 640 cells, maximum AR=60 by setting *factor*=0.5. Besides the boundary cells, some interior cells near wall are also marked to be modified as curved (quadratic) cells, which clearly shows that the present algorithm for correction of self-intersection is necessary and valid. Figure 5.6 gives zoom-in view for some leading region part with comparisons between the original mesh and the modified mesh. It shows that the present algorithm has successfully resolved the self-intersection problem for the highly stretched cells near the curved wall.



(a)

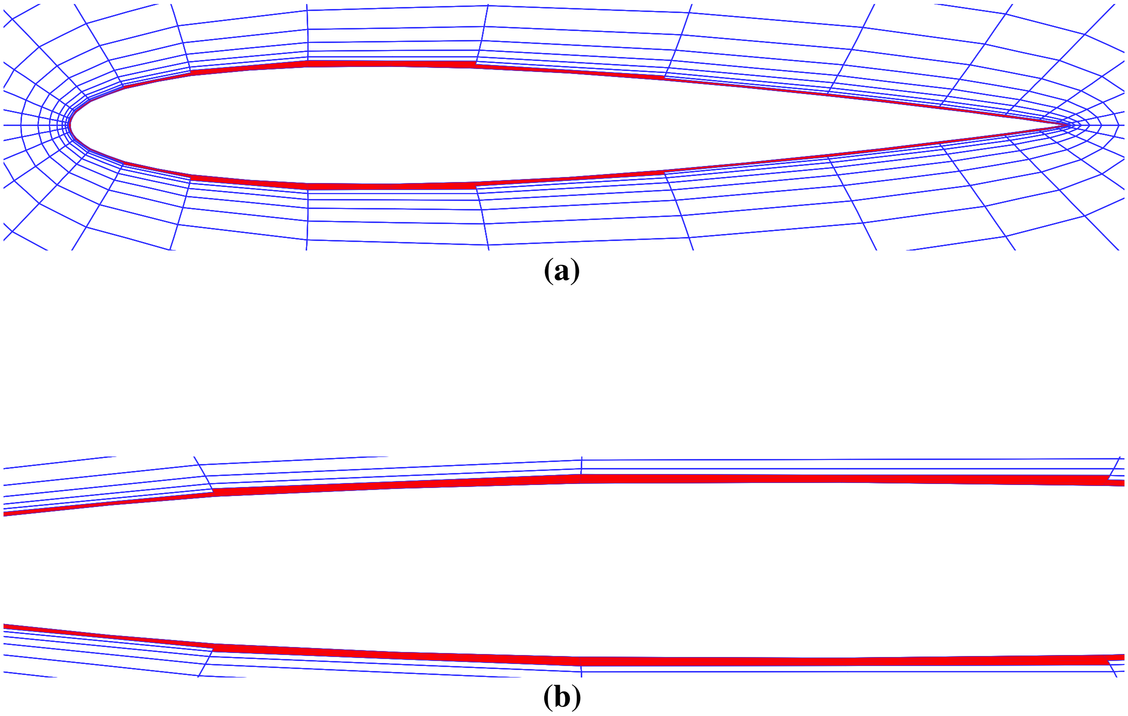
(b)

(c)

Figure 5.4 Comparison test on curved boundary on a simple domain.

(a) Original mesh; (b) modified mesh, *factor*=0.5; (c) modified mesh, *factor*=0.2.

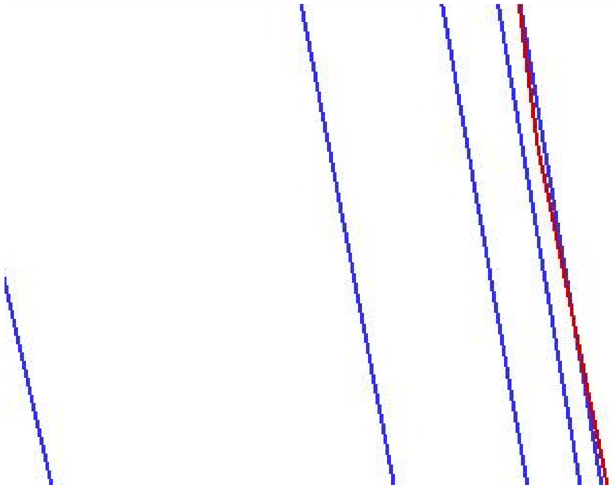
Blue line: Interior gridline; Red line: Curved boundary gridline.



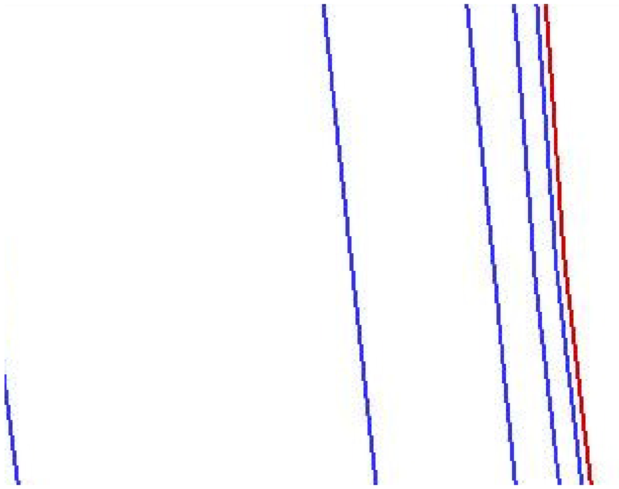
(a)

(b)

Figure 5.5 Curved boundary test on NACA0012.
Total 640 cells, maximum AR=60, *factor*=0.5.
Red color mark: curved cells. (b) is the close-up view in (a).



(a)



(b)

Figure 5.6 Comparison test on curved boundary on NACA0012.
Zoom view of local leading edge region. Total 640 cells, maximum AR=60.
Red line: curved boundary gridline; Blue line: interior gridline.
(a) Original mesh; (b) modified mesh, *factor*=0.5.

5.4 Conclusion

A guaranteed algorithm to resolve self-intersection problem in high-order mesh generation is presented, which is robust and fully automatic. The present algorithm offers the advantage of correcting grid self-intersection without changing the basic aspect ratio of the original grids or degrading the original grid quality. The present algorithm has been successfully carried out for solving the gridline intersection problem with anisotropic quadrilateral and hexahedral boundary cells used in high Reynolds number flows.

CHAPTER 6 SUMMARY AND FUTURE WORK

Several new methods have been developed to meet the critical and diversified challenges in the state-of-art unstructured-grids based high-order methods for 3D real-world applications:

1. Parameter-free high-order generalized moment limiter for arbitrary mesh. Firstly the discontinuity marker created in this method does not need any user-specified free parameter to detect the discontinuities and exclude the smooth extrema. Secondly the limiter has been designed to be naturally compact and efficient. Finally it is generic, which can be applied to arbitrary mesh and all the high-order methods.

2. Efficient line implicit solver with several new features including: 1) a scheme of a line BLU-SGS solver for the lined-up cells within the anisotropic thin boundary layer coupled with a cell BLU-SGS solver for other regions of less anisotropy stiffness, which significantly improves convergence rate for highly stretched wall grids. Up to 3 times of saving on CPU time has been demonstrated compared with the cell BLU-SGS solver. The present line implicit method also shows better robustness than the cell BLU-SGS solver in some high Reynolds number flows with high-order scheme. The present line implicit method is formulated in a compact and generic form and the solver has been programmed as a black box so as to be easily applied in general high-order methods. 2) low memory storage requirement due to the partial line solver/partial cell solver scheme and an efficient low-storage strategy for LU decomposition of the cell Jacobians; 3) robust and accurate viscous fluxes for anisotropic grids based on the second approach of Bassi and Rebay (BR2); 4)

generic and compact formulation and coding as a black box so as to be easily applied in general high-order methods.

3. Efficient quadrature-free SV methods for 3D application. This approach has improved the original SV method by replacing the large number of quadrature for face integrals in 3D case with many less nodal operations based on analytical shape functions. The analytical shape functions on the nodal points, which are to be used for flux reconstruction in the flow solver, have been pre-computed by using symbolic software such as Mathematica. The major contributions from the present author focus on the core parts of the 3D quadrature-free SV method, which include 1) found the complicated connectivity in 3D partition (linear, quadratic, and cubic) of a SV cell, which includes sub-faces, nodes, sub-cells (CVs), flux directions, and orientations relative to the neighboring cells; 2) successful computed those complicated shape functions for each node. Also the Korivanona's efficient method to deal with curved boundary has been coupled into this quadrature-free SV method.

4. High-order mesh generation for 3D hexahedral mesh. This novel and fully automatic algorithm guarantee to resolve the self-intersection problem for high-order quadrilateral or hexahedral mesh with strong robustness. The algorithm also offers the advantage of correcting grid self-intersection without changing the basic aspect ratio of the original grids or degrading the original grid quality.

For the future work, the above limiter and implicit method could be tested with and applied to the shock and boundary layer interaction problem, hypersonic viscous flow, or unsteady viscous supersonic problems, etc. Our preliminary result from the Spalart-Allmaras (S-A) turbulence model shows that the present implicit solver is promising for turbulent flow, but more investigation is needed to overcome the instability in S-A model computation and obtain better convergence. Also the present methods could be coupled with other turbulence models or LES/DNS methods. To promote a more efficient use of computer resources, the present methods could be applied as a smoother in a multi-grid approach.

BIBLIOGRAPHY

[1] Abgrall, R. (1994). On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics , 114*, 45-58.

[2] Abgrall, R. (2006). Residual distribution schemes: Current status and future trends. *Computers & Fluids , 35* , 641–669.

[3] Abgrall, R., & Roe, P. (2003). High-order fluctuation splitting schemes on triangular mesh. *Journal of Scientific Computing , 19*, 3-36.

[4] Atkins, H., & Shu, C.-W. (1998). Quadrature-free implementation of Discontinuous Galerkin method for hyperbolic equations. *AIAA J. , 36* (5).

[5] Barth, T., & Deconinck, H. (1999). High-order methods for computational physics. Berlin: Springer.

[6] Barth, T., & Frederickson, P. (1990). High-order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA Paper No. 90-0013.*

[7] Barth, T., & Jesperson, D. (1989). The design and application of upwind schemes on unstructured meshes. *AIAA Paper No.89-0366.*

[8] Barth, T., & Roe, P. (1990). and P.O. Frederickson PO. High-order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA Paper No. 90-0013.*

[9] Bassi, F., & Rebay, S. (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *J Comp Phys , 131* (1), 267–79.

[10] Bassi, F., & Rebay, S. (1997). High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys , 138*, 251-285.

[11] Bassi, F., & Rebay, S. (2002). Numerical evaluation of two discontinuous Galerkin methods for thecompressible Navier–Stokes equations. *Int J Num Meth Fluids , 40* (1), 197–207.

[12] Bassi, F., Crivellini, A., Rebay, S., & Savini, M. (2005). Discontinuous Galerkin solutions of the Reynolds-averaged Navier–Stokes and K2o turbulence model equations. *Comput Fluids , 34* (4-5), 507–40.

[13] Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., & Savini, M. (1997). A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In R. Decuypere, & G. Dibelius (Ed.), *Proceedings of 2nd European Conference on Turbomachinery*, (pp. 99–108). Antwerpen, Belgium.

[14] Biswas, R., Devine, K., & Flaherty, J. (1994). Parallel, adaptive finite element methods for conservation laws. *Appl. Numer.Math. , 14*, 255–283.

[15] Blacker, T. (1996). The Cooper Tool. *Proceedings of 5th International Meshing Roundtable*, (pp. 13-29).

[16] Boris, J., & Book, D. (1969). Flux corrected transport,1 SHASTA, a fluid transport algorithm that works. *J. Comput. Phys. , 11*, 38–69.

[17] Canuto, C., Hussaini, M., Quarteroni, A., & Zang, T. (1987). Spectral Methods in Fluid Dynamics. New York: Springer-Verlag.

[18] Caraeni, D., & Fuchs, L. (2005). Compact third-order multidimensional upwind discretization for steady and unsteady flow simulations. *Computers & Fluids , 34*, 419–441.

[19] Cass, R., Benzley, S., Meyers, R., & Blacker, T. (1996). Generalized 3-D Paving: An Automated Quadrilateral Surface Mesh Generation Algorithm. *Int. J. Numer. Meth Eng. , 39*, 1475-1489.

[20] Chen, H., & Bishop, J. (1997). Delaunay Triangulation for Curved Surfaces. *Proceedings, 6th International Meshing Roundtable*, (pp. 115-127).

[21] Chen, Q. (2006). Partitions for Spectral (Finite) Volume Reconstruction in the Tetrahedron. *Journal of Scientific Computing , 29* (3), 299-319.

[22] Chen, Q. (2006). Partitions of a simplex leading to accurate spectral (finite) volume reconstruction. *SIAM J. Sci. Comput. , 27* (4), 1458-1470.

[23] Chen, R., & Wang, Z. (2000). Fast, Block Lower-Upper Symmetric Gauss Seidel Scheme for Arbitrary Grids. *AIAA Journal , 38* (12), 2238-2245.

[24] Cockburn, B., & Shu, C.-W. (2001). Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing , 16* (3).

[25] Cockburn, B., & Shu, C.-W. (1998). The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J Numer. Anal. , 35*, 2440–2463.

[26] Cockburn, B., & Shu, C.-W. (1998). The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *J. Comput. Phys , 141*, 199-224.

[27] Cockburn, B., & Shu, C.-W. (1989). TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation , 52*, 411-435.

[28] Cockburn, B., Hou, S., & Shu, C.-W. (1990). TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation , 54*, 545-581.

[29] Cockburn, B., Karniadakis, G., & Shu, C.-W. (2000). The development of discontinuous Galerkin methods, in Discontinuous Galerkin Methods. Berlin: Springer.

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

[31] Cook, A., & Cabot, W. (2004). A high-wavenumber viscosity for high resolution numerical method. *J. Comput. Phys. , 195*, 594-601.

[32] Cuilliere, J. (1998). An adaptive method for the automatic triangulation of 3D parametric surfaces. *Computer-Aided Design , 30* (2), 139-149.

[33] Deconinck, H., Struijs, R., Bourgeois, G., & Roe, P. (1993). Compact advection schemes on unstructured meshes. *VKI Lecture Series 1993-04: Computational fluid dynamics.*

[34] Dey, S., M. O'Bara, R., & Shepard, M. (1999). Curvilinear mesh generation in 3D. *Proceedings of the Eighth International Meshing Roundtable*, (pp. 407–417).

[35] Douglas, J., & Dupont, T. (1976). Interior Penalty Procedures for Elliptic and Parabolic Galerkin Methods. In *Lecture Notes in Phys.* (Vol. 58). Berlin.

[36] Durlofsky, L., Enquist, B., & Osher, S. (1992). Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *J. Comput. Phys , 98* (64).

[37] Farouk, R. (1997). Optimal paramaterizations. *Comuter Aided Geometric Design , 14* , 153-168.

[38] Fidkowski, K., Oliver, T., Lu, J., & Darmofal, D. (2005). p-Multigrid solution of high-

order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations. *J. Comput. Phys. , 207*, 92–113.

[39] Fiorina, B., & Lele, S. (2007). An artificial nonlinear diffusivity method for supersonic reacting flows with shocks. *J. Comput. Phys. , 222*, 246-264.

[40] Friedrich, O. (1998). Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *J. Comput. Phys , 144*, 194-212.

[41] George, P., & Borouchaki, H. (1998). *Delaunay Triangulation and Meshing: Application to Finite Elements.* Hermes, France.

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

[43] Gottlieb, D., & Orszag, S. (1977). Numerical Analysis of Spectral Methods. *SIAM.* Philadelphia.

[44] Gottlieb, S., & Shu, C.-W. (1998). Total variation diminishing Runge–Kutta schemes. *Math Comput , 67*, 73–85.

[45] Haga, T., Ohnishi, N., Sawada, K., & Masunaga, A. (2006). Spectral Volume Computation of Flowfield in Aerospace Application Using Earth Simulator. *AIAA Paper No. 2006-2823.*

[46] Harris, R., Wang, Z., & Liu, Y. (2007). Efficient Implementation of High-Order Spectral Volume Method for Multidimensional Conservation Laws on Unstructured Grids. *AIAA Paper No. 2007-912.*

[47] Harten, A. (1989). ENO schemes with subcell resolution. *J. Comput. Phys. , 83*, 148-184.

[48] Harten, A. (1983). High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys. , 49*, 357-393.

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

[50] Hesthaven, J., & Teng, C. (2000). Stable spectral methods on tetrahedral elements. *SIAM J. Sci. Comput. , 21*, 2352-2380.

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

[52] Huang, P., Wang, Z., & Liu, Y. (2005). An Implicit Space-Time Spectral Difference Method for Discontinuity Capturing Using Adaptive Polynomials. *AIAA-2005-5255.*

[53] Huynh, H. (2007). A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. *AIAA paper 2007-4079.*

[54] Huynh, H. (2009). A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion. *AIAA paper 2009-403.*

[55] Jaffre, J., Johnson, C., & Szepessy, A. (1995). Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Mathematical Models and Methods in Applied Sciences , 5*, 367–386.

[56] Jameson, A., & Caughey, D. (2001). How many steps are required to solve the Euler equations of steady compressible flow: in search of a fast solution algorithm. *15th AIAA Computational Fluid Dynamics Conference.* Anaheim, CA.

[57] Jameson, A., Schmidt, W., & Turkel, E. (1981). Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA Paper 81-1259.*

[58] Jiang, G., & Shu, C.-W. (1996). Efficient implementation of weighted ENO schemes. *J. Comput.Phys. , 126*, 202.

[59] Kannan, R., Sun, Y., & Wang, Z. (2008). A Study of Viscous Flux Formulations for an Implicit P-Multigrid Spectral Volume Navier Stokes solver. *AIAA paper, 2008-783.*

[60] Kopriva, D. (1998). A Staggered-Grid Multidomain Spectral Method for the Compressible Navier–Stokes Equations. *J. Comput. Phys , 143*, 125-148.

[61] Kopriva, D., & Kolias, J. (1996). A conservative staggered-grid Chebyshev multidomain method for compressible flows. *J. Comput. Phys , 125*, 244.

[62] Krividonova, L., & Berger, M. (2006). High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys. , 211*, 492-512.

[63] Krivodonova, L. (2007). Limiters for high-order discontinuous Galerkin methods. *J. Comput. Phys. , 226*, 879-896.

[64] Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugeon, N., & Flaherty, J. (2004). Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math. , 48*, 323-338.

[65] Kuzmin, D. (2009). A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *Submitted to J. Comput. Appl. Math.*

[66] Lau, T., & Lo, S. (1996). Finite Element Mesh Generation Over Analytical Surfaces. *Computers and Structures , 59* (2), 301-309.

[67] Laug, P., Borouchaki, H., & George, P.-L. (1996). Maillage de courbes gouverné par une carte de métriques. *Technical Report RR-2818, INRIA.*

[68] Lax, P., & Wendroff, B. (1960). Systems of conservation laws. *Comm. Pure Appl. Math. , 13*, 217.

[69] Liu, M. (1996). A sequel to AUSM: AUSM+. *J. Comput. Phys. , 129.*

[70] Liu, X., Osher, S., & Chan, T. (1994). Weighted essentially non-oscillatory schemes. *J. Comput. Phys. , 115*, 200-212.

[71] Liu, Y. (2005). Central schemes on overlapping cells. *J. Comput. Phys. , 209*, 82-104.

[72] Liu, Y., Shu, C.-W., Tadmor, E., & Zhang, M. (2007). Central Discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction. *SIAM J. Numer. Anal. , 45*, 2442-2467.

[73] Liu, Y., Vinokur, M., & Wang, Z. (2004). Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids. *Proceedings of the 3rd International Conference on Computational Fluid Dynamics,.* Toronto, Canada.

[74] Liu, Y., Vinokur, M., & Wang, Z. (2006). Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids. *J. Comput, Phys. , 216*, 780-801.

[75] Liu, Y., Vinokur, M., & Wang, Z. (2006). Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems. *J. Comput. Phys. , 212*, 454-472.

[76] Liu, Y., Vinokur, M., & Wang, Z. (2006). Spectral Difference Method for Unstructured Grids I: Basic Formulation. *J. Comput. Phys.* , 780-801.

[77] Luo, H., Baum, J., & Lohner, R. (2007). A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. *J. Comput. Phys. , 225*, 686–713.

[78] MacCormack, R., & Baldwin, B. (1975). A numerical method for solving the Navier–Stokes equations with application to shock-boundary layer interaction. *AIAA 75-1.*

[79] Martin, M. (2007). Direct numerical simulation of hypersonic turbulent boundary layers.

Part 1. Initialization and comparison with experiments. *Journal of Fluid Mechanics , 570*, 347-364.

[80] Mavriplis, D. (1998). Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes. *J. Comput. Phys. , 145*, 141–165.

[81] Mavriplis, D. (1997). Unstructured grid techniques. *Annu. Rev. Fluid. Mech , 29*, 473–514.

[82] May, G., & Jameson, A. (2006). A spectral difference method for the Euler and Navier-Stokes equations. *AIAA paper No. 2006-304.*

[83] May, G., & Jameson, A. (2005). High-Order Accurate Methods for High-Speed Flow. *AIAA pape rNo. 2005-5251.*

[84] Nessyahu, H., & Tadmor, E. (1990). Non-oscillatory central differencing for hyperbolic conservation laws. *J. Comput. Phys. , 87* (2), 408-463.

[85] Nishikawa, H. (2006). Higher order discretization of diffusion terms in residual distribution methods. In H. Deconinck, & M. Ricchiuto (Eds.), *CFD – Higher Order Discretization Methods.* von Karman Institute for Fluid Dynamics.

[86] Ollivier-Gooch, C. (1997). Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction. *J. Comput. Phys.* , 6–17.

[87] Paillere, H., Deconinck, H., & van der Weide, E. (1997). Upwind residual distribution methods for compressible flow: an alternative to finite volume and finite element methods. *28th CFD VKI Lecture Series.*

[88] Paillere, H., Deconinck, H., Struijs, R., Roe, P., Mesaros, L., & Muller, J. (1993). Computations of compressible flows using fluctuation splitting on triangular meshes. *AIAA Paper 93-3301-CP.*

[89] Persson, P., & Peraire, J. (2006). Sub-cell shock capturing for discontinuous Galerkin methods. *AIAA-2006-112.*

[90] Qiu, J., & Shu, C.-W. (2005). A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin Methods using weighted essentially nonoscillatory limiters. *SIAM J. Sci. Comput. , 27* (3).

[91] Qiu, J., & Shu, C.-W. (2003). Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case.

J. Comput. Phys. , 193, 115–135.

[92] Qiu, J., & Shu, C.-W. (2005). Runge–Kutta discontinuous Galerkin method using WENO limiters. *SIAM J. Sci. Comput. , 26*, 907-929.

[93] Rasetarinera, P., & Hussaini, M. (2001). An efficient implicit discontinuous Galerkin method. *J. Comput.Phys. , 172* (2), 718–738.

[94] Reed, W., & Hill, T. (1973). Triangular mesh methods for the neutron transport equation. *Tech. Report LA-UR-73-479.* Los Alamos Scientific Laboratory.

[95] Ricchiuto, M., Csik, A., & Deconinck, H. (2005). Residual distribution for general time-dependent conservation laws. *J. Comput. Phys. , 209*, 249-289.

[96] Roe, PL (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* , 357-372.

[97] Roe, PL (1990). *Optimum upwind advection on a triangular mesh.* ICASE Report 90-75.

[98] Saad, Y., & Schultz, M. (1986). GMRES: A genearlized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp , 7*, 865.

[99] Sharov, D., & Nakahashi, K. (1998). Low Speed Preconditioning and LUSGS Scheme for 3D Viscous Flow Computations on Unstructured Grids. *AIAA Paper 98-0614.*

[100] Shu, C.-W. (1998). Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In A. Quarteroni (Ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics* (Vol. 1697, p. 325). Berlin/New York: Springer-Verlag.

[101] Shu, C.-W. (1988). Total-Variation-Diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing , 9*, 1073-1084.

[102] Shu, C.-W., & Osher, S. (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys. , 77*, 439-471.

[103] Soetrisno, M., Imlay, S., & Roberts, D. (1994). A Zonal Implicit Procedure for Hybrid Structured-Unstructured Grids. *AIAA Paper 94-0617.*

[104] Sonar, T. (1997). On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: Polynomial recovery, accuracy and stencil selection.

Comput. Methods Appl. Mech. Engrg. , *140*, 157.

[105] Spiteri, R., & Ruuth, S. (2002). A new class of optimal high-order strong-stability preserving time discretization methods. *SIAM J. Numer. Anal.* , *40*, 469–491.

[106] Sun, Y., & Wang, Z. (2004). Evaluation of Discontinuous Galerkin and Spectral Volume Methods for Scalar and System Conservation Laws on Unstructured Grid. *Int. J. Numer. Meth. Fluids.* , *45* (8), 819-838.

[107] Sun, Y., & Wang, Z. (2004). Formulations and analysis of the spectral volume method for the diffusion equation. *Commun. Numer. Meth.ENgng* , *20*, 927–937.

[108] Sun, Y., & Wang, Z. (2006). High-Order Multidomain Spectral Difference Method for the Navier-Stokes Equations. *AIAA-2006-0301.*

[109] Sun, Y., Wang, Z., & Liu, Y. (2009). Efficient Implicit Non-linear LU-SGS Approach for Compressible Flow Computation Using High-Order Spectral Difference Method. *Commun. Comput. Phys* , *5* (2-4), 760-778.

[110] Sun, Y., Wang, Z., & Liu, Y. (2007). High-Order Multidomain Spectral Difference Method for the Navier-Stokes Equations on Unstructured Hexahedral Grids. *Comm.Comp. Phy.* , *2* (2), 310-333.

[111] Sun, Y., Wang, Z., & Liu, Y. (2006). Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids VI: Extension to Viscous Flow. *J. Comput. Phys.* , *215* (1), 41-58.

[112] Sun, Y., Wang, Z., Liu, Y., & Chen, C. (2007). Efficient Implicit LU-SGS Algorithm for High-Order Spectral Difference Method on Unstructured Hexahedral Grids. *AIAA Paper No. 2007-0313.*

[113] Suresh, A., & Huynh, H. (1997). Accurate monotonicity-preserving schemes with Runge-Kutta time stepping. *J. Comput. Phys.* , *136*.

[114] Tadmor, E. (1989). Convergence of spectral methods for nonlinear conservation laws. *SIAM J. Numer. Anal.* , *26*, 30.

[115] Taylor, E., Wu, M., & Martin, P. (2007). Optimization of nonlinear error for weighted essentially non-oscillatory methods in direct numerical simulations of compressible turbulence. *Journal of Computational Physics* , *223*, 384-397.

[116] Tristano, J., Owen, S., & Canann, S. (1998). Advancing Front Surface Mesh

Generation in Parametric Space Using a Riemannian Surface Definition. *7th International Meshing Roundtable* .

[117] Van den Abeele, K., & Lacor, C. (2008). On the stability and accuracy of the spectral difference method. *J. Sci. Comput. , 37* (2), 162-188.

[118] Van den Abeele, K., Lacor, C., & Wang, Z. (2007). On the connection between the spectral volume and the spectral difference method. *J. Comput. Phys. , 227* (2), 877–885.

[119] Van Leer, B. (1974). Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys. , 14*, 361.

[120] van Leer, B. (1977). Towards the ultimate conservative difference scheme iv: A new approach to numerical convection. *J. Comput. Phys. , 23*, 276–299.

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

[122] Van Leer, B., & Lo, M. (2009). Unification of Discontinuous Galerkin Methods for Advection and Diffusion. *AIAA paper 2009-0400.*

[123] Van Leer, B., & Normua, S. (2005). Discontinuous Galerkin for diffusion. *AIAA paper 2005-5108.*

[124] Van Leer, B., Lo, M., & Van Raalte, M. (2007). A discontinuous Galerkin method for diffusion based on recovery. *AIAA paper 2007-4083.*

[125] Von Neumann, J., & Richtmyer, R. (1950). A method for the numerical calculations of hydrodynamical shocks. *J. Math.Phys. , 21.*

[126] Wait, R., & Mitchell, A. (1985). *Finite Element Analysis and Applications.* Wiley, Chichester.

[127] Wang, Z. (2007). High-order methods for Euler and Navier-Stokes equations on unstructured grids. *Progress in Aerospace Sciences , 43* (1-3), 1-41.

[128] Wang, Z. (2002). Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids: Basic Formulation. *J. Computational Physics , 178*, 210-251.

[129] Wang, Z., & Liu, Y. (2002). Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation.

J Comp Phys , 179 (2), 665–97.

[130] Wang, Z., & Gao, H. (2009). A unifying lifting collocation penalty formulation for the Euler equations on mixed grids. *AIAA paper 2009-401.*

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

[132] Wang, Z., & Liu, Y. (2004). Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization. *J Sci Comp , 20*, 137–57.

[133] Wang, Z., & Liu, Y. (2005). The Spectral Difference Method for the 2D Euler Equations on Unstructured Grids. *AIAA paper 2005-5112.*

[134] Wang, Z., Liu, Y., May, G., & Jameson, A. (in press). Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations. *Journal of Scientific Computing* .

[135] Wang, Z., Sun, Y., Liang, C., & Liu, Y. (2006). Extension of the SD Method to Viscous Flow on Unstructured Grids. *the 4th International Conference on Computational Fluid Dynamics.* Gent, Belgium.

[136] Wang, Z., Zhang, L., & Liu, Y. (2004). Spectral finite volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems. *J Comp Phys , 194* (2), 716–741.

[137] Yang, M., & Wang, Z. (2009). A parameter-free generalized moment limiter for high-order methods on unstructured grids. *Adv. Appl. Math. Mech. , 1*, 451-480.

[138] Yang, M., & Wang, Z. (2009). A parameter-free generalized moment limiter for high-order methods on unstructured grids. *AIAA paper 2009-605.*

[139] Yang, M., & Wang, Z. (2007). Efficient Quadrature-Free 3D High-Order Spectral Volume Method on Unstructured Grids. *AIAA paper 2007-4325.*

[140] Yee, H., Warming, R., & Harten, A. (1985). Implicit TVD schemes for steady-state calculations. *J. Comput. Phys. , 57*, 327-360.

[141] Yoon, S., & Jameson, A. (1988). Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. *AIAA Journal , 26*, 1025-1026.

[142] Zhang, M., & Shu, C.-W. (2005). An analysis and a comparison between the

discontinuous Galerkin method and the spectral finite volume methods. *Comput Fluids , 34* (4-5), 581–592.

[143] Zhang, S., Zhang, Y.-T., & Shu, C.-W. (2006). Interaction of a shock wave with an oblique vortex pair: shock dynamics and mechanism of sound generation. *Physics of Fluids , 18*, 126101.

[144] Zhang, S., Zhang, Y.-T., & Shu, C.-W. (2005). Multistage interaction of a shock wave and a strong vortex. *Physics of Fluids , 17*, 116101.

[145] Zhang, Y.-T., Shi, J., Shu, C.-W., & Zhou, Y. (2003). Numerical viscosity and resolution of high-order weighted essential nonoscillatory schemes for compressible flows with high Reynolds numbers. *Physical Review E , 68*, 046709.

[146] Zhang, Y.-T., Shu, C.-W., & Zhou, Y. (2006). Effects of shock waves on Reyleigh-Taylor instability. *Physics of Plasmas , 13*, 062705.